دليل عمليات OmniCRM

OmniCRM هي الحل الشامل من Omnitouch لإدارة أنظمة الأعمال (BSS) وإدارة العمليات (OSS) لمقدمي خدمات الهاتف المحمول والثابت. منصة كاملة تتعامل مع كل شيء من تسجيل العملاء إلى الفوترة، والتزويد، والدعم - كل ذلك في نظام متكامل واحد.

البدء

لموظفي خدمة العملاء

خطواتك الأولى:

1. تعرف على التنقل والبحث - **تعلم الواجهة**
2. إنشاء عميل خطوة بخطوة - **إنشاء عميل**
3. تزويد خدمتك الأولى - **إضافة خدمة**
4. التعامل مع مدفوعات العملاء - **معالجة الدفع**
5. إضافة رصيد لحسابات العملاء - **إعادة شحن الخدمات**

المهام اليومية:

- إدارة خدمات العملاء - **إدارة الخدمات**
- تحقق من الأرصدة والاستخدام - **عرض الاستخدام**
- فحص سجلات المكالمات والرسائل النصية وسجلات البيانات (CDRs) - **عرض سجلات المكالمات**
- تغيير تكوينات الخدمة - **تعديل الخدمات**
- إنشاء وإرسال الفواتير - **إنشاء فواتير**

لمشرفي النظام

الإعداد والتكوين:

1. نظرة عامة على النظام - **فهم الهيكلية**
2. إعدادات النظام - **تكوين النظام**

3. بناء الكتالوج الخاص بك - **إنشاء المنتجات**
4. أتمتة التزويد - **كتابة كتب التشغيل**
5. إنشاء حسابات وتعيين أدوار - **إعداد المستخدمين**

المواضيع المتقدمة:

- **إدارة المخزون** - إدارة الأصول
- **تكامل API** - ربط الأنظمة الخارجية
- **إعداد الأمان** (2FA) والأذونات - تكوين المصادقة الثنائية

للعملاء

استخدام بوابة الخدمة الذاتية:

- **الوصول إلى حسابك** - تسجيل الدخول والتنقل
- **عرض الخدمات** - رؤية خدماتك النشطة
- **تحقق من الاستخدام** - مراقبة البيانات والأرصدة
- **دفع الفواتير** - إجراء المدفوعات عبر الإنترنت
- **إعادة شحن الخدمات** - إضافة رصيد على الفور

مرجع سريع للمهام

الوثائق	أريد أن...
إنشاء عميل	إنشاء عميل جديد
إضافة خدمة	إضافة خدمة لعميل
استخدام الخدمة	عرض استخدام الخدمة والأرصدة
سجلات تفاصيل المكالمات	عرض سجلات المكالمات/الرسائل النصية/البيانات لخدمة
معالجة الدفع	معالجة الدفع
إدارة الفواتير	إنشاء فاتورة
إعادة الشحن	إعادة شحن خدمة
نظام المخزون	إدارة المخزون
دورة حياة المنتج	إنشاء منتج
كتب تشغيل Ansible	كتابة كتاب تشغيل للتزويد
RBAC	إعداد حسابات المستخدمين
المصادقة الثنائية	تفعيل المصادقة الثنائية (2FA)
البحث العالمي	البحث عن أي شيء
هيكل النظام	فهم النظام
وثائق API	استخدام API

مكتبة الوثائق

القراءة الأساسية

- **هيكل النظام** - نظرة تقنية كاملة مع الرسوم البيانية
- **دليل دورة حياة المنتج** - إدارة المنتجات من البداية إلى النهاية
- **دليل كتب تشغيل Ansible** - اتقن أتمتة التزويد

إدارة العملاء

- العملاء، جهات الاتصال والمواقع - نموذج البيانات والعلاقات
- إنشاء العملاء - دليل خطوة بخطوة
- علامات العملاء - تنظيم باستخدام العلامات
- سمات العملاء - بيانات وصفية مخصصة
- سجل النشاط - تتبع جميع التغييرات

الدعم

- نظام تذاكر الدعم - إدارة تذاكر دعم العملاء
- قاعدة المعرفة للأسئلة الشائعة - إجابات الخدمة الذاتية للمشاكل الشائعة

إدارة العمل

- اللوحات - إدارة العمل بأسلوب كانبان

عمليات الخدمة

- إدارة الخدمة - نظرة عامة
- إضافة خدمات - سير عمل التزويد
- تعيين الخطط - تعيين الخطط
- تعديل الخدمات - إجراء التغييرات
- استخدام الخدمة والأرصدة - مراقبة الاستخدام
- سجلات تفاصيل المكالمات (CDRs) - عرض سجلات المكالمات والرسائل النصية وسجلات البيانات

- إعادة الشحن - إضافة أرصدة

الفوترة والمالية

- نظرة عامة على الفوترة - دليل الفوترة الكامل
- دليل نظام الدفع - هيكل نظام الدفع
- طرق الدفع - إدارة طرق الدفع
- المعاملات - الرسوم والائتمانات
- معالجة المدفوعات - سير عمل المدفوعات
- إدارة الفواتير - إنشاء وإدارة الفواتي��

الإدارة

- تكوين النظام - تكوين OmniCRM
- إدارة المخزون - تتبع الأصول
- تزويد بطاقة SIM - إدارة دورة حياة بطاقة SIM
- إدارة مخزون eSIM - تتبع وتعيين eSIM
- مفاتيح API - إدارة الوصول إلى API
- المراقبة والقياسات - صحة النظام وقابلية المراقبة

الأمان والوصول

- سير عمل المصادقة - كيفية عمل المصادقة
- المصادقة الثنائية (2FA) - إعداد المصادقة الثنائية
- التحكم في الوصول القائم على الأدوار (RBAC) - المستخدمون والأدوار والأذونات

التكاملات

- حملات البريد الإلكتروني والرسائل النصية (EDM) - حملات الاتصال الجماعي
- تكامل Mailjet - خدمة البريد الإلكتروني
- الإشعارات API - تكامل الإشعارات الصادرة
- وثائق API - مرجع REST API

واجهة العملاء

- بوابة الخدمة الذاتية - دليل بوابة العملاء
- رعاية العملاء - ميزات الخدمة الذاتية
- نظام البث الخلوي - تنبيهات الطوارئ

المرجع

- سجل التغييرات - تاريخ الإصدارات
- معجم المصطلحات - مرجع المصطلحات

ما الذي يجعل OmniCRM مميزًا؟

يجمع OmniCRM جميع الأدوات التي تحتاجها لتشغيل عمل حديث في مجال الاتصالات:

تزويد الخدمة الآلي

يقوم OmniCRM **أتمتة** بإزالة التكوين اليدوي. يستخدم **Ansible** لتزويد الخدمات في ثوانٍ. يتولى النظام العمل تلقائيًا عند تفعيل بطاقة SIM، أو إعداد حزمة معقدة، أو تكوين معدات العملاء.

ما يقوم به:

- يزود الخدمات بنقرة واحدة
- يقوم بتكوين معدات الشبكة تلقائ��ا
- ينشئ حسابات الفوترة في الوقت الحقيقي
- يرسل رسائل ترحيب عبر البريد الإلكتروني والرسائل النصية
- يتراجع تلقائيًا إذا فشل شيء ما

Ansible تعرف على المزيد حول التزويد | انظر دليل كتب تشغيل

الفوترة والمدفوعات الذكية

معالجة Stripe تقييمًا وشحنًا في الوقت الحقيقي، بينما تتولى **CGRateS** يوفر التكامل المدمج مع المدفوعات بسلاسة.

ما يقوم به:

- تتبع الاستخدام والتقييم في الوقت الحقيقي
- إنشاء الفواتير تلقائيًا
- معالجة بطاقات الائتمان عبر Stripe
- نماذج فوترة مسبقة الدفع وما بعد الدفع
- تسعير مرن ورموز ترويجية
- حساب الضرائب وإعداد التقارير

استكشاف ميزات الفوترة | معالجة المدفوعات | إدارة الفواتير

إدارة العملاء الكاملة

إدارة العملاء، جهات الاتصال، المواقع المتعددة، والخدمات مع نموذج علاقة شامل.

ما يقوم به:

- تتبع العملاء وجهات الاتصال ومواقع الخدمة
- إدارة خدمات متعددة لكل عميل
- تخزين سمات وبيانات وصفية مخصصة
- وضع علامات على العملاء للتنظيم
- تاريخ نشاط كامل وسجلات تدقيق
- تكامل مع خرائط Google لتحديد المواقع الجغرافية

دليل إدارة العملاء | إنشاء عميل | علامات العملاء

إدارة المخزون

تتبع وإدارة جميع الأصول المادية والافتراضية - من بطاقات SIM إلى أجهزة التوجيه إلى كتل عناوين IP.

ما يقوم به:

- تتبع بطاقات SIM والمعدات والإكسسوارات
- إدارة أرقام الهواتف وكتل عناوين IP
- تعيين تلقائي أثناء التزويد
- قدرات استيراد وتصدير جماعية

- تخزين تكوين المعدات
- قوالب مخزون مخصصة

وثائق نظام المخزون

بوابة الخدمة الذاتية للعملاء

مكّن عملاءك من خلال **بوابة الخدمة الذاتية** حيث يمكنهم إدارة خدماتهم الخاصة.

ما يمكن للعملاء القيام به:

- عرض الخدمات والاستخدام
- التحقق من الأرصدة وبدلات البيانات
- دفع الفواتير عبر الإنترنت
- تنزيل الفواتير كملفات PDF
- تحديث معلومات الاتصال
- **إعادة شحن خدماتهم** على الفور

دليل بوابة الخدمة الذاتية | **ميزات رعاية العملاء**

أمان المؤسسات

بُني مع الأمان في جوهره، ويتميز بمصادقة وتفويض شاملة.

ميزات الأمان:

- **المصادقة المعتمدة على JWT**
- **المصادقة الثنائية (2FA)** مع رموز TOTP ورموز النسخ الاحتياطي
- **التحكم في الوصول القائم على الأدوار (RBAC)** مع أذونات دقيقة
- التحقق من البريد الإلكتروني لتغييرات الحساب
- تسجيل تدقيق كامل عبر سجل النشاط
- إدارة الجلسات وضوابط المهلة

وثائق | **دليل المصادقة RBAC** | **إعداد 2FA**

تكاملات قوية

يتكامل OmniCRM بسلاسة مع الخدمات الرائدة في الصناعة:

- **CGRateS** - محرك الفوترة والتقييم في الوقت الحقيقي لشحنات من مستوى الاتصالات
- **Stripe** - معالجة المدفوعات الآمنة وتخزين البطاقات
- **Mailjet** - تسليم البريد الإلكتروني الاحترافي مع القوالب
- **خرائط Google** - التحقق من العناوين وتحديد المواقع الجغرافية لمواقع دقيقة
- **RESTful API** - دمج مع أنظمتك الحالية

تكامل Mailjet | **وثائق API**

كتالوج منتجات مرن

إنشاء أي نوع من عروض المنتجات - من الخدمات المستقلة البسيطة إلى الحزم المعقدة مع مكونات متعددة.

أنواع المنتجات المدعومة:

- **مستقل** - خدمات فردية (خطط الهاتف المحمول، حزم الإنترنت)
- **حزم** - عروض مجمعة (إنترنت + تلفزيون + هاتف)
- **إضافات** - خدمات تكميلية (إعادة شحن البيانات، المكالمات الدولية)
- **عروض ترويجية** - عروض خاصة وخصومات

يمكن أن تحتوي كل منتج على أتمتة تزويد خاصة بها، وقواعد تسعير، ومنطق عمل.

دليل دورة حياة المنتج | **مفاهيم المنتجات والخدمات**

نظام البث الطارئ

لتنبيهات السلامة العامة OmniCRM لمشغلي الشبكات، يتضمن **نظام البث الخلوي**.

ما يقوم به:

- إرسال تنبيهات الطوارئ إلى المناطق الجغرافية
- دعم الرسائل متعددة اللغات
- استهداف بواسطة مشغل الشبكة المتنقلة

- الامتثال لمعايير تنبيهات الحكومة

وثائق البث الخلوي

بحث وتنقل قوي

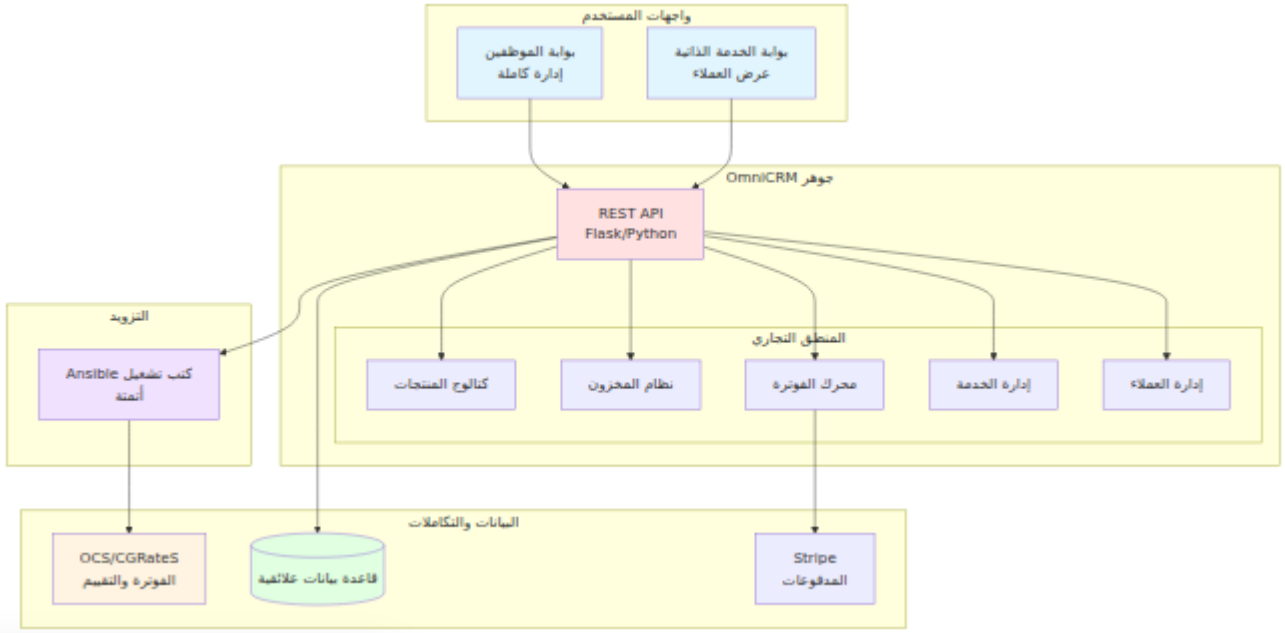
ابحث عن أي شيء على الفور باستخدام **البحث العالمي** عبر العملاء والخدمات والفواتير والمزيد.

ميزات البحث:

- البحث عبر جميع الكيانات
- تصفية وترتيب النتائج
- اختصارات تنقل سريعة
- اقتراحات ذكية

دليل التنقل | **البحث العالمي**

هيكل النظام



عرض الوثائق الكاملة للهيكل

المفاهيم الأساسية

فهم المنتجات مقابل الخدمات

هذه واحدة من أهم المفاهيم في OmniCRM:

- **المنتج** = قالب أو عرض في كتالوجك (مثل "خطة الهاتف المحمول غير المحدودة")
- **الخدمة** = حالة نشطة من منتج لعميل معين (مثل "خطة الهاتف المحمول غير المحدودة لجون سميث")

لإنشاء الخدمة (أو الخدمات) الفعلية. يمكن أن **Ansible** عند تزويد منتج، يستخدم النظام **أتمتة** ينشئ منتج واحد خدمات متعددة، أو لا خدمات (تكوين فقط)، أو تعديل خدمات موجودة.

تعرف على المزيد حول المنتجات والخدمات

سحر التزويد

السر في OmniCRM هو **نظام التزويد الآلي**:



إلى تكوين أجهزة التوجيه إلى CGRateS يتولى كتاب التشغيل كل شيء - من إنشاء الحسابات في إرسال رسائل الترحيب عبر البريد الإلكتروني. وإذا فشل أي شيء، فإنه يتراجع تلقائيًا.

نظام التزويد | **استكشاف: كتب تشغيل Ansible**

الفوترة بسيطة



تولد كل خدمة **معاملات** (رسوم أو ائتمانات). يتم تجميعها في **فواتير** وإرسالها للعملاء. مع تكامل Stripe، يمكن معالجة المدفوعات تلقائيًا.

نظرة عامة على الفوترة | **معالجة المدفوعات**

هيكلية العملاء

```
العميل (شركة ABC)
├── جهات الاتصال (جون سميث - الفوترة، جين دو - التقنية)
├── المواقع (المكتب الرئيسي، فرع المكتب)
└── الخدمات
    ├── الإنترنت - المكتب الرئيسي
    ├── خطة الهاتف المحمول - جون سميث
    └── VoIP - فرع المكتب
```

كل شيء منظم بشكل هرمي، مما يجعل من السهل إدارة العملاء مع مواقع وخدمات متعددة.

العملاء، جهات الاتصال والمواقع

لماذا تختار OmniCRM؟

منصة شاملة

كل ما تحتاجه في نظام متكامل واحد - لا حاجة للتلاعب بأدوات أو بائعين متعددين.

الأتمتة أولاً

يعني أن الخدمات يتم نشرها في ثوانٍ، وليس ساعات. تقليل الأخطاء، Ansible تزويد مدعوم من توفير الوقت، وزيادة رضا العملاء.

مرن وقابل للتخصيص

من تعريفات المنتجات إلى كتب تشغيل التزويد إلى السمات المخصصة - قم بتخصيص OmniCRM لتناسب عمليات عملك الدقيقة.

مصمم للاتصالات

مصمم خصيصًا لمقدمي الخدمات مع ميزات مثل تكامل CGRateS، إدارة المخزون، والتقييم في الوقت الحقيقي.

أمان المؤسسات

تضمن المصادقة المعتمدة على JWT، و المصادقة الثنائية، والRBAC، وتسجيل التدقيق الكامل أن بياناتك آمنة ومتوافقة.

API مدفوع بـ

OmniCRM شامل يعني أنه يمكنك دمج API REST مع أي أنظمة موجودة أو بناء أدوات مخصصة.

احصل على الدعم

موارد الوثائق

- ابدأ بـ **هيكل النظام** للحصول على نظرة تقنية عامة
- انظر **أدلة البدء** للتوجيه حسب الدور
- تحقق من **مرجع المهام السريع** للعمليات الشائعة
- استشر المعجم للمصطلحات

تحتاج إلى مزيد من المساعدة؟

تصفح مكتبة الوثائق الكاملة أعلاه أو استخدم **البحث العالمي** للعثور على مواضيع محددة.

OmniCRM - BSS/OSS كاملة لمقدمي الخدمات الحديثة

آخر تحديث: 23-12-2025

المصادقة الثنائية (2FA)

أمان معزز باستخدام كلمات مرور لمرة واحدة تعتمد على الوقت

يدعم OmniCRM **المصادقة الثنائية (2FA)** باستخدام كلمات مرور لمرة واحدة تعتمد على الوقت (TOTP). يضيف هذا طبقة إضافية من الأمان من خلال مطالبة المستخدمين بتقديم كل من كلمة المرور الخاصة بهم ورمز حساس للوقت من تطبيق المصادقة.

انظر أيضًا: RBAC 2 إدارة الأذونات لـ2FA، تدفقات المصادقة لتفاصيل عملية تسجيل الدخول.

الغرض

توفر 2FA:

1. **أمان معزز**: تحمي الحسابات حتى إذا تم اختراق كلمات المرور.
2. **امتثال**: تلبي متطلبات الأمان للصناعات المنظمة.
3. **اختيار المستخدم**: اختيارية للمستخدمين، يمكن فرضها حسب الدور أو عالميًا.
4. **معيار صناعي**: تستخدم بروتوكول TOTP المتوافق مع Google Authenticator وMicrosoft Authenticator وAuthy وغيرها من التطبيقات القياسية.

كيفية عمل 2FA

عندما يتم تفعيل 2FA لمستخدم:

1. **الإعداد**: يقوم المستخدم بمسح رمز QR باستخدام تطبيق المصادقة الخاص به أثناء التسجيل.
2. **تسجيل الدخول**: بعد أن يدخل المستخدم اسم المستخدم وكلمة المرور، يقدم المستخدم الرمز المكون من 6 أرقام من تطبيقه.
3. **التحقق**: يتحقق النظام من أن الرمز المعتمد على الوقت يتطابق مع القيمة المتوقعة.
4. **الوصول**: يحصل المستخدم على الوصول فقط بعد التحقق من كلا العاملين.

2 تفعيلFA

للمستخدمين الفرديين

2 تفعيلFAيمكن للمستخدمين لحسابهم الخاص:

1. انتقل إلى **إعدادات المستخدم** أو **الملف الشخصي**
2. اختر **تفعيل المصادقة الثنائية**
3. امسح رمز QR باستخدام تطبيق المصادقة
4. أدخل رمز التحقق لتأكيد الإعداد
5. احفظ رموز النسخ الاحتياطي في مكان آمن

النسخ الاحتياطي والاسترداد

رموز النسخ الاحتياطي

2 تفعيلFAعند، يتلقى المستخدمون رموز نسخ احتياطي يمكن استخدامها إذا كان جهازهم غير متاح:

- كل رمز يستخدم لمرة واحدة
- قم بتخزين الرموز بأمان خارج الإنترنت

- قم بإنشاء رموز جديدة إذا تم استخدام جميع الرموز

إعادة تعيين المسؤول

إذا فقد المستخدم الوصول إلى تطبيق المصادقة وجميع رموز النسخ الاحتياطي مستنفدة، يمكن لمشرف لديه وصول إلى قاعدة البيانات إعادة تعيين 2FA يدويًا للمستخدم عن طريق مسح حقل *totp_secret* في قاعدة البيانات. يمكن للمستخدم بعد ذلك إعادة تفعيل 2FA.

تكامل API

تفعيل 2FA لمستخدم

POST /2fa/enable/user/{user_id}

```
{
  "password": "current_password"
}
```

تتضمن الاستجابة URI التزويد (لرمز QR) ورموز النسخ الاحتياطي.

التحقق من إعداد 2FA

POST /2fa/verify-setup/user/{user_id}

```
{
  "code": "123456"
}
```

التحقق من 2FA أثناء تسجيل الدخول

POST /2fa/verify/user/{user_id}

```
{
  "code": "123456"
}
```

تُرجع رمز الوصول ورمز التحديث وبيانات المستخدم عند التحقق الناجح.

إعادة إنشاء رموز النسخ الاحتياطي

POST /2fa/backup-codes/regenerate/user/{user_id}

يتطلب المصادقة. تُرجع مجموعة جديدة من رموز النسخ الاحتياطي.

أفضل الممارسات

- **رموز النسخ الاحتياطي أولاً.** احفظ دائمًا رموز النسخ الاحتياطي قبل إكمال إعداد 2FA.
- **تثقيف المستخدمين.** قدم تعليمات واضحة للإعداد والاسترداد.
- **عملية إعادة تعيين آمنة.** تحقق من هوية المستخدم قبل إعادة تعيين 2FA يدويًا في قاعدة البيانات.

الأسئلة الشائعة

ما هي تطبيقات المصادقة المدعومة؟ أي تطبيق متو��فق مع TOTP (Google Authenticator وAuthy وMicrosoft Authenticator 1وPassword، إلخ).

ماذا أفعل إذا فقدت هاتفي؟ استخدم رمز النسخ الاحتياطي أو اتصل بمشرف لإعادة تعيين 2FA.

هل يمكنني استخدام SMS بدلاً من تطبيق؟ حاليًا، يتم دعم تطبيقات المصادقة TOTP فقط.

هل 2FA مطلوب؟ يعتمد ذلك على سياسة مؤسستك. عادةً ما تكون 2FA مطلوبة للأدوار الإدارية (لا يرى) لحسابات العملاء 2FA ودعم الموظفين ولكنها اختيارية لمستخدمي العملاء. لا يفرض النظام 2FA مستخدمو دور العملاء مطالبات تسجيل 2FA).

ما هي مدة صلاحية رموز TOTP؟ يتم تحديث الرموز كل 30 ثانية ولها نافذة زمنية صغيرة للتحقق (تقبل عادةً الرموز من نافذة الـ 30 ثانية الحالية بالإضافة إلى النوافذ السابقة/اللاحقة لتحمل انحراف الساعة).

API إدارة مفاتيح

توفر **واجهة مستخدم قائمة على الويب** لإنشاء ومراقبة وإدارة مفاتيح API واجهة إدارة مفاتيح المستخدمة للوصول البرمجي إلى واجهة OmniCRM API.

> **ⓘ ملاحظة**
> وأمثلة الاستخدام، راجع API للحصول على مفاهيم المصادقة العامة على `concepts_api`.

نظرة عامة

تمكن مفاتيح API لـ **المصادقة الآمنة وطويلة الأمد** من:

- تكاملات الخادم إلى الخادم
- سكربتات الأتمتة
- التطبيقات الخارجية
- المهام المجدولة ووظائف cron
- أنظمة المراقبة الخارجية

على عكس رموز JWT (التي تنتهي صلاحيتها بعد دقائق/ساعات)، تظل مفاتيح API صالحة حتى يتم إبطالها يدويًا أو حتى تاريخ انتهاء صلاحيتها.

API الوصول إلى إدارة مفاتيح

انتقل إلى:

أو مباشرة:

الإذن المطلوب: `MANAGE_API_KEYS` (دور المسؤول)

API عرض قائمة مفاتيح

تعرض الصفحة الرئيسية جميع مفاتيح API في تنسيق جدول:

الأعمدة:

- **الاسم** - تسمية وصفية لمفتاح API (مثل "نظام التزويد"، "أداة المراقبة")
- **أنشأ بواسطة** - اسم المستخدم للشخص الذي أنشأ المفتاح
- **مفتاح API** - سلسلة المفتاح الفعلية (مخفية جزئيًا لأغراض الأمان)
- **الحالة** - نشط، منتهي، أو ملغى
- **تاريخ الإنشاء** - متى تم إنشاء المفتاح
- **تاريخ الانتهاء** - متى سينتهي صلاحية المفتاح تلقائيًا
- **الإجراءات** - أزرار تحرير، حذف، إعادة إنشاء

مثال على العرض:

عناصر واجهة المستخدم في لوحة المعلومات

في أعلى الصفحة، يتم عرض إحصائيات ملخصة:

- **إجمالي مفاتيح API** - عدد جميع مفاتيح API (نشطة وغير نشطة)
- **المفاتيح النشطة** - المفاتيح الصالحة حاليًا
- **ستنتهي قريبًا** - المف��تيح التي ستنتهي صلاحيتها في الـ 30 يومًا القادمة
- **المفاتيح المنتهية** - المفاتيح التي تجاوزت تاريخ انتهاء صلاحيتها

إنشاء مفتاح API

الخطوة 1: انقر على "إضافة مفتاح API"

انقر على زر **+ إضافة** في أعلى يمين قائمة مفاتيح API.

الخطوة 2: املأ التفاصيل

تظهر نافذة نموذج تطلب:

الاسم: _____
(مثل "نظام التزويد")

الوصف: _____
(اختياري - الغرض من هذا المفتاح)

> تاريخ الانتهاء: [اختيار التاريخ]
> (اختياري - اتركه فارغًا لعدم وجود انتهاء)
>
> الأذونات: ☐ عرض العملاء ☐ إنشاء العملاء ☐ عرض الخدمات ☐ إنشاء الخدمات ☐ التزويد
> ☐ عرض المخزون ☑ مسؤول (جميع الأذونات)
>
> [إنشاء المفتاح] [إلغاء]

إرشادات الحقول:

الاسم (مطلوب)

- معرف قصير ووصف
- "أمثلة: "نظام التزويد"، "تكامل الفواتير"، "ال��راقبة
- يستخدم في سجلات التدقيق ويظهر في القائمة

الوصف (اختياري)

- تفسير أكثر تفصيلًا
- مزامنة الفواتير من طرف ثالث"، "Ansible أمثلة: "يستخدم بواسطة خادم تزويد
- يساعد المسؤولين المستقبليين على فهم غرض المفتاح

تاريخ الانتهاء (اختياري)

- إذا كان فارغًا: المفتاح لا ينتهي أبدًا (غير موصى به)
- إذا تم تعيينه: يصبح المفتاح غير صالح تلقائيًا بعد هذا التاريخ
- موصى به: تعيين تاريخ انتهاء للأمان (90 يومًا إلى سنة واحدة)

الأذونات

- اختر أذونات محددة أو تحقق من "مسؤول" للوصول الكامل
- يتبع نفس نموذج الأذونات المعتمد على الدور كما هو الحال في حسابات المستخدمين
- **أفضل ممارسة:** تعيين الحد الأدنى من الأذونات اللازمة

الخطوة 3: إنشاء ونسخ المفتاح

بعد النقر على **"إنشاء المفتاح"**، يعرض النظام مفتاح API الذي تم إنشاؤه حديثًا:

> ⚠ انسخ هذا المفتاح الآن - لن يتم عرضه مرة أخرى!

sk_live_a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0

[نسخ إلى الحافظة]

[إغلاق]

> ⚠️ **WARNING**
>
> **احفظ مفتاح API على الفور!**
>
> بمجرد إغلاق هذه النافذة، لا يمكن استرداد المفتاح الكامل مرة أخرى. سترى فقط نسخة مخفية (`sk_live_...XYZ`) في عرض القائمة.
>
> إذا فقدت المفتاح، يجب عليك **إعادة إنشائه**، مما يلغي المفتاح القديم وقد يكسر التكاملات الحالية.

الخطوة 4: تكوين تطبيقك

استخدم مفتاح API في طلبات تطبيقك:

```
curl -X GET "https://yourcrm.com/crm/customers" \
  -H "X-API-KEY: sk_live_a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0"
```

أو في متغيرات البيئة:

```
export
CRM_API_KEY="sk_live_a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0"
```

إدارة المفاتيح الحالية

عرض تفاصيل المفتاح

انقر على أي اسم مفتاح API لعرض التفاصيل الكاملة:

- الاسم الكامل للمفتاح والوصف
- طابع زمن الإنشاء

- اسم المستخدم للمنشئ
- الأذونات المرتبطة
- إحصائيات الاستخدام (إذا تم تنفيذها)
- سجلات الوصول الحديثة

إعادة إنشاء مفتاح API

إذا تم اختراق مفتاح API أو فقد، قم بإعادة إنشائه:

1. انقر على ⋮ **(ثلاث نقاط)** في عمود الإجراءات
2. اختر **"إعادة إنشاء المفتاح"**
3. أكد الإجراء

> ⚠️ **WARNING**
>
> **إعادة الإنشاء تلغي المفتاح القديم على الفور.**
>
> ستتوقف أي تطبيقات تستخدم المفتاح القديم عن العمل. قم بتحديث جميع التكاملات بالمفتاح الجديد قبل إعادة الإنشاء.

ماذا يحدث:

- يتم إلغاء المفتاح القديم
- يتم إنشاء مفتاح جديد بنفس الأذونات
- يتم عرض المفتاح الجديد (انسخه على الفور)
- الاسم والوصف والأذونات تظل دون تغيير

إلغاء (حذف) مفتاح API

لإزالة مفتاح API بشكل دائم:

1. انقر على ⋮ **(ثلاث نقاط)** في عمود الإجراءات
2. اختر **"حذف"**
3. أكد الحذف

ماذا يحدث:

- يتم إلغاء المفتاح على الفور
- جميع الطلبات باستخدام هذا المفتاح تعيد `401 غير مصرح`
- يتم إزالة المفتاح من قاعدة البيانات
- **لا يمكن التراجع عنه** - لا يمكن استرداد المفتاح

متى يجب الإلغاء:

- لم يعد التكامل مطلوبًا
- تم اختراق المفتاح
- تم إيقاف النظام الذي يستخدم المفتاح
- استبداله بمفتاح جديد بأذونات مختلفة

تحرير تفاصيل مفتاح API

لتعديل تفاصيل مفتاح API:

1. انقر على **⋮ (ثلاث نقاط)** في عمود الإجراءات
2. اختر **"تحرير"**
3. تحديث الاسم، الوصف، تاريخ الانتهاء، أو الأذونات
4. انقر على **"حفظ التغييرات"**

الحقول القابلة للتحرير:

- الاسم
- الوصف
- تاريخ الانتهاء
- الأذونات

غير القابلة للتحرير:

- قيمة المفتاح نفسها (استخدم إعادة الإنشاء للتغيير)
- تاريخ الإنشاء
- أنشأ بواسطة المستخدم

حالة مفتاح API

يمكن أن تحتوي مفاتيح API على عدة حالات:

نشط

- المفتاح صالح ويمكن استخدامه
- ضمن تاريخ الانتهاء (أو لم يتم تعيين انتهاء)
- لم يتم إلغاؤه يدويًا
- يظهر بشارة خضراء

ستنتهي قريبًا

- نشط ولكن سينتهي في الـ 30 يومًا القادمة
- يظهر بشارة برتقالية/تحذيرية
- يُعتبر تدويره قبل انتهاء الصلاحية

منتهي

- تجاوز تاريخ الانتهاء
- لم يعد يقبل المصادقة
- يظهر بشارة حمراء
- يمكن حذفه أو تمديد انتهاء الصلاحية

ملغى

- محذوف/معطل يدويًا
- غير صالح بشكل دائم
- لم يعد يظهر في القائمة النشطة

التصفية والبحث

تدعم قائمة مفاتيح API:

البحث:

ابحث بالاسم أو الوصف أو جزء من المفتاح:

تصفية حسب الحالة:

قائمة منسدلة للتصفية لعرض:

- جميع المفاتيح
- المفاتيح النشطة فقط
- المفاتيح التي ستنتهي قريبًا (الـ 30 يومًا القادمة)
- المفاتيح المنتهية

الفرز:

انقر على رؤوس الأعمدة للفرز حسب:

- الاسم
- تاريخ الإنشاء
- تاريخ الانتهاء
- أنشأ بواسطة

أفضل ممارسات الأمان

إنشاء مفتاح API

- **الطول:** يجب أن تكون المفاتيح على الأقل 32 حرفًا (النظام يفرض ذلك)
- **العشوائية:** يتم إنشاؤها باستخدام مولدات أرقام عشوائية آمنة تشفيرياً
- **التنسيق:** عادةً ما تكون مسبوقة (مثل `sk_live_`) للتعرف

تخزين مفتاح API

في CRM:

- يتم تجزئة المفاتيح قبل التخزين (مثل كلمات المرور)
- يتم عرض المفتاح الكامل مرة واحدة فقط أثناء الإنشاء
- تخزن قاعدة البيانات التجزئة للتحقق
- حتى المسؤولين لا يمكنهم استرداد المفتاح الكامل لاحقًا

في تطبيقك:

- خزنه في متغيرات البيئة، وليس في الكود
- استخدم أنظمة إدارة الأسرار (AWS Secrets Manager، HashiCorp Vault)
- لا تقم بتسجيل المفاتيح في نظام التحكم في الإصدارات
- قم بتدوير المفاتيح بشكل دوري (90-365 يومًا)

إدارة الأذونات

- **مبدأ أقل الامتيازات** - منح الأذونات اللازمة فقط
- تجنب إنشاء مفاتيح المسؤولين ما لم يكن ذلك ضروريًا للغاية
- استخدم مفاتيح منفصلة للأنظمة/الأغراض المختلفة
- مراجعة الأذونات بانتظام

المراقبة والتدقيق

- راقب استخدام مفتاح API عبر سجلات النشاط
- إعداد تنبيهات للأنماط غير العادية للوصول
- مراجعة طوابع "آخر استخدام" بانتظام
- إزالة المفاتيح غير المستخدمة

تدوير المفاتيح

قم بإنشاء سياسة لتدوير المفاتيح:

1. **إنشاء مفتاح جديد** بنفس الأذونات
2. **تحديث التطبيقات** لاستخدام المفتاح الجديد
3. **مراقبة** لضمان عدم استخدام المفتاح القديم
4. **إلغاء المفتاح القديم** بعد فترة السماح

استكشاف الأخطاء وإصلاحها

"401 غير مصرح" عند استخدام مفتاح API

- **السبب:** المفتاح غير صالح، منتهي، أو غير صحيح
- **الإصلاح:**
  - تحقق من نسخ المفتاح بشكل صحيح (لا توجد مسافات إضافية)

- تحقق من حالة المفتاح (نشط مقابل منتهي)
- تأكد من أن المفتاح لديه ال��ذونات المطلوبة
- تأكد من استخدام رأس `X-API-KEY` (ليس `Authorization`)

"لم يتم العثور على مفتاح API" بعد الإنشاء

- **السبب:** قد يكون المفتاح قد تم إنشاؤه ولكن لم يتم تخزينه بشكل صحيح
- **الإصلاح:**
  - تحقق من قائمة مفاتيح API للمدخل الجديد
  - إذا كان مفقودًا، أنشئ مفتاحًا جديدًا
  - أبلغ المسؤول عن المشكلة

مفتاح API سينتهي قريبًا

- **السبب:** تاريخ الانتهاء يقترب (خلال 30 يومًا)
- **الإصلاح:**
  - أنشئ مفتاحًا جديدًا مع انتهاء موسع
  - تحديث التطبيقات لاستخدام المفتاح الجديد
  - إلغاء المفتاح القديم بعد الترحيل

لا يمكن حذف مفتاح API

- **السبب:** قد يكون محميًا أو قيد الاستخدام
- **الإصلاح:**
  - تأكد من أن لديك أذونات المسؤول
  - تحقق مما إذا كان المفتاح مقفلاً/محميًا
  - اتصل بالمسؤول إذا استمرت المشكلة

نقاط نهاية API (لإدارة برمجية)

يمكن أيضًا إدارة مفاتيح API عبر API (تتطلب أذونات المسؤول):

قائمة مفاتيح API

```
GET /crm/api-keys
Authorization: Bearer <admin-token>
```

إنشاء مفتاح API

```
POST /crm/api-keys
Authorization: Bearer <admin-token>
Content-Type: application/json

{
  "name": "تكامل جديد",
  "description": "مزامنة الفواتير من طرف ثالث",
  "expiry_date": "2026-01-10",
  "permissions": ["view_customer", "view_service"]
}
```

الاستجابة:

```
{
  "api_key_id": 123,
  "name": "تكامل جديد",
  "api_key": "sk_live_a1b2c3d4e5f6g7h8i9j0",
  "status": "active",
  "created": "2025-01-10T10:00:00Z",
  "expiry_date": "2026-01-10T23:59:59Z"
}
```

إلغاء مفتاح API

```
DELETE /crm/api-keys/{api_key_id}
Authorization: Bearer <admin-token>
```

تحديث مفتاح API

```
PATCH /crm/api-keys/{api_key_id}
Authorization: Bearer <admin-token>
Content-Type: application/json

{
  "name": "اسم محدث",
  "expiry_date": "2026-12-31"
}
```

حالات الاستخدام الشائعة

حالة الاستخدام 1: تكامل نظام التزويد

إنشاء مفتاح API لخادم تزويد Ansible الخاص بك:

1. إضافة API → انتقل إلى مفاتي��
2. الاسم: "خادم تزويد Ansible"
3. الوصف: "يستخدم بواسطة أتمتة التزويد"
4. الأذونات: التزويد، عرض/إنشاء الخدمات، عرض/تحديث المخزون
5. تاريخ الانتهاء: سنة واحدة
6. انسخ المفتاح وأضفه إلى Ansible في `crm_config.yaml`

حالة الاستخدام 2: تكامل فواتير من طرف ثالث

إنشاء مفتاح للقراءة فقط لتصدير الفواتير:

1. الاسم: "مزامنة الفواتير - QuickBooks"
2. الأذونات: عرض العملاء، عرض المعاملات، عرض الفواتير
3. تاريخ الانتهاء: 90 يومًا (تدوير ربع سنوي)
4. استخدم في سكربت التصدير المجدول

حالة الاستخدام 3: المراقبة والتنبيه

إنشاء مفتاح لجمع مقاييس Prometheus/Grafana:

1. الاسم: "المراقبة - Grafana"
2. الأذونات: عرض الخدمات، عرض التزويد

3. تاريخ الانتهاء: أبداً (تحتاج المراقبة إلى وصول مستمر)
4. تكوين في مصدر بيانات Grafana

حالة الاستخدام 4: واجهة برمجة تطبيقات بوابة العملاء

إنشاء مفتاح لبوابة الخدمة الذاتية للعملاء:

1. الاسم: "خلفية بوابة العملاء"
2. الأذونات: عرض العميل الخاص، عرض الخدمات الخاصة، إنشاء المدفوعات
3. تاريخ الانتهاء: 180 يومًا
4. استخدم في مكالمات واجهة برمجة التطبيقات لخلفية البوابة

الوثائق ذات الصلة

- `concepts_api` - مفاهيم المصادقة على API وأمثلة
- `rbac` - التحكم في الوصول المعتمد على الدور والأذونات
- `2fa` - المصادقة الثنائية لمزيد من الأمان

سمات العملاء

سمات العملاء هي أزواج مرنة من المفاتيح والقيم يمكن إرفاقها بأي سجل عميل لتخزين بيانات تعريفية مخصصة، إعدادات تكوين، أو معلومات خاصة بالأعمال لا تتناسب مع الحقول القياسية للعملاء.

للتصنيف البصري للعملاء والروابط القابلة للنقر، انظر علامات العملاء. لمعلومات العملاء الأساسية، انظر العملاء، جهات الاتصال، المواقع والخدمات.

على عكس الحقول الثابتة في قاعدة البيانات، تتيح السمات لك توسيع سجلات العملاء ديناميكيًا دون تعديل مخطط قاعدة البيانات. وهذا يجعلها مثالية لتخزين بيانات محددة للنشر، معلمات التكامل، أو علامات منطق الأعما�ل المخصصة.

الغرض وحالات الاستخدام

تشمل حالات الاستخدام الشائعة لسمات العملاء:

1. بيانات التكامل

تخزين معرفات الأنظمة الخارجية أو مفاتيح API الخاصة بهذا العميل:

- `external_crm_id` = "SF-12345" (معرف عميل Salesforce)
- `legacy_system_id` = "OLD-CRM-789" (مرجع الهجرة)
- `hubspot_contact_id` = "12345678" (تكامل HubSpot)

2. منطق الأعمال المخصص

تخزين علامات أو إعدادات تتحكم في سلوكيات محددة للعملاء:

- `billing_method` = "quarterly" (تجاوز الفوترة الشهرية الافتراضية)
- `auto_provision` = "true" (تمكين توفير الخدمة تلقائيًا)
- `support_tier` = "premium" (مستوى دعم مخصص)
- `credit_limit` = "10000" (حد ائتماني خاص بالعميل)

3. بيانات الامتثال والتنظيم

تتبع البيانات الوصفية المتعلقة بالامتثال:

- `gdpr_consent_date` = "2025-01-01" (موافقة معالجة البيانات)
- `tax_exempt` = "true" (حالة الإعفاء الضريبي)
- `regulatory_entity` = "FCC-123456" (معرف تنظيمي)

4. البيانات الوصفية التشغيلية

تخزين المعلومات التشغيلية:

- `preferred_contact_method` = "email" (تفضيل التواصل)
- `account_manager` = "<john.smith@company.com>" (مدير الحساب المعين)
- `onboarding_date` = "2025-01-15" (تتبع دورة حياة العميل)
- `churn_risk_score` = "0.23" (تحليلات تنبؤية)

5. معلمات التوفير

تخزين إعدادات التكوين الخاصة بالتوفير:

- `radius_username_format` = "email" (تنسيق RADIUS مخصص)
- `vlan_id` = "100" (تكوين الشبكة)
- `ipv6_enabled` = "true" (علامات الميزات)

السمات مقابل الحقول القياسية

استخدم السمات عندما:

- تكون البيانات محددة للنشر أو تختلف حسب التثبيت
- تتغير المتطلبات بشكل متكرر
- تخزين البيانات الوصفية المحددة للتكامل
- تصميم ميزات جديدة قبل إضافة حقول قاعدة البيانات
- لا تحتاج البيانات إلى استعلامات أو انضمامات معقدة

استخدم الحقول القياسية عندما:

- تكون البيانات أساسية لنموذج العميل (الاسم، البريد الإلكتروني، العنوان)
- يتطلب الأمر بحثًا أو تصفية أو تقارير متكررة
- تحتوي البيانات على قيود على التكامل المرجعي

- تكون الأداء حاسمًا لاستعلامات واسعة النطاق

إدارة السمات عبر واجهة المستخدم

عرض سمات العملاء

لعرض السمات لعميل:

1. انتقل إلى صفحة نظرة عامة العميل
2. انقر على علامة **السمات**
3. سترى جدولًا بجميع السمات لهذا العميل، يظهر:
    - اسم السمة (المفتاح)
    - قيمة السمة
    - تاريخ الإنشاء
    - تاريخ آخر تعديل

إنشاء سمة جديدة

لإنشاء سمة جديدة لعميل:

1. انتقل إلى صفحة نظرة عامة العميل
2. انقر على علامة **السمات**
3. انقر على زر **إضافة سمة**
4. املأ الحقول المطلوبة:
    - **اسم السمة** (مطلوب): المفتاح/الاسم لهذه السمة (مثل، `external_crm_id`)
    - **قيمة السمة** (مطلوب): القيمة التي سيتم تخزينها (مثل، `SF-12345`)
5. انقر على **إنشاء سمة**

اتفاقيات التسمية:

- استخدم الأحرف الصغيرة مع الشرطات السفلية: `external_system_id` ✓
- تجنب المسافات: `external system id` ✗
- اجعل الأسماء وصفية ولكن مختصرة
- استخدم تسمية متسقة عبر العملاء لنفس أنواع السمات

تعديل سمة

لتعديل سمة موجودة:

1. انتقل إلى صفحة نظرة عامة العميل
2. انقر على علامة **السمات**
3. ابحث عن السمة التي تريد تعديلها في الجدول
4. انقر على زر **تعديل** (قلم)
5. عدل اسم السمة أو القيمة
6. انقر على **تحديث السمة**

> **ⓘ ملاحظة**
> تغيير اسم السمة ينشئ زوج مفتاح-قيمة جديد. تأكد من أن هذا لا يكسر التكاملات التي تعتمد على اسم السمة الأصلي.

حذف سمة

لحذف سمة:

1. انتقل إلى صفحة نظرة عامة العميل
2. انقر على علامة **السمات**
3. ابحث عن السمة التي تريد حذفها في الجدول
4. انقر على زر **حذف** (سلة المهملات)
5. أكد الحذف في النافذة المنبثقة

> **⚠ WARNING**
> حذف السمات المستخدمة من قبل التكاملات، سير العم�� الخاص بالتوفير، أو منطق الفوترة قد يتسبب في فشل. تحقق من التبعيات قبل الحذف.

مرجع حقل السمة

تكامل API

يمكن إدارة السمات برمجيًا عبر API:

إنشاء أو تحديث سمة

نقطة النهاية: `PUT /crm/attribute/`

إذن مطلوب: `create_customer_attribute`

جسم الطلب:

```
{
  "customer_id": 123,
  "attribute_name": "external_crm_id",
  "attribute_value": "SF-12345"
}
```

الاستجابة:

```
{
  "attribute_id": 456,
  "customer_id": 123,
  "attribute_name": "external_crm_id",
  "attribute_value": "SF-12345",
  "created": "2025-01-04 10:30:00",
  "last_modified": "2025-01-04 10:30:00"
}
```

تحديث سمة موجودة

نقطة النهاية: `PATCH /crm/attribute/attribute_id/{attribute_id}`

إذن مطلوب: `update_customer_attribute`

جسم الطلب:

```
{
  "attribute_value": "SF-54321"
}
```

الحصول على سمة حسب المعرف

نقطة النهاية: `GET /crm/attribute/attribute_id/{attribute_id}`

إذن مطلوب: `view_customer_attribute`

الاستجابة:

```
{
  "attribute_id": 456,
  "customer_id": 123,
  "attribute_name": "external_crm_id",
  "attribute_value": "SF-12345",
  "created": "2025-01-04 10:30:00",
  "last_modified": "2025-01-04 10:30:00"
}
```

الحصول على جميع السمات حسب معرف العميل

نقطة النهاية: `GET /crm/attribute/customer_id/{customer_id}`

إذن مطلوب: `view_customer_attribute`

الاستجابة:

```
[
  {
    "attribute_id": 456,
    "customer_id": 123,
    "attribute_name": "external_crm_id",
    "attribute_value": "SF-12345",
    "created": "2025-01-04 10:30:00",
    "last_modified": "2025-01-04 10:30:00"
  },
  {
    "attribute_id": 457,
    "customer_id": 123,
    "attribute_name": "billing_method",
    "attribute_value": "quarterly",
    "created": "2025-01-04 10:35:00",
    "last_modified": "2025-01-04 10:35:00"
  }
]
```


حذف سمة

نقطة النهاية: `DELETE /crm/attribute/attribute_id/{attribute_id}`

إذن مطلوب: `delete_customer_attribute`

الاستجابة:


```
{
  "result": "success"
}
```


عمليات السمات بالجملة

إدارة سمات متعددة

لتعيين سمات متعددة لعميل في وقت واحد (م��ل، أثناء عملية الانضمام أو مزامنة التكامل):

```
import requests

customer_id = 123
attributes = [
    {"attribute_name": "external_crm_id", "attribute_value": "SF-
12345"},
    {"attribute_name": "billing_method", "attribute_value":
"quarterly"},
    {"attribute_name": "support_tier", "attribute_value":
"premium"}
]

for attr in attributes:
    attr["customer_id"] = customer_id
    requests.put(
        "https://api.example.com/crm/attribute/",
        json=attr,
        headers={"Authorization": "Bearer YOUR_TOKEN"}
    )
```

استعلام العملاء حسب السمة

بينما لا تحتوي السمات على نقاط نهاية بحث مدمجة، يمكنك تصفية العملاء حسب السمة باستخدام بحث العملاء مع تصفية مخصصة API:

```
# احصل على جميع العملاء، ثم قم بتصفية حسب السمة في كود التطبيق
customers = requests.get("https://api.example.com/crm/customer/").jso

for customer in customers:
    attributes = requests.get(

f"https://api.example.com/crm/attribute/customer_id/{customer['custom
    ).json()

    # ابحث عن العملاء الذين لديهم سمة محددة
    for attr in attributes:
        if attr['attribute_name'] == 'support_tier' and
attr['attribute_value'] == 'premium':
            print(f"عميل مميز: {customer['customer_name']}")
```

ⓘ ملاحظة

للاستعلامات المتكررة المعتمدة على السمات، ضع في اعتبارك إضافة حقول قاعدة بيانات مفهرسة أو تنفيذ نقطة نهاية بحث مخصصة.

أفضل الممارسات

1. اتفاقيات التسمية

- استخدم snake_case: `external_system_id` ✓
- كن وصفيًا: `billing_method` ✓ مقابل `method` ✗
- تجنب الكلمات المحجوزة أو الأحرف الخاصة
- وثق معاني السمات في دليل النشر الخاص بك

2. أنواع البيانات

- تخزن السمات القيم كسلاسل نصية (بحد أقصى 150 حرفًا)
- بالنسبة للقيم المنطقية، استخدم "true"/"false" (حروف صغيرة)
- بالنسبة للتواريخ، استخدم تنسيق ISO 8601: "2025-01-04 10:30:00"
- بالنسبة لبيانات JSON الكبيرة، ضع في اعتبارك استخدام حقول قاعدة بيانات مخصصة بدلاً من ذلك

3. التحقق

- تحقق من قيم السمات في كود التطبيق قبل الحفظ
- استخ��دم تنسيقات قيمة متسقة عبر العملاء
- وثق القيم المتوقعة لكل اسم سمة

4. الوثائق

- احتفظ بسجل لأسماء السمات وأغراضها
- وثق الأنظمة/التكاملات التي تعتمد على سمات معينة
- أدرج أمثلة على القيم الصالحة

5. الهجرة والتنظيف

- قم بمراجعة السمات غير المستخدمة بانتظام

- قم بإزالة السمات القديمة بعد عمليات الهجرة للنظام
- قم بإصدار أسماء السمات عند تغيير المخططات (مثل، `api_key_v2`)

أمثلة على سير العمل

تكامل الانضمام

عند ترحيل العملاء من نظام قديم:

```
# تخزين مرجع النظام القديم لأغراض التصحيح
PUT /crm/attribute/
{
  "customer_id": 123,
  "attribute_name": "legacy_crm_id",
  "attribute_value": "OLD-12345"
}

# تتبع تاريخ الهجرة
PUT /crm/attribute/
{
  "customer_id": 123,
  "attribute_name": "migrated_date",
  "attribute_value": "2025-01-04"
}
```

قواعد الفوترة المخصصة

تجاوز دورة الفوترة الافتراضية لعميل محدد:

```
# تعيين فوترة ربع سنوية
PUT /crm/attribute/
{
  "customer_id": 123,
  "attribute_name": "billing_cycle",
  "attribute_value": "quarterly"
}

# ثم في كود الفوترة، تحقق من السمة قبل المعالجة
attributes = GET /crm/attribute/customer_id/123
billing_cycle = next(
    (a['attribute_value'] for a in attributes if
a['attribute_name'] == 'billing_cycle'),
    'monthly'  # الافتراضي
)
```

علامات الميزات

تمكين الميزات التجريبية لعملاء محددين:

```
# تمكين توفير IPv6
PUT /crm/attribute/
{
  "customer_id": 123,
  "attribute_name": "feature_ipv6_enabled",
  "attribute_value": "true"
}
```

الأذونات

تتطلب عمليات السمات الأذونات التالية:

- `view_customer_attribute` - عرض السمات
- `create_customer_attribute` - إنشاء سمات جديدة
- `update_customer_attribute` - تعديل السمات الموجودة
- `delete_customer_attribute` - إزالة السمات

انظر `rbac` لتكوين التحكم في الوصول القائم على الأدوار.

استكشاف الأخطاء وإصلاحها

السمة لا تظهر في واجهة المستخدم

- تحقق من أنه تم إنشاء السمة (حقق من استجابة API)
- قم بتحديث الصفحة لإعادة تحميل بيانات العميل
- تحقق من أن المستخدم لديه إذن `view_customer_attribute`

لا يمكن تحديث السمة

- تأكد من أن لديك إذن `update_customer_attribute`
- تحقق من أن attribute_id صحيح
- تحقق من أن السمة تنتمي إلى العميل المحدد

فشل التكامل بعد حذف السمة

- استعد السمة بالقيمة السابقة
- قم بتحديث كود التكامل للتعامل مع السمات المفقودة بشكل سلس
- قم بمراجعة تبعيات السمة قبل الحذف

قيمة السمة مقطوعة

- تحتوي قيم السمات على حد 150 حرفًا
- بالنسبة للبيانات الأطول، قم بتقسيمها إلى سمات متعددة أو استخدم حقل ملاحظات العميل
- ضع في اعتبارك تخزين البيانات الكبيرة في حقول قاعدة بيانات مخصصة

دليل التكوين والتخصيص

يغطي هذا الدليل الشامل جميع خيارات التكوين والتخصيص لـ OmniCRM، بما في ذلك إعدادات الخلفية، العلامة التجارية للواجهة الأمامية، التخصيص المرئي، وأفضل الممارسات للنشر.

نظرة عامة

يستخدم OmniCRM نظامي تكوين رئيسيين:

تكوين الخلفية:

- **الملف:** `OmniCRM-API/crm_config.yaml`
- **التنسيق:** YAML
- **يتطلب:** إعادة تشغيل API بعد التغييرات
- **يستخدم لـ:** قاعدة البيانات، التكاملات، الأمان، التزويد

تكوين الواجهة الأمامية (وقت التشغيل، الإنتاج):

- **المصدر:** `crm_config.yaml` → `ui:` كتلة، يتم تقديمها بواسطة `GET /crm/public/ui-config`
- **التنسيق:** JSON في وقت التشغيل (YAML في تكوين النشر)
- **يتطلب:** إعا**ة تحميل API / **لا إعادة بناء صورة** تحديث تركيب التكوين فقط: **واجهة المستخدم**
- **يستخدم لـ:** العلامة التجارية، السمة، اللغة، لوحات معلومات Grafana، اختصارات تطبيق الويب، علامات الميزات

تكوين الواجهة الأمامية (تطوير محلي احتياطي):

- **الملف:** `OmniCRM-UI/.env` (انظر `.env.example`)
- **التنسيق:** متغيرات البيئة `VITE_*`
- **يتطلب:** إعادة تشغيل خادم تطوير واجهة المستخدم
- **يستخدم عندما:** `/crm/public/ui-config` غير متاحة؛ `runtimeConfig.js` يتراجع إلى هذه القيم

العلامة التجارية في وقت التشغيل. تم إهمال صور `omnicrm-ui` العامة Docker تقرأ صورة `VITE_*` واجهة المستخدم الخاصة بالعميل ومتغيرات البناء للنشر في الإنتاج.

تكوين الخلفية (crm_config.yaml)

يحتوي ملف `crm_config.yaml` على جميع إعدادات نظام الخلفية.

الموقع: `/OmniCRM/OmniCRM-API/crm_config.yaml`

تكوين قاعدة البيانات

```yaml
database:
  username: omnitouch
  password: omnitouch2024
  server: localhost
```

الحقول:

- `username` - اسم مستخدم قاعدة بيانات MySQL
- `password` - كلمة مرور قاعدة بيانات MySQL
- `server` - اسم مضيف خادم قاعدة البيانات أو IP (افتراضي: localhost)

اتصال قاعدة البيانات:

- اسم قاعدة البيانات مشفر كـ `omnicrm`
- المنفذ الافتراضي: 3306 (افتراضي MySQL)
- سلسلة الاتصال: `mysql+pymysql://username:password@server/omnicrm`

ملاحظة أمان: لا تقم أبداً بإدخال هذا الملف مع بيانات الاعتماد الحقيقية في التحكم في الإصدار. استخدم تكويناً خاصاً بالبيئة أو إدارة الأسرار.

ملاحظة: يجب أن يتطابق هذا مع بيانات اعتماد قاعدة البيانات في `.env`.

أنواع الخدمات

```yaml
service_types:
  - omnicharge
  - mobile
  - fixed
  - fixed-voice
  - hotspot
  - dongle
```

الغرض: يحدد أنواع الخدمات الصالحة لنشر الخاص بك.

تستخدم هذه الأنواع في جميع أنحاء النظام لـ:

- تصنيف المنتجات
- تصفية الإضافات (تتطابق الإضافات مع أنواع الخدمات)
- سير العمل في التزويد
- التقارير والتحليلات

أضف أنواع خدمات مخصصة هنا لحالات الاستخدام الخاصة بك.

تكامل HSS (خادم المشترك المنزلي)

لمشغلي الشبكات المحمولة مع تكامل HSS:

```yaml
hss:
  hss_peers:
    - 'http://10.179.2.140:8080'
  apn_list: "1,2,3,4,5,6"
```

التكوين:

- `hss_peers` - قائمة نقاط نهاية HSS لتزويد المشتركين
- `apn_list` - قائمة مفصولة بفواصل من معرفات APN المتاحة للتزويد

يستخدم لـ: تزويد الشبكة المحمولة ومصادقة المشتركين.

Mailjet تكوين قالب البريد الإلكتروني

يستخدم OmniCRM Mailjet للبريد الإلكتروني المعاملاتي. يحتوي كل نوع بريد إلكتروني على تكوين قالب خاص به:

```yaml
mailjet:
  api_key: your_mailjet_api_key
  api_secret: your_mailjet_api_secret

  # بريد الترحيب بالعميل
  api_crmCommunicationCustomerWelcome:
    from_email: "support@yourcompany.com"
    from_name: "Your Company Support"
    template_id: 5977509
    subject: "Welcome to YourCompany"

  # بريد فاتورة العميل
  api_crmCommunicationCustomerInvoice:
    from_email: "billing@yourcompany.com"
    from_name: "Your Company Billing"
    template_id: 6759851
    subject: "Your Invoice - "

  # تذكير بالفاتورة
  api_crmCommunicationCustomerInvoiceReminder:
    from_email: "billing@yourcompany.com"
    from_name: "Your Company Billing"
    template_id: 5977570
    subject: "Invoice Payment Reminder"

  # بريد الترحيب بالمستخدم (موظف/مدير)
  api_crmCommunicationUserWelcome:
    from_email: "admin@yourcompany.com"
    from_name: "Your Company Admin"
    template_id: 6118112
    subject: "Welcome to the Team"

  # طلب إعادة تعيين كلمة المرور
  api_crmCommunicationUserPasswordReset:
    from_email: "security@yourcompany.com"
    from_name: "Your Company Security"
    template_id: 6735666
    subject: "Password Reset Request"

  # تأكيد نجاح إعادة تعيين كلمة المرور
  api_crmCommunicationUserPasswordResetSuccess:
    from_email: "security@yourcompany.com"
    from_name: "Your Company Security"
```

```
    template_id: 6118378
    subject: "Password Reset Successful"

  # إشعار تغيير كلمة المرور
  api_crmCommunicationUserPasswordChange:
    from_email: "security@yourcompany.com"
    from_name: "Your Company Security"
    template_id: 6118423
    subject: "Password Changed"

  # تحقق من البريد الإلكتروني
  api_crmCommunicationEmailVerification:
    from_email: "verify@yourcompany.com"
    from_name: "Your Company Verification"
    template_id: 6267350
    subject: "Verify Your Email Address"

  # إشعار انتهاء الرصيد
  api_crmCommunicationsBalanceExpired:
    from_email: "alerts@yourcompany.com"
    from_name: "Your Company Alerts"
    template_id: 7238252
    subject: "Service Balance Expired"

  # تحذير من انخفاض الرصيد
  api_crmCommunicationsBalanceLow:
    from_email: "alerts@yourcompany.com"
    from_name: "Your Company Alerts"
    template_id: 7238263
    subject: "Low Balance Warning"

  # بريد تسليم eSIM
  # قم بتعيين template_id إلى قالب Mailjet مخصص، صالح لاستخدام قالب
  # المدمج HTML أو اتركه كـ 0 لاستخدام البريد الإلكتروني
  api_crmCommunicationEsimDelivery:
    from_email: "support@yourcompany.com"
    from_name: "Your Company Support"
    template_id: 0
    subject: "Your eSIM is Ready"
```

بريد تسليم eSIM

يتم إرساله عندما يتم تزويد eSIM ويكون جاهزاً للتفعيل.

```
api_crmCommunicationEsimDelivery:
  from_email: "support@yourcompany.com"
  from_name: "Your Company Support"
  template_id: 0
  subject: "Your eSIM is Ready"
```

متى يتم الإرسال:

- أثناء التسجيل الذاتي eSIM تم تزويد
- لخدمة العملاء eSIM تم تعيين

متغيرات القالب المتاحة (قالب Mailjet عند استخدام):

- `{{ var:customer_name }}` - الاسم الكامل للعميل
- `{{ var:iccid }}` - ICCID لـ eSIM
- `{{ var:activation_code }}` - سلسلة رمز تفعيل LPA
- `{{ var:has_qr_code }}` - قيمة منطقية، صحيحة إذا تم إرفاق صورة رمز QR
- `{{ var:has_activation_code }}` - قيمة منطقية، صحيحة إذا كان رمز التفعيل متاحاً

الاحتياطي المدمج:

عندما يتم تعيين `template_id` إلى `0`، يرسل OmniCRM بريدًا إلكترونيًا HTML مدمجًا يحمل eSIM الخاص بـ QR علامة تجارية مع رمز مضمنًا. يتم التحكم في العلامة التجارية لهذا البريد الإلكتروني بواسطة قسم `branding` (انظر أدناه).

العلامة التجارية (قوالب البريد الإلكتروني المدمجة)

عندما يرسل OmniCRM رسائل بريد إلكتروني باستخدام قوالب HTML المدمجة الخاصة به (بدلاً من قوالب Mailjet)، يتحكم قسم `branding` في المظهر المرئي.

```
branding:
  brand_name: 'YourCompany'
  primary_dark: '#594712'
  dark_color: '#32280B'
  accent_bg: '#f5f0e0'
```

الحقول:

المفتاح	الوصف	الافتراضي	المثال
`brand_name`	اسم الشركة المعروض في موضوع البريد الإلكتروني، والتحية، والتوقيع	مستمد من `from_name`	`'Example Mobile'`
`primary_dark`	اللون الرئيسي للعلامة التجارية المستخدم في العناوين وحدود التمييز	`#594712`	`'#1a73e8'`
`dark_color`	اللون الداكن المستخدم في تدرج رأس البريد الإلكتروني	`#32280B`	`'#0d47a1'`
`accent_bg`	لون الخلفية الفاتح لصناديق المعلومات	`#f5f0e0`	`'#e3f2fd'`

يستخدم حالياً بواسطة:

- بريد تسليم eSIM (عندما يكون `api_crmCommunicationEsimDelivery.template_id` هو `0`)

إنشاء قوالب Mailjet:

1. تسجيل الدخول إلى لوحة معلومات Mailjet (<https://app.mailjet.com>)
2. الانتقال إلى **Transactional** ← **Templates**
3. إنشاء قالب جديد أو استنساخ قالب موجود
4. ملاحظة **معرف القالب** (قيمة عددية)
5. إضافة متغيرات القالب التي تتطابق مع هيكل بيانات OmniCRM
6. تحديث `crm_config.yaml` بمعرف القالب

متغيرات القالب المتاحة:

يتلقى كل نوع بريد إلكتروني متغيرات محددة. أمثلة شائعة:

- `{{customer_name}}` - اسم العميل أو المستخدم
- `{{service_name}}` - اسم الخدمة أو المنتج

- `{{invoice_id}}` - رقم الفاتورة
- `{{invoice_amount}}` - إجمالي مبلغ الفاتورة
- `{{due_date}}` - تاريخ استحقاق الدفع
- `{{reset_link}}` - رابط إعادة تعيين كلمة المرور
- `{{verification_link}}` - رابط تحقق البريد الإلكتروني
- `{{balance}}` - الرصيد الحالي
- `{{expiry_date}}` - تاريخ انتهاء الرصيد أو الخدمة

تكوين التزويد

```
provisioning:
  failure_list: ['admin@yourcompany.com', 'ops@yourcompany.com']
```

الغرض:

- `failure_list` - عناوين البريد الإلكتروني التي يتم إخطارها عند ��شل تزويد Ansible
- تشمل الإشعارات اسم البرنامج النصي، تفاصيل الخطأ، ومعلومات العميل
- يسمح لفريق العمليات بالاستجابة بسرعة لمشاكل التزويد

تكوين الفاتورة

```
invoice:
  template_filename: 'yourcompany_invoice_template.html'
```

الغرض:

يحدد قالب HTML Jinja2 الذي يجب استخدامه لتوليد الفواتير PDF.

موقع القالب: `/OmniCRM-API/invoice_templates/`

انظر قسم **توليد** **PDF** أدناه لمزيد من التفاصيل حول إنشاء قوالب مخصصة **الفاتورة**.

عنوان URL الأساسي لـ CRM

```yaml
crm:
  base_url: 'http://localhost:5000'
```

الغرض:

- يستخدم بواسطة برامج Ansible لإجراء استدعاءات API
- يستخدم في قوالب البريد الإلكتروني لإنشاء روابط إلى CRM
- يجب أن يكون عنوان URL القابل للوصول علنًا لـ API الخاص بك

أمثلة:

- التطوير: `http://localhost:5000`
- الإنتاج: `https://api.yourcompany.com`

مهم:

- لا تقم بتضمين شرطة مائلة في النهاية
- استخدم عنوان URL القابل للوصول علنًا إذا كانت البرامج النصية تعمل على خوادم مختلفة
- تحديث عند النشر في الإنتاج

تكوين OCS و CGRates

```yaml
ocs:
  ocsApi: 'http://10.179.2.142:8080/api'
  ocsTenant: 'mnc001.mcc001.3gppnetwork.org'
  cgrates: 'localhost:2080'
```

التكوين:

- `ocsApi` - نقطة نهاية API لـ OCS لإدارة المشتركين
- `ocsTenant` - معرف المستأجر لنشر متعدد المستأجرين OCS
- `cgrates` - نقطة نهاية API JSON-RPC لـ CGRates (المضيف:المنفذ)

يستخدم لـ: الشحن في الوقت الحقيقي، إدارة الرصيد، تتبع الاستخدام.

تكوين SMSC (بوابة SMS)

```yaml
smsc:
  source_msisdn: 'YourCompany'
  smsc_url: 'http://10.179.2.216/SMSc/'
  api_key: 'your_smsc_api_key'
```

الغرض:

- إرسال إشعارات SMS للعملاء (2FA رصيد منخفض، تنبيهات الخدمة، رموز)
- `source_msisdn` - معرف المرسل المعروض للمستلمين (أحرف أو رقم هاتف)
- `smsc_url` - نقطة نهاية API لبوابة SMSC
- `api_key` - مصادقة API لـ SMSC

تكوين بث الخلايا

```yaml
cbc_url: 'http://10.179.1.113:8080'
```

الغرض: نقطة نهاية API لمركز بث الخلايا (CBC) لتنبيهات الطوارئ.

انظر `features_cell_broadcast` لتفاصيل الاستخدام.

مفتاح السر JWT

```yaml
jwt_secret: 'CHANGE_ME_ON_DEPLOYMENT'  # تم إنشاؤه تلقائيًا بواسطة Ansible
```

الأمان:

- يستخدم لتوقيع والتحقق من رموز المصادقة (JWT)
- **حرج لأمان المصادقة** - تعامل معه مثل كلمة مرور
- يجب أن يكون لكل نشر/بيئة مفتاح فريد
- لا تشارك أو تلتزم بالتحكم في الإصدار

إدارة السر تلقائيًا (نشر Ansible):

عند النشر باستخدام Ansible، يتم إدارة سر JWT تلقائيًا:

1. **أول نشر:** يقوم Ansible بإنشاء سلسلة سداسية عشوائية بطول 64 حرفًا (256 بت من الانتروبيا)
2. **النشرات اللاحقة:** يحتفظ Ansible بالسر الموجود من الخادم
3. **يحصل كل خادم على سر فريد خاص به** - الجلسات لا تعمل عبر البيئات

هذا يضمن:

- عدم وجود أسرار مشفرة في مستودعات git
- كل بيئة معزولة تشفيرياً
- تستمر الأسرار عبر النشرات (لا يتم إبطال الجلسات عن إعادة النشر)
- إعادة بناء خادم يولد سرًا جديدًا (يتم إبطال الجلسات بشكل صحيح)

تشغيل غير Ansible): توليد السر يدويًا

إذا كنت تنشر بدون Ansible، يجب عليك توليد سر فريد يدويًا:

```
# توليد مفتاح عشوائي آمن تشفيرياً
python3 -c "import secrets; print(secrets.token_hex(32))"
# أو
openssl rand -hex 32
```

ثم تحديث `crm_config.yaml`:

```
jwt_secret: 'your_generated_secret_here'
```

مهم:

- **لا تستخدم القيمة الافتراضية** `CHANGE_ME_ON_DEPLOYMENT` في الإنتاج
- **لا تشارك علنًا** - أي شخص لديه السر يمكنه تزوير رموز المصادقة
- تغيير هذا يبطل جميع الجلسات الحالية للمستخدمين (يجب على المستخدمين إعادة تسجيل الدخول)
- يجب أن تحتوي كل بيئة (تطوير، اختبار، إنتاج) على أسرار مختلفة

تكوين الدفع Stripe

```
stripe:
  secret_key: 'sk_live_xxxxxxxxxx'
  publishable_key: 'pk_live_xxxxxxxxxx'
  currency: 'aud'
  statement_descriptor_suffix: 'YOURCOMPANY'
```

التكوين:

- `secret_key` - مفتاح API السري لـ Stripe (من جانب الخادم، احتفظ به سريًا)
- `publishable_key` - مفتاح Stripe القابل للنشر (من جانب العميل، آمن للكشف)
- `currency` - رمز العملة ISO 4217 (aud، usd، gbp، eur، إلخ.)
- `statement_descriptor_suffix` - يظهر على بيانات بطاقات الائتمان للعملاء

أنواع المفاتيح:

- مفاتيح الاختبار: `sk_test_...` و `pk_test_...` (للتطوير)
- مفاتيح الإنتاج: `sk_live_...` و `pk_live_...` (للإنتاج)

استخدام وصف البيان:

- يظهر على بيانات العملاء المصرفية كـ "YOURCOMPANY"
- الحد الأقصى 22 حرفًا
- يساعد العملاء في التعرف على الرسوم
- يستخدم أيضًا في أسماء ملفات PDF للفواتير (مثل `YOURCOMPANY_12345.pdf`)

انظر *تكاملات بائعي الدفع* للحصول على تفاصيل الإعداد.

مفاتيح API وتحديد IP المسموح

حدد مفاتيح API وقيود IP مع وصول قائم على الأدوار:

```
api_keys:
  "YOUR_API_KEY_1":
    roles: ["admin"]
    ips: ["127.0.0.1", "::1"]
  "YOUR_API_KEY_2":
    roles: ["customer_service_agent_1"]
    ips: ["127.0.0.1", "::1", "10.0.1.0/24"]

# قائمة IP المسموح بها (بدون مفتاح API، مستقلة)
ip_whitelist:
  "10.179.2.142":
    roles: ["admin"]
```

الغرض:

- السماح للأنظمة الخارجية بالمصادقة عبر مفتاح API
- تقييد الوصول حسب عنوان IP
- منح أدوار محددة لمستهلكي API
- مفيد للتكاملات (أنظمة الفوترة، المراقبة، الأتمتة)

توليد مفاتيح API:

```
openssl rand -base64 48
```

الأدوار:

- `admin` - وصول كامل إلى جميع نقاط النهاية
- أدوار مخصصة محددة في نظام RBAC

أفضل ممارسات الأمان:

- استخدم مفاتيح API طويلة وعشوائية (حد أدنى 32 حرفًا)
- قيد IPs إلى مصادر معروفة فقط
- منح الحد الأدنى من الأدوار اللازمة
- بانتظام تدوير مفاتيح API
- مراقبة استخدام مفاتيح API في السجلات

تحذير أمني:

- استخدم قائمة IP المسموح بها فقط للشبكات الداخلية الموثوقة
- يجب ألا تستخدم IPs localhost (127.0.0.1، ::1)
- استخدم مفاتيح API بدلاً من ذلك للوصول الخارجي
- اعتبر قواعد جد��ر الحماية كحماية إضافية

انظر `administration_api_keys` و `concepts_api` للحصول على التفاصيل.

تكوين واجهة المستخدم في وقت التشغيل (`ui:` في crm_config.yaml)

العامة. يتم تحميل العلامة التجارية، `omnicrm-ui` تستخدم عمليات النشر في الإنتاج صورة السمة، وعلامات الميزات في وقت التشغيل من API:

- **نقطة النهاية:** `GET /crm/public/ui-config` (مدعوم من ETag واجهة؛ تخزن المستخدم في `localStorage`)
- **الأصول:** الشعارات، الرموز المفضلة، وأيقونات تطبيق الويب تحت `ui.assets_dir`، يتم تقديمها من `/crm/public/assets/...`
- **القديمة:** كتلة `branding:` لا تزال مستخدمة لقوالب البريد الإلكتروني؛ هما المصدر الحقيقي لواجهة المستخدم `ui.theme` و `ui.branding`

مثال على كتلة `ui:` (انظر أيضًا `OmniCRM-API/crm_config.yaml`):

```yaml
ui:
  version: '1'
  assets_dir: '/etc/omnicrm/deployment_assets'
  branding:
    client_name: 'Example Mobile'
    portal_name: 'Omnitouch'
    self_care_name: 'Example Mobile'
    logos:
      sm: logos/logoSm.png
      dark: logos/logoDark.png
      light: logos/logoLight.png
      favicon: logos/favicon.ico
  theme:
    primary: '#0D0629'
    secondary: '#CBB42B'
    font_family: Quicksand
  locale:
    default_language: en
    locale: en-GB
    default_country: Australia
  integrations:
    google_maps_api_key: ''
    google_tag_id: ''
    grafana:
      dashboard_ids: 'id1,id2'
      dashboard_labels: 'Ops,Finance'
  features:
    cbc_mode: no
    simplified_addresses: no
  links:
    faqs_url: 'https://example.com/faqs'
    support_url: 'https://example.com/support'
  web_apps:
    - name: GitHub
      url: https://github.com
      icon_path: webapp_icons/github.png
```

تُشحن شعارات Omnitouch **مع حزمة** الافتراضية `omnicrm-api` تحت `/usr/share/omnicrm/deployment_assets/logos/`. عمليات النشر بدون تغطية `deployment_assets` للعميل لا تزال تقدم العلامة التجارية من تلك الافتراضات. يتم مزامنة Ansible (يستنسخ `/etc/omnicrm/deployment_assets` التعديلات الخاصة بالعميل إلى `group_vars/deployment_assets/` و/أو `logo_*` ويشير إلى المسارات في)

`ui.branding.logos`. قم بتعيين `ui.assets_dir: '/etc/omnicrm/deployment_assets'`، وليس `/etc/omnicrm` نفسها. إذا كانت تلك المسار مفقودة (Docker)، فإن API تنتقل إلى `UI_ASSETS_DIR` ثم `/app/deployment_assets`. تخدم `GET /crm/public/assets/` فقط ملفات الصور تحت `logos/`، `login/`، و `webapp_icons/`.

الهجرة من `VITE_*`: قم بتعيين كل متغير بناء سابق إلى حقل `ui:` المطابق (انظر تعليقات `.env.example`). لا تزال `npm run dev` المحلية تقرأ `VITE_*` عبر `runtimeConfig.js` غير قابلة للوصول API عندما يكون نقطة النهاية.

تطوير (.env): تكوين الواجهة الأمامية محلي احتياطي

الموقع: `/OmniCRM/OmniCRM-UI/.env`

مفاتيح API والخدمات الخارجية

```
# Google Maps API (لإكمال العنوان والتشفير الجغرافي)
VITE_GOOGLE_API_KEY=your_google_api_key

# تعطيل تشغيل المتصفح التلقائي عند بدء npm
BROWSER=none
```

تكوين بائع الدفع

```
# بائعي الدفع الممكّنين (مفصولة بفواصل): stripe، paypal، أو كلاهما
VITE_PAYMENT_VENDORS="stripe,paypal"

# مفتاح Stripe القابل للنشر (مطلوب إذا تم تمكين Stripe)
VITE_STRIPE_PUBLISHABLE_KEY=pk_test_xxxxx

# معرف عميل PayPal (مطلوب إذا تم تمكين PayPal)
VITE_PAYPAL_CLIENT_ID=YOUR_PAYPAL_CLIENT_ID
```

يجب أن تتطابق مع الخلفية:

- يجب أن يتطابق `VITE_STRIPE_PUBLISHABLE_KEY` مع `stripe.publishable_key` في `crm_config.yaml`
- يجب أن يتطابق `VITE_PAYPAL_CLIENT_ID` مع `paypal.client_id` في `crm_config.yaml`

إعادة بناء وإعادة تشغيل `VITE_*` **مهم:** هذه هي **متغيرات وقت البناء**. يتطلب تغيير أي قيمة واجهة المستخدم. انظر دليل نظام الدفع - تكوين البائع لإجراءات الإعداد الكاملة بما في ذلك تكوين الخلفية.

العلامة التجارية ومعلومات الشركة

```
# العلامة التجارية للشركة
VITE_COMPANY_NAME="ShellFone"
VITE_PORTAL_NAME="ShellManager"
VITE_SELF_CARE_NAME="ShellCare"
VITE_COMPANY_TAGLINE="Phones with Shells"
```

تظهر هذه القيم في جميع أنحاء واجهة المستخدم:

- `COMPANY_NAME` - يظهر في عناوين الصفحات، والبريد الإلكتروني، والتواصل مع العملاء
- `PORTAL_NAME` - اسم بوابة الإدارة/الموظفين (مثل "ShellManager")
- `SELF_CARE_NAME` - اسم بوابة الخدمة الذاتية للعملاء (مثل "ShellCare")
- `COMPANY_TAGLINE` - يظهر في شاشات تسجيل الدخول والمواد التسويقية

الإعدادات الإقليمية والتوطين

```
# اللغة واللغة المحلية
# اللغات المدعومة: ar، ch، en، fr، gr، it، ru، sp
VITE_DEFAULT_LANGUAGE=en
VITE_LOCALE="en-GB"

# الموقع الافتراضي (لإكمال العنوان)
VITE_DEFAULT_LOCATION="Sydney, Australia"
VITE_DEFAULT_COUNTRY="Australia"

# إعدادات العملة
VITE_CURRENCY_CODE="GBP"
VITE_CURRENCY_SYMBOL="£"
```

اللغات المدعومة:

- `ar` - العربية
- `ch` - الصينية
- `en` - الإنجليزية (افتراضي)
- `fr` - الفرنسية
- `gr` - اليونانية
- `it` - الإيطالية
- `ru` - الروسية
- `sp` - الإسبانية

العملات الشائعة:

- USD - $ (دولار أمريكي)
- GBP - £ (جنيه إسترليني)
- EUR - € (يورو)
- AUD - $ (دولار أسترالي)
- CAD - $ (دولار كندي)

ملاحظة: يجب أن تتطابق مع `stripe.currency` في `crm_config.yaml`.

تكوين سمة اللون

```
# (اللون الأساسي (اللون الرئيسي للعلامة التجارية
VITE_PRIMARY_COLOR=#405189

# (خيارات الألوان الإضافية (أمثلة معلق عليها
# VITE_SECONDARY_COLOR=#2bFFcf
# VITE_TERTIARY_COLOR=#1a9fbf
# VITE_SUCCESS_COLOR=#28a745
# VITE_INFO_COLOR=#17a2b8
# VITE_WARNING_COLOR=#ffc107
# VITE_DANGER_COLOR=#dc3545
```

الألوان المتاحة:

- `VITE_PRIMARY_COLOR` - اللون الرئيسي للعلامة التجارية (الأزرار، الروابط، التمييزات)
- `VITE_SECONDARY_COLOR` - لون التمييز
- `VITE_TERTIARY_COLOR` - تمييز إضافي
- `VITE_SUCCESS_COLOR` - رسائل النجاح (#28a745)
- `VITE_INFO_COLOR` - رسائل المعلومات (#17a2b8)
- `VITE_WARNING_COLOR` - التحذيرات (#ffc107)
- `VITE_DANGER_COLOR` - الأخطاء (#dc3545)
- `VITE_LIGHT_COLOR` - الخلفيات الفاتحة (#f8f9fa)
- `VITE_DARK_COLOR` - النص الداكن (#343a40)
- `VITE_PRIMARY_DARK_COLOR` - النسخة الداكنة من اللون الأساسي

التنسيق: رموز الألوان السداسية (#RRGGBB)

يتم تطبيق اللون الأساسي على:

- رؤوس التنقل
- أزرار العمل
- الروابط والتمييزات
- الحالات النشطة
- عناصر العلامة التجارية

تكوين الخط

```
# تكوين عائلة الخط
# Sans-Serif: Inter، Roboto، Open Sans، Lato، Quicksand، Poppins،
Nunito، Montserrat، Work Sans، Source Sans Pro، Raleway، Ubuntu،
Josefin Sans، HKGrotesk
# Serif: Merriweather، Lora، Playfair Display
# System: System (خطوط الجهاز الأصلية)
# الافتراضي: Quicksand
VITE_FONT_FAMILY=Quicksand
```

مهم: جميع الخطوط **مستضافة محليًا** داخل تطبيق OmniCRM-UI. هذا يعني:

- **لا تحميل خطوط خارجية** - يتم تضمين الخطوط مع التطبيق
- **متوافقة مع الحدائق المسورة** - لا حاجة للوصول إلى الإنترنت لتعمل الخطوط
- **تشغيل غير متصل** - وظيفة كاملة في بيئات الشبكة المعزولة أو المقيدة
- **الخصوصية** - لا طلبات خارجية إلى Google Fonts، Adobe Fonts، أو CDNs أخرى
- **الأداء** - تحميل أسرع بدون اعتمادات خارجية
- **الأمان** - لا تتبع من طرف ثالث أو تسرب بيانات من خلال طلبات الخطوط

الخيارات المتاحة:

خطوط Sans-Serif:

- Inter، Roboto، Open Sans، Lato، Quicksand (افتراضي)، Poppins، Nunito، Montserrat، Work Sans، Source Sans Pro، Raleway، Ubuntu، Josefin Sans، HKGrotesk

خطوط Serif:

- Merriweather، Lora، Playfair Display

خطوط النظام:

- System - يستخدم خطوط الجهاز الأصلية لأفضل أداء وأصغر حجم حزمة

إضافة خطوط مخصصة:

نعم، يمكنك إضافة خطوط إضافية! يتم تخزين جميع الخطوط محليًا في التطبيق.

لإضافة خط مخصص جديد:

1. **أضف ملفات الخط** إلى `OmniCRM-UI/src/assets/fonts/your-font-name/`
    - استخدم تنسيق WOFF2 لأفضل ضغط ودعم المتصفح
    - قم بتضمين أوزان متعددة (300، 400، 500، 600، 700) للتقديم الصحيح
    - اسم الملفات: `your-font-name-300.woff2`، `your-font-name-400.woff2`، إلخ.
2. **حدد قواعد @font-face** في `OmniCRM-UI/src/assets/scss/fonts/_google-fonts.scss`

```
//
// خطك المخصص - الوصف
//

@font-face {
  font-family: 'Your Font Name';
  font-style: normal;
  font-weight: 400;
  font-display: swap;
  src: url("../../fonts/your-font-name/your-font-name-400.woff2") format('woff2');
}

@font-face {
  font-family: 'Your Font Name';
  font-style: normal;
  font-weight: 700;
  font-display: swap;
  src: url("../../fonts/your-font-name/your-font-name-700.woff2") format('woff2');
}
```

3. **تعيين في ملف .env:**

```
VITE_FONT_FAMILY=Your Font Name
```

إرشادات وزن الخط:

- 300 - (اختياري، للعناوين الدقيقة) خفيف
- 400 - (مطلوب، نص افتراضي) عادي
- 500 - (اختياري، للتأكيد) متوسط
- 600 - (اختياري، العناوين الفرعية) شبه غامق
- 700 - (مطلوب، العناوين والنص القوي) غامق

روابط سريعة لتطبيقات الويب

قم بتكوين ما يصل إلى 6 تطبيقات ويب للوصول السريع تظهر في لوحة معلومات الإدارة:

```
# تطبيق الويب 1: GitHub
VITE_WEB_APP_1_NAME="GitHub"
VITE_WEB_APP_1_URL="https://github.com"
VITE_WEB_APP_1_ICON_PATH="resources/webapp_icons/github.png"

# تطبيق الويب 2: Xero
VITE_WEB_APP_2_NAME="Xero"
VITE_WEB_APP_2_URL="https://go.xero.com/"
VITE_WEB_APP_2_ICON_PATH="resources/webapp_icons/xero.png"

# تطبيق الويب 3-6: تكاملات إضافية
# (قم بتكوينها بشكل مشابه مع NAME و URL و ICON_PATH)
```

النمط:

- `VITE_WEB_APP_N_NAME` - اسم العرض
- `VITE_WEB_APP_N_URL` - عنوان URL المستهدف
- `VITE_WEB_APP_N_ICON_PATH` - مسار ملف الأيقونة (نسبي إلى public/)

أيقونات المثال: GitHub، Xero، Monday.com، Gmail، MailJet، Slack

تكامل لوحة معلومات Grafana

يتكامل OmniCRM مع Grafana لتوفير لوحات معلومات التحليلات. يمكن للمستخدمين الذين لديهم إذن `grafana_access` الوصول إلى Grafana عند `/grafana` لإنشاء لوحات معلومات مخصصة، ويمكن تضمين تلك اللوحات في واجهة OmniCRM.

```
# تكامل لوحة معلومات Grafana
# معرفات اللوحات (مفصولة بفواصل) - احصل على هذه من
/grafana/d/ID/name URLs
VITE_GRAFANA_DASHBOARD_IDS=abc123,def456,ghi789

# تسميات اللوحات (مفصولة بفواصل، يجب أن تتطابق مع ترتيب المعرفات)
VITE_GRAFANA_DASHBOARD_LABELS=System Metrics,User
Analytics,Network Performance
```

الحقول:

- `VITE_GRAFANA_DASHBOARD_IDS` - قائمة مفصولة بفواصل من معرفات لوحات معلومات Grafana من مسار URL
- `VITE_GRAFANA_DASHBOARD_LABELS` - قائمة مفصولة بفواصل من أسماء عرض اللوحات (يجب أن تتطابق مع ترتيب المعرفات)

كيفية العثور على معرفات اللوحات:

عند عرض لوحة معلومات في Grafana، سيبدو عنوان URL مثل `/grafana/d/abc123/dashboard-name`. معرف اللوحة هو الجزء بعد `/d/` (وقبل الاسم في هذا المثال، `abc123`).

الاستخدام:

بمجرد التكوين، تظهر اللوحات تلقائيًا في الشريط الجانبي لـ OmniCRM تحت قائمة "لوحات المعلومات". يمكن للمستخدمين النقر لعرض اللوحات المضمنة مع الحفاظ على التنقل في OmniCRM مرئيًا.

مثال:

إذا كان عنوان URL للوحة المعلومات لديك هو `http://your-server/grafana/d/system-overview-123/system-metrics`، إذن:

- معرف اللوحة: `system-overview-123`
- تسمية اللوحة: اختر أي اسم ودود مثل "نظرة عامة على النظام"

مهم:

- يجب أن يتطابق عدد المعرفات مع عدد التسميات

- تظهر اللوحات في الشريط الجانبي بالترتيب الذي تم تحديده
- عرض اللوحات `grafana_access` يمكن فقط للمستخدمين الذين لديهم إذن
- تتطلب التغييرات إعادة بناء واجهة المستخدم الأمامية

روابط الدعم والوثائق

```
# روابط الأسئلة الشائعة والدعم
VITE_FAQS_URL=https://support.yourcompany.com/faqs
VITE_SUPPORT_URL=https://support.yourcompany.com
```

الغرض: روابط لموارد الدعم الخارجية تظهر في واجهة المستخدم.

تسجيل الدخول الاجتماعي

```
# السماح بتسجيل الدخول الاجتماعي (نعم|لا)
VITE_ALLOW_SOCIAL_LOGINS=yes
```

الخيارات:

- `نعم` - تمكين أزرار تسجيل الدخول الاجتماعي (Google، Facebook، إلخ.)
- `لا` - تعطيل تسجيل الدخول الاجتماعي

ملاحظة: يجب تكوين مزودي تسجيل الدخول الاجتماعي بشكل منفصل.

التحليلات والتتبع

```
# Google Analytics 4 أو Google Tag Manager (اختياري)
# لـ GA4، استخدم التنسيق: G-XXXXXXXXXX
# لـ GTM، استخدم التنسيق: GTM-XXXXXXX
VITE_GOOGLE_TAG_ID=G-XXXXXXXXXX
```

الغرض: تمكين تتبع Google Tag Manager (GTM) أو Google Analytics 4 (GA4) لمراقبة سلوك المستخدم واستخدام التطبيق.

التنسيقات المدعومة:

- **معرف قياس GA4:** `G-XXXXXXXXXX` لتتبع Google Analytics 4
- **معرف حاوية GTM:** `GTM-XXXXXXX` لـ Google Tag Manager
- **معطل:** اتركه فارغًا أو احذفه لتعطيل التتبع

كيف يعمل:

يكتشف النظام خدمة التتبع تلقائيًا بناءً على تنسيق المعرف:

- المعرفات التي تبدأ بـ `G-` تقوم بتهيئة Google Analytics 4
- المعرفات التي تبدأ بـ `GTM-` تقوم بتهيئة Google Tag Manager
- القيمة الفارغة أو المفقودة تعطل جميع التتبع

الميزات:

- **تتبع الصفحات التلقائي** - يتم تتبع كل تغيير في المسار كعرض صفحة
- **تتبع الأحداث المخصصة** - تتبع تفاعلات المستخدم، نقرات الأزرار، تقديم النماذج
- **لا تغييرات في التعليمات البرمجية مطلوبة** - فقط قم بتعيين متغير البيئة وأعد بناء واجهة المستخدم

تعليمات الإعداد:

1. **احصل على معرف التتبع الخاص بك:**

   لـ Google Analytics 4:

   - تسجيل الدخول إلى Google Analytics
   - إنشاء أو اختيار خاصية GA4
   - الانتقال إلى **الإدارة** → **تدفقات البيانات**
   - تحديد تدفق الويب الخاص بك
   - نسخ **معرف القياس** (التنسيق: `G-XXXXXXXXXX`)

   لـ Google Tag Manager:

   - تسجيل الدخول إلى Google Tag Manager
   - إنشاء أو اختيار حاوية
   - نسخ **معرف الحاوية** (التنسيق: `GTM-XXXXXXX`)

2. **أضف إلى ملف .env:**

```
VITE_GOOGLE_TAG_ID=G-XXXXXXXXXX
```

3. **إعادة بناء واجهة المستخدم:**

```
cd OmniCRM-UI
npm run build
```

4. **تحقق من التتبع:**

   - افتح تطبيقك في متصفح
   - تحقق من وحدة التحكم في المتصفح لرسائل التهيئة
   - استخدم تقارير Google Analytics في الوقت الحقيقي لرؤية المستخدمين النشطين

الاستخدام المتقدم - تتبع الأحداث المخصصة:

يوفر التكامل وظائف مساعدة لتتبع الأحداث المخصصة في جميع أنحاء تطبيقك.

مثال - تتبع نقرات الأزرار:

```
import { trackEvent } from './utils/googleAnalytics';

// في مكونك
const handleCheckout = () => {
  trackEvent('checkout_initiated', {
    cart_value: 99.99,
    item_count: 3,
    page: '/cart'
  });
  // تابع مع منط الخروج
};
```

مثال - تتبع تقديم النماذج:

```
import { trackEvent } from './utils/googleAnalytics';

const handleFormSubmit = (formType) => {
  trackEvent('form_submission', {
    form_type: formType,
    timestamp: new Date().toISOString()
  });
};
```

مثال - تتبع مشاهدات الصفحات المخصصة:

```
import { trackPageView } from './utils/googleAnalytics';

// تتبع عرض صفحة يدويًا (مفيد لنوافذ الحوار، علامات التبويب، إلخ.)
trackPageView('/virtual/page-path', 'عنوان الصفحة');
```

وظائف التتبع المتاحة:

الوظيفة	الغرض	المعلمات
`initializeGoogleTracking()`	تهيئة التتبع عند بدء التطبيق	لا شيء (يتم استدعاؤه تلقائيًا)
`trackEvent(eventName, eventParams)`	تتبع الأحداث المخصصة	اسم الحدث (سلسلة)، المعلمات (كائن)
`trackPageView(path, title)`	تتبع مشاهدات الصفحات يدويًا	المسار (سلسلة)، العنوان (سلسلة)

الخصوصية والامتثال:

يجمع تتبع Google Analytics بيانات سلوك المستخدم. تأكد من الامتثال للوائح حماية البيانات:

- **GDPR (أوروبا)** - احصل على موافقة المستخدم قبل التتبع، وقدم آلية إلغاء الاشتراك
- **CCPA (كاليفورنيا)** - السماح للمستخدمين بإلغاء الاشتراك في جمع البيانات
- الكشف عن استخدام Google Analytics في سياسة **سياسة الخصوصية** الخاصة بك

- **موافقة ملفات تعريف الارتباط** - تنفيذ لافتة موافقة ملفات تعريف الارتباط إذا كانت مطلوبة بموجب القوانين المحلية
- **احتفاظ البيانات** Google Analytics - تكوين فترات احتفاظ البيانات المناسبة في إعدادات

استكشاف الأخطاء:

المشكلة	السبب المحتمل	الحل
لا بيانات في لوحة معلومات GA	معرف التتبع غير صحيح	تحقق من أن تنسيق المعرف يتطابق مع GTM-XXXXXXX أو G-XXXXXXXXXX
أخطاء وحدة التحكم عند تحميل الصفحة	معرف التتبع غير صالح	تحقق من الأخطاء المطبعية في ملف .env
لا يتم تتبع مشاهدات الصفحات	واجهة المستخدم لم تُعاد بناؤها بعد تغيير .env	أعد `npm run build` قم بتشغيل تشغيل واجهة المستخدم
الأحداث لا تظهر	تم تمكين مانع الإعلانات	اختبر مع تعطيل مانع الإعلانات
البيانات المتأخرة	سلوك عادي لـ GA	استخدم تقارير الوقت الحقيقي للتحقق الفوري

اعتبارات الأداء:

- يتم تحميل برامج التتبع بشكل غير متزامن ولا تعيق عرض الصفحة
- تأثير أداء ضئيل (< 50KB حمولة إضافية)
- يتم تخزين البرامج النصية في ذاكرة التخزين المؤقت بواسطة المتصفحات بعد التحميل الأول
- لا يحدث تتبع إذا كانت `VITE_GOOGLE_TAG_ID` فارغة

تفاصيل التنفيذ:

- كود التتبع: `OmniCRM-UI/src/utils/googleAnalytics.js`
- هوك تتبع الصفحة: `OmniCRM-UI/src/utils/usePageTracking.js`
- التهيئة: تلقائية عند بدء التطبيق في `App.js` أو `index.js`

تكوين الشحن وإعادة الشحن

```
VITE_TOPUP_PRICE_PER_DAY=10
```

الغرض: يحدد السعر اليومي لخدمات الشحن/إعادة الشحن في بوابة الخدمة الذاتية.

الحقول:

- `VITE_TOPUP_PRICE_PER_DAY` - السعر اليومي لخدمات إعادة الشحن (قيمة عددية)

مثال: إذا تم تعيينه إلى `10` وكانت العملة بالدولار الأمريكي، يدفع العملاء 10 دولارات أمريكية في اليوم من الخدمة.

ملاحظة: يجب أن تتطابق هذه القيمة مع تكوين الأسعار في الخلفية. انظر `features_topup_recharge` للحصول على تفاصيل الإعداد الكاملة.

العلامة التجارية والتخصيص المرئي

يتيح لك OmniCRM استبدال صور العلامة التجارية الافتراضية بشعار شركتك وشاشات البداية دون تعديل الكود.

ملفات الشعار ونظام الاحتياطي

يتم تخزين الشعارات في `/OmniCRM-UI/src/assets/images/omnitouch/` وتستخدم نظام احتياطي:

الشعارات الافتراضية (موجودة دائمًا):

- `DefaultLogoDark.png` - شعار السمة الداكنة (يستخدم على الخلفيات الفاتحة)
- `DefaultLogoLight.png` - شعار السمة الفاتحة (يستخدم على الخلفيات الداكنة)

الشعارات المخصصة (اختياري، تأخذ الأولوية عند وجودها):

- `logoSm.png` - شعار صغير للشريط الجانبي المنهار (100 :موصى به x100px)
- `logoDark.png` - شعار داكن بالحجم الكامل للرؤوس (200 :موصى به x50px)

- `logoLight.png` - شعار فاتح بالحجم الكامل لشاشات المصادقة (موصى به: 200x50px)

كيف يعمل الاحتياطي للشعار:

يحاول النظام تحميل الشعارات المخصصة أولاً. إذا لم يكن هناك ملف شعار مخصص، فإنه يتراجع إلى الافتراضي:

```
// من Header.js
const tryImport = (filename) => {
    try {
        return require(`../assets/images/omnitouch/${filename}`);
    } catch (err) {
        return null; // يتراجع إلى الافتراضي
    }
};

const userLogoSm = tryImport("logoSm.png");
const userLogoDark = tryImport("logoDark.png");
const userLogoLight = tryImport("logoLight.png");
```

أين تظهر الشعارات:

- **logoSm.png** - الشريط الجانبي المنهار، التنقل المحمول، عروض الرأس صغيرة
- **logoDark.png** - شريط الرأس الرئيسي (وضع الضوء)، رأس لوحة المعلومات الإدارية
- **logoLight.png** - شاشات تسجيل الدخول/التسجيل، الخلفيات الداكنة، دوار المصادقة

استبدال الشعارات:

1. **إنشاء ملفات الشعار الخاصة بك:**
    - استخدم تنسيق PNG لدعم الشفافية
    - تطابق الأبعاد الموصى بها أعلاه
    - تأكد من أن الشعارات واضحة عند الدقة العادية وعالية الدقة
2. **إضافة إلى OmniCRM:**

```
# انسخ ملفات الشعار الخاصة بك إلى دليل صور omnitouch
cp /path/to/your/logoSm.png OmniCRM-
UI/src/assets/images/omnitouch/
cp /path/to/your/logoDark.png OmniCRM-
UI/src/assets/images/omnitouch/
cp /path/to/your/logoLight.png OmniCRM-
UI/src/assets/images/omnitouch/
```

3. **إعادة بناء واجهة المستخدم:**

```
cd OmniCRM-UI
npm run build
```

4. **تحقق من التغييرات:**
    - تحقق من رأس الوضع الفاتح (يجب أن يظهر `logoDark.png`)
    - تحقق من رأس الوضع الداكن (يجب أن يظهر `logoLight.png`)
    - تحقق من الشريط الجا��بي المنهار (يجب أن يظهر `logoSm.png`)
    - تحقق من شاشة تسجيل الدخول (يجب أن يظهر `logoLight.png`)

أفضل ممارسات تصميم الشعار:

- **التباين** - تأكد من أن الشعارات مرئية على كل من الخلفيات الفاتحة والداكنة
- **البساطة** - يجب أن تكون الشعارات قابلة للتعرف عليها في الأحجام الصغيرة
- **التنسيق** - استخدم PNG مع خلفيات شفافة
- **الدقة العالية** - قدم دقة 2x للشاشات عالية الدقة
- **الاتساق** - استخدم نفس ألوان العلامة التجارية عبر جميع متغيرات الشعار

شاشات البداية وخلفيات المصادقة

تستخدم شاشات المصادقة (تسجيل الدخول، التسجيل، إعادة تعيين كلمة المرور) خلفية دوارة مع صور قابلة للتخصيص.

الموقع: `/OmniCRM-UI/src/pages/AuthenticationInner/authCarousel.js`

التكوين الافتراضي:

```
import logoLight from "../../assets/images/logo-light.png";

// الشعار المعروض على شاشات المصادقة
<img src={logoLight} alt="" height="18" />
```

تخصيص شاشات المصادقة:

1. استبدل `logo-light.png` في `/OmniCRM-UI/src/assets/images/`
2. أضف CSS خلفية مخصصة إلى فئة `.auth-one-bg`
3. عدل اقتباسات الدوار في `authCarousel.js` لتتناسب مع صوت علامتك التجارية

مثال على التخصيص:

```
/* أضف إلى CSS المخصص الخاص بك */
.auth-one-bg {
    background-image: url('/assets/images/custom-auth-bg.jpg');
    background-size: cover;
    background-position: center;
}
```

تولید PDF الفاتورة

ينشئ OmniCRM فواتير PDF احترافية باستخدام قوالب HTML Jinja2 و WeasyPrint لتوليد PDF.

كيف تعمل تولید PDF

1. **اختيار القالب:**
   - يتم تحديد اسم القالب في `crm_config.yaml` تحت `invoice.template_filename`
   - يتم تحميل القالب من دليل `/OmniCRM-API/invoice_templates/`
2. **إعداد البيانات:**
   - يتم جلب بيانات الفاتورة (ID، الحالة، المبالغ، التواريخ) من قاعدة البيانات

- يتم استرداد معلومات العميل (الاسم، العنوان)
- يتم تجميع قائمة المعاملات (جميع الرسوم/الائتمانات على الفاتورة)

3. **تقديم القالب:**

- يقوم Jinja2 بتقديم قالب HTML مع بيانات الفاتورة
- يتم استبدال المتغيرات مثل `{{ invoice_number }}`، `{{ total_amount }}`، إلخ.
- يتم حفظ HTML المقدم في `invoice_templates/rendered/` لأغراض التصحيح

4. **توليد PDF:**

- يقوم WeasyPrint بتحويل HTML المقدم إلى PDF
- يدعم PDF تنسيق CSS، الرؤوس/التذييلات، فواصل الصفحات، الصور
- يتم إنشاء بيانات PDF الثنائية في الذاكرة

5. **التخزين المؤقت:**

- يتم ترميز PDF بتنسيق Base64 وتخزينه في جدول `Invoice_PDF_Cache`
- يتم حساب تجزئة SHA256 للتحقق من النزاهة
- تعيد الطلبات اللاحقة PDF (تسليم فوري) المخزن

6. **إبطال التخزين المؤقت:**

- يتم إبطال التخزين المؤقت عند تعديل الفاتورة، أو إبطالها، أو استردادها
- لا تؤدي تغييرات القالب تلقائيًا إلى إبطال التخزين المؤقت الحالي

هيكل قالب الفاتورة

تكون قوالب الفواتير ملفات HTML Jinja2 مع متغيرات مندمجة ومنطق.

موقع القالب: `/OmniCRM-API/invoice_templates/yourcompany_invoice_template.html`

المتغيرات المتاحة:

```
{
    'invoice_number': 12345,
    'date': '2025-01-04',
    'client': {
        'name': 'John Smith',
        'address': {
            'address_line_1': '123 Main St',
            'city': 'Sydney',
            'state': 'NSW',
            'zip_code': '2000',
            'country': 'Australia'
        }
    },
    'transaction_list': [
        [
            {
                'transaction_id': 1,
                'title': 'Mobile Service - Monthly Fee',
                'retail_cost': 30.00,
                'wholesale_cost': 10.00,
                'created': '2025-01-01'
            },
            {
                'transaction_id': 2,
                'title': 'Data Addon - 5GB',
                'retail_cost': 15.00,
                'wholesale_cost': 5.00,
                'created': '2025-01-15'
            }
        ]
    ],
    'total_amount': 45.00,
    'due_date': '2025-01-31',
    'start_date': '2025-01-01',
    'end_date': '2025-01-31',
    'paid': False,
    'void': False
}
```

مثال على مقتطف القالب:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Invoice {{ invoice_number }}</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 40px;
        }
        .header {
            text-align: center;
            margin-bottom: 30px;
        }
        .invoice-details {
            margin-bottom: 20px;
        }
        table {
            width: 100%;
            border-collapse: collapse;
        }
        th, td {
            border: 1px solid #ddd;
            padding: 12px;
            text-align: left;
        }
        th {
            background-color: #405189;
            color: white;
        }
        .total {
            text-align: right;
            font-size: 18px;
            font-weight: bold;
            margin-top: 20px;
        }
    </style>
</head>
<body>
    <div class="header">
        <h1>INVOICE</h1>
    </div>
```

```
    <div class="invoice-details">
        <p><strong>Invoice Number:</strong> {{ invoice_number }}
</p>
        <p><strong>Date:</strong> {{ date }}</p>
        <p><strong>Due Date:</strong> {{ due_date }}</p>
        <p><strong>Billing Period:</strong> {{ start_date }} to {{
end_date }}</p>
    </div>

    <div class="customer-details">
        <h3>Bill To:</h3>
        <p>{{ client.name }}</p>
        <p>{{ client.address.address_line_1 }}</p>
        <p>{{ client.address.city }}, {{ client.address.state }}
{{ client.address.zip_code }}</p>
        <p>{{ client.address.country }}</p>
    </div>

    <table>
        <thead>
            <tr>
                <th>Description</th>
                <th>Date</th>
                <th>Amount</th>
            </tr>
        </thead>
        <tbody>
            {% for transaction in transaction_list[0] %}
            <tr>
                <td>{{ transaction.title }}</td>
                <td>{{ transaction.created }}</td>
                <td>${{ "%.2f"|format(transaction.retail_cost) }}
</td>
            </tr>
            {% endfor %}
        </tbody>
    </table>

    <div class="total">
        <p>Total Amount Due: ${{ "%.2f"|format(total_amount) }}
</p>
    </div>

    {% if paid %}
```

```
    <div style="text-align: center; color: green; font-weight:
bold;">
        PAID
    </div>
    {% endif %}

    {% if void %}
    <div style="text-align: center; color: red; font-weight:
bold;">
        VOID
    </div>
    {% endif %}
</body>
</html>
```

أفضل ممارسات القالب:

- **استخدم مسارات مطلقة للصور** - `file:///absolute/path/to/image.png`
- **مضمّن** WeasyPrint لا يقوم - **CSS** بتحميل أوراق الأنماط الخارجية بشكل موثوق
- **اختبر مع بيانات عينة** - استخدم `invoice_templates/rendered/` لفحص HTML
- **فواصل الصفحات** - استخدم `<div style="page-break-after: always;"></div>` لفواتير متعددة الصفحات
- **الرؤوس والتذييلات** - استخدم قواعد CSS `@page` للعناصر المتكررة
- **تنسيق العملة** - استخدم مرشحات Jinja2: `{{ "%.2f"|format(amount) }}`

إنشاء قالب فاتورة مخصص

1. **انسخ قالب المثال:**

   ```
   cd /OmniCRM/OmniCRM-API/invoice_templates
   cp sample_invoice_template.html
   yourcompany_invoice_template.html
   ```

2. **تحرير القالب:**

   - استبدل اسم الشركة، الشعار، معلومات الاتصال

- اضبط التنسيق (الألوان، الخطوط، التخطيط) ليتناسب مع العلامة التجارية
- أضف أو أزل الأقسام حسب الحاجة (تفاصيل الضرائب، تعليمات الدفع، إلخ.)

3. **تحديث التكوين:**

   تحرير `crm_config.yaml`:

   ```
   invoice:
     template_filename: 'yourcompany_invoice_template.html'
   ```

4. **اختبار توليد الفاتورة:**

   - قم بإنشاء فاتورة اختبار في CRM
   - قم بتنزيل PDF والتحقق من التنسيق
   - تحقق من `invoice_templates/rendered/{invoice_id}.html` لأغراض التصحيح

5. **إبطال التخزين المؤقت القديم (إذا لزم الأمر):**

   إذا قمت بتغيير القالب وتريد إعادة توليد الفواتير الحالية:

   ```
   -- مسح جميع PDFs المخزنة (يؤدي إلى إعادة التوليد)
   DELETE FROM Invoice_PDF_Cache;
   ```

نظام التخزين المؤقت PDF

لتحسين الأداء، يقوم OmniCRM بتخزين PDFs المولدة:

سلوك التخزين المؤقت:

- **الطلب الأول** - يتم إنشاء PDF، تخزينه، وإرجاعه
- **الطلبات اللاحقة** - يتم إرجاع PDF المخزن على الفور (بدون إعادة توليد)
- **إبطال التخزين المؤقت** - يحدث عند تعديل الفاتورة، أو إبطالها، أو استردادها
- **تنظيف التخزين المؤقت** - يتم تلقائيًا تطهير التخزين المؤقت القديم بعد 30 يومًا من عدم النشاط

تخزين التخزين المؤقت:

- يتم تخزين PDF المشفر بتنسيق Base64 في جدول `Invoice_PDF_Cache`
- تجزئة SHA256 للتحقق من النزاهة
- يتضمن اسم الملف، تاريخ الإنشاء، تاريخ آخر وصول

إدارة التخزين المؤقت يدويًا:

```
# في OmniCRM API أو Python shell
from services.invoice_service import cleanup_old_pdf_cache,
invalidate_invoice_cache
from utils.db_helpers import get_db_session

session = get_db_session()

# تنظيف التخزين المؤقت الأقدم من 30 يومًا
result = cleanup_old_pdf_cache(session, days_old=30)
print(result)  # {'status': 'success', 'deleted_count': 15}

# إبطال التخزين المؤقت لفاتورة معينة
invalidate_invoice_cache(session, invoice_id='12345')
```

نقاط نهاية API:

تنزيل/توليد PDF الفاتورة:

GET /invoice/pdf/{invoice_id}

الاستجابة: تنزيل ملف PDF مع اسم الملف من وصف بيان Stripe

رؤوس التخزين المؤقت:

- الطلب الأول: استجابة أبطأ (وقت التوليد)
- الطلبات المخزنة: استجابة فورية
- يكون التخزين المؤقت hit/miss شفافًا للمستخدم

تطبيق تغييرات التكوين

الخلفية (crm_config.yaml)

1. تحرير `OmniCRM-API/crm_config.yaml`
2. حفظ التغييرات
3. إعادة تشغيل خدمة API:

```
cd OmniCRM-API
sudo systemctl restart omnicrm-api
# أو
./restart_api.sh
```

تدخل التغييرات حيز التنفيذ على الفور بعد إعادة التشغيل.

الواجهة الأمامية (.env)

1. تحرير `OmniCRM-UI/.env`
2. حفظ التغييرات
3. إعادة بناء واجهة المستخدم:

```
cd OmniCRM-UI
npm run build
```

4. إعادة تشغيل خدمة واجهة المستخدم أو خادم الويب

وضع التطوير:

أعد تشغيل خادم التطوير لتطبيق التغييرات، `npm start` أثناء التطوير باستخدام.

أفضل ممارسات التكوين

الأمان

- استخدم `.gitignore` لملفات التكوين التي تحتوي **لا تقم أبداً بإدخال الأسرار** - على بيانات الاعتماد
- **استخدم كلمات مرور قوية** - حد أدنى 16 حرفًا مع حالة مختلطة، أرقام، ورموز
- **قم بتدوير بيانات الاعتماد بانتظام** - خاصة للنشرات الإنتاجية
- **تقييد الوصول إلى قاعدة البيانات** - استخدم تحديد IP وقواعد جدار الحماية
- **استخدم تكوينات خاصة بالبيئة** - فصل تكوينات التطوير/الاختبار/الإنتاج
- **حدد أذونات مفتاح API** - منح الحد الأدنى من الأدوار اللازمة
- **استخدم تحديد IP بحذر** - يفضل استخدام مفاتيح API لأمان أفضل

الصيانة

- **وثق التغييرات** - احتفظ بسجل تغييرات تكوين التعديلات
- **نسخ احتياطي للتكوينات** - احتفظ بنسخ قبل التغييرات الكبيرة
- **اختبر في بيئة الاختبار** - تحقق من تغييرات التكوين قبل نشر الإنتاج
- **تحكم في الإصدار** - تتبع قوالب التكوين (بدون أسرار) في git

الأداء

- **استخدم قاعدة بيانات محلية** - تجنب قاعدة بيانات بعيدة لأداء أفضل
- **تكوين التخزين المؤقت** - تمكين التخزين المؤقت لـ OCS إذا كان متاحًا
- **تحسين Grafana** - تحديد عدد لوحات المعلومات المضمنة

العلامة التجارية

- **تطابق الألوان** - تأكد من أن ألوان واجهة المستخدم تكمل شعارك
- **اختبار التباين** - تحقق من قابلية قراءة النص على الخلفيات الملونة
- **اختبار الهاتف المحمول** - تحقق من العلامة التجارية على الأجهزة المحمولة
- **موضع الشعار** - استخدم أحجام الشعار المناسبة للسياقات المختلفة

استكشاف الأخطاء

المشاكل الشائعة

لم يتم تطبيق التغييرات

- **السبب:** لم يتم إعادة تشغيل الخدمة أو لم يتم إعادة بناء واجهة المستخدم
- **الحل:** أعد تشغيل خدمات API/واجهة المستخدم بعد تغييرات التكوين

أخطاء صياغة YAML

- **السبب:** تنسيق YAML غير صالح (التهيئة، الاقتباسات، إلخ.)
- **الحل:** تحقق من صحة YAML عبر الإنترنت أو استخدم `yamllint crm_config.yaml`

فشل اتصال قاعدة البيانات

- **السبب:** بيانات الاعتماد خاطئة أو الخادم غير متاح
- **الحل:** تحقق من أن قاعدة البيانات تعمل، وأن بيانات الاعتماد صحيحة

لا تعمل Stripe مدفوعات

- **السبب:** مفاتيح غير متطابقة بين الخلفية والواجهة الأمامية
- **الحل:** تأكد من أن `publishable_key` يتطابق في كلا الملفين

البريد الإلكتروني لا يرسل

- **السبب:** بيانات اعتماد Mailjet غير صالحة أو معرفات القالب
- **الحل:** تحقق من مفتاح/سر Mailjet، تحقق من وجود معرفات القالب

فشل توليد PDF:

- تحقق من أن WeasyPrint مثبت: `pip install weasyprint`
- تحقق من أن اسم القالب يتطابق مع `crm_config.yaml`
- تحقق من `invoice_templates/rendered/` لأخطاء تقديم HTML
- راجع سجلات API لأخطاء قالب Jinja2

الصور لا تظهر في PDF:

- استخدم مسارات ملفات مطلقة: `file:///full/path/to/image.png`
- تأكد من وجود ملفات الصور وقابليتها للقراءة
- تحقق من تنسيق الصورة (يعمل PNG وJPEG بشكل أفضل)
- تحقق من أن مسارات الصور لا تحتوي على أحرف خاصة

مشاكل التنسيق:

- قم بتضمين جميع CSS (لا تدعم أوراق الأنماط الخارجية بشكل موثوق)
- تجنب ميزات CSS المعقدة (قد لا يتم عرض flexbox، الشبكة بشكل صحيح)
- اختبر مع تخطيطات بسيطة أولاً، ثم أضف التعقيد تدريجياً
- استخدم الجداول للتخطيط بدلاً من divs حيثما كان ذلك ممكنًا

التخزين ��لمؤقت لا يتم إبطاله:

- تحقق من أن `invalidate_invoice_cache()` يتم استدعاؤه عند تعديل الفاتورة
- تحقق من أن تحديثات المعاملات تؤدي إلى إبطال التخزين المؤقت
- إذا لزم الأمر `Invoice_PDF_Cache` احذف يدويًا من جدول

قائمة التحقق من التكوين

استخدم هذه القائمة عند نشر OmniCRM:

تكوين الخلفية

- [ ] انسخ `.env.example` إلى `.env`
- [ ] تعيين كلمات مرور قاعدة بيانات قوية
- [ ] تكوين بيانات اعتماد CGRates
- [ ] تحديث `crm_config.yaml` بإعداداتك:
  - [ ] اتصال قاعدة البيانات
  - [ ] أنواع الخدمات
  - [ ] مفاتيح Mailjet API ومعرفات القالب
  - [ ] إشعارات فشل التزويد عبر البريد الإلكتروني
  - [ ] اسم ملف قالب الفاتورة
  - [ ] عنوان URL الأساسي لـ CRM (قابل للوصول علنًا)

- [ ] نقاط نهاية OCS/CGRates
- [ ] تكوين SMSC
- [ ] توليد مفتاح سر JWT جديد
- [ ] مفاتيح Stripe (حية، ليست تجريبية)
- [ ] مفاتيح API وتحديد IP المسموح

تكوين الواجهة الأمامية

- [ ] انسخ `OmniCRM-UI/.env.example` إلى `OmniCRM-UI/.env`
- [ ] تعيين مفتاح Google Maps API
- [ ] تعيين مفتاح Stripe القابل للنشر
- [ ] تحديث العلامة التجارية للشركة:
  - [ ] اسم الشركة
  - [ ] اسم البوابة
  - [ ] اسم الخدمة الذاتية
  - [ ] شعار الشركة
- [ ] تكوين التوطين:
  - [ ] اللغة الافتراضية
  - [ ] اللغة المحلية
  - [ ] الموقع الافتراضي والدولة
  - [ ] رمز العملة ورمزها
- [ ] تعيين اللون الأساسي للعلامة التجارية
- [ ] تكوين تكاملات تطبيق الويب (اختياري)
- [ ] إضافة روابط الدعم والأسئلة الشائعة (اختياري)
- [ ] تعيين معرف تتبع Google Analytics/Tag Manager (اختياري)

أصول العلامة التجارية

- [ ] إنشاء ملفات الشعار (logoSm.png، logoDark.png، logoLight.png)
- [ ] تحميل الشعارات إلى `OmniCRM-UI/src/assets/images/omnitouch/`
- [ ] إنشاء قالب فاتورة HTML مخصص
- [ ] تحميل قالب الفاتورة إلى `OmniCRM-API/invoice_templates/`
- [ ] تحديث `crm_config.yaml` باسم ملف قالب الفاتورة
- [ ] اختبار توليد PDF الفاتورة

- [ ] إعادة بناء واجهة المستخدم: `npm run build`

الأمان

- [ ] تغيير جميع كلمات المرور الافتراضية
- [ ] فريد JWT توليد سر
- [ ] استخدام مفاتيح Stripe (وليس مفاتيح الاختبار) للإنتاج
- [ ] تدوير مفاتيح Mailjet API
- [ ] تمكين قواعد جدار الحماية
- [ ] تكوين تحديد IP المسموح للوصول إلى API
- [ ] إعداد شهادات SSL/TLS
- [ ] تمكين HTTPS لجميع نقاط النهاية
- [ ] مراجعة إعدادات CORS
- [ ] تنفيذ تحديد المعدل
- [ ] تكوين إجراءات النسخ الاحتياطي والاسترداد

الاختبار

- [ ] اختبار تدفق تسجيل العميل
- [ ] اختبار تزويد الخدمة من البداية إلى النهاية
- [ ] التحقق من إرسال إشعارات البريد الإلكتروني بشكل صحيح
- [ ] اختبار توليد الفاتورة وتنزيل PDF
- [ ] التحقق من معالجة المدفوعات (Stripe)
- [ ] التحقق من مصادقة المستخدم و2FA
- [ ] اختبار الانتحال وتسجيل التدقيق
- [ ] التحقق من مزامنة بيانات الاستخدام من OCS
- [ ] اختبار إنشاء خطة العمل وتجديدها
- [ ] التأكد من أن تخصيص المخزون يعمل بشكل صحيح

النشر

- [ ] بدء قاعدة البيانات (MySQL)
- [ ] بدء CGRates
- [ ] بدء OmniCRM API

- [ ] بدء OmniCRM UI
- [ ] تكوين الوكيل العكسي (nginx، traefik)
- [ ] إعداد المراقبة (Grafana، Prometheus)
- [ ] تكوين تجميع السجلات
- [ ] إعداد النسخ الاحتياطية التلقائية
- [ ] تدريب الموظفين على استخدام النظام

الوثائق ذات الصلة

- إدارة RBAC والمستخدمين
- المنتجات والخدمات
- تزويد Ansible
- إدارة المخزون
- فواتير العملاء
- المصادقة الثنائية
- رعاية العملاء والانتحال
- *تكاملات بائعي الدفع*
- `administration_api_keys` - إدارة مفاتيح API
- `integrations_mailjet` - تكامل البريد الإلكتروني
- `concepts_api` - مصادقة API

نظرة عامة على المخزون في OmniCRM

نظام OmniCRM مصمم لإدارة وتتبع العناصر المادية والافتراضية المستخدمة من قبل مشغلي الشبكات والعملاء في **المخزون**.

هذا يعني أننا يمكننا تتبع جميع أنواع العناصر، مثل المودمات، وأرقام الهواتف، وكتل عناوين IP، أو حتى الأجهزة المادية مثل CPEs أو GPON ONTs اللاسلكية الثابتة.

انظر أيضًا: سمات العملاء لتخزين بيانات التعريف المخصصة، وعلامات العملاء للتصنيف البصري.

لدعم العملاء الذين لديهم بصمة شبكة ثابتة، يمكن لنظام المخزون أيضًا تتبع المنازل التي تم تمريرها لخدمة معينة، مما يسمح للمشغلين بإجراء تأهيل الخدمة عن بُعد، ��لأولئك الذين يعملون في الميدان مع مواقعها CPEs في شبكة لاسلكية ثابتة، يمكننا تتبع المنصوبة.

> **ⓘ ملاحظة**
> ترتبط عناصر المخزون بالمنتجات أثناء التوفير من خلال حقل `inventory_items_list`. للحصول على شرح كامل حول كيفية تكامل المخزون مع توفير المنتجات، بما في ذلك واجهة Ansible، اختيار المخزون ودمج برنامج، انظر دليل دورة حياة المنتج الكامل - متطلبات المخزون.

الهدف

يخدم مخزون OmniCRM عدة أغراض رئيسية:

1. **توفير الخدمات**: عندما يقوم العميل بالتسجيل للحصول على خدمة، قد تحتاج عناصر مثل المودمات، وبطاقات SIM، أو أرقام الهواتف إلى التخصيص. يتتبع نظام المخزون هذه العناصر ويربطها بالعملاء.
2. **إدارة المخزون**: بالنسبة للمخزون المادي، مثل الأجهزة أو المعدات الأخرى، يساعد المخزون المشغلين في الحفاظ على رؤية لما هو متاح، وأين يتم تخزينه، وما تم تخصيصه أو بيعه للعملاء.
3. **تخصيص العملاء**: يسمح النظام بتخصيص العناصر للعملاء، سواء للاستخدام في خدمة (مثل تخصيص مودم لحساب الإنترنت الخاص بالعميل) أو للبيع المباشر.

4. **تأهيل الخدمة / بصمة الشبكة**: من خلال تخزين المعلومات حول بصمة الشبكة، مثل لاسلكية ثابتة تم نشرها، يسمح CPE أو كل GPON، كل منزل تم تمريره لخدمة للموظفين بإجراء تأهيل الخدمة عن بُعد ورؤية ما إذا كانت هناك بصمة شبكة في منطقة معينة.

مثال على دورة حياة المخزون

لتوضيح كيفية عمل نظام المخزون، اعتبر الأمثلة التالية

مثال بطاقة SIM

أولاً، يتم إنشاء نموذج مخزون لبطاقات SIM من Omnitouch. تم طلب دفعة من 1000 بطاقة SIM المطلوبة في المخزون في حالة *في* SIM ويتم تحميل بطاقات (إذا لم يكن موجودًا بالفعل) *النقل*.

يتم وضع علامة عليها كـ *في المخزون*، ويمكن تخصيصها لمتاجر التجزئة، SIM بمجرد استلام بطاقات SIM - هذا المختلفة، مع تحديث موقع عنصر المخزون ليعكس أي متجر تجزئة تتواجد فيه كل بطاقة مفيد لتتبع مستويات المخزون في كل متجر وضمان أن كل متجر لديه مخزون كافٍ لتلبية طلب العملاء.

عندما يقوم العميل بالتسجيل للحصول على خدمة موبايل في المتجر، يتم تخصيص بطاقة SIM للعميل، ويتم تحديث SIM للعميل وتغيير الحالة إلى *مخصصة*. ثم يتم تعيين عنصر مخزون بطاقة الحالة إلى *قيد الاستخدام*.

إذا قام العميل بإلغاء الخدمة أو أصبحت الخدمة خاملة، يتم وضع علامة على بطاقة SIM *كـ مفككة*.

مثال المنازل المارة GPON

بالنسبة لشبكة GPON، يمكن لنظام المخزون تتبع كل منزل تم تمريره لخدمة معينة.

عندما يتم بناء منطقة جديدة، يمكن إضافة كل عنوان تم تمريره إلى المخزون.

هذا يسمح للمشغلين برؤية أي المنازل تم تمريرها لخدمة معينة، وأي المنازل لم يتم تمريرها بعد.

عندما يقوم العميل بالتسجيل للحصول على خدمة، يمكن لـ OmniCRM تشغيل تأهيل الخدمة تلقائيًا ضد عنوان العميل، لمعرفة ما إذا كان العنوان في مخزون المنازل المارة وما الخدمات التي يمكن تقديمها.

نماذج المخزون

تتيح ميزة **نموذج المخزون** إنشاء أي عدد من أنواع العناصر مع حقول محددة مسبقًا. تعمل هذه النماذج كخطط تحدد الخصائص الأساسية لعناصر مختلفة، مثل:

- **المودمات** مع عنوان MAC (`itemtext1`) ورقم تسلسلي (`itemtext2`).
- **المنازل المارة** لخدمة معينة، مع موقع وحالة (مثل، تم تمريرها أو لم يتم تمريرها).
- **أرقام الهواتف** مع رقم أساسي (`itemtext1`) وموقع جغرافي (`itemtext2`).
- **الموارد الافتراضية** مثل كتل عناوين IP، مع معرفات مرتبطة من خلال النموذج.

يحدد كل نموذج مخزون ما يصل إلى 20 حقل نصي قابل للتخصيص (`itemtext1` إلى `itemtext20`) مع تسميات مقابلة (`itemtext1_label` إلى `itemtext20_label`) التي تصف ما يمثله كل حقل. على سبيل المثال، قد يحدد نموذج المودم ليكون `itemtext1_label` "عنوان MAC" و `itemtext2_label` ليكون "رقم تسلسلي".

يمكن للمشغلين تخصيص الحقول لكل نوع عنصر باستخدام **نموذج المخزون**. تتيح هذه النماذج تصنيف العناصر وإدارتها بطريقة منظمة، مما يضمن الاتساق في كيفية تتبع العناصر.

الربط بالمنتجات:

يتم الإشارة إلى أسماء نماذج المخزون في تعريفات المنتجات عبر حقل `inventory_items_list`. عند توفير منتج، يعرض النظام منتقي المخزون الذي يظهر فقط العناصر المطابقة لأنواع النماذج المطلوبة.

مثال: يتطلب منتجًا مع `inventory_items_list: "['SIM Card', 'Mobile Number']"` وجود نموذجين مخزنين مسميين بدقة "Mobile Number" و "SIM Card". أسماء النماذج حساسة لحالة الأحرف.

للحصول على تفاصيل كاملة حول كيفية ارتباط نماذج المخزون بتوفير المنتجات، انظر دورة حياة المنتج - متطلبات المخزون.

إنشاء نماذج المخزون ��بر واجهة المستخدم

لإنشاء نموذج مخزون جديد:

1. انتقل إلى **المخزون** ← **النماذج** من القائمة الرئيسية
2. انقر على زر **إضافة نموذج**
3. املأ الحقول المطلوبة:

   المعلومات الأساسية:

   - **الأيقونة** (اختياري): اسم فئة الأيقونة للتعريف البصري (مثل، `fa-solid fa-sim-card`)
   - **العنصر** (مطلوب): اسم النموذج يجب أن يتطابق بالضبط مع ما يتم استخدامه في `inventory_items_list` للمنتجات)

   معلومات التكلفة (مطلوبة):

   - **تكلفة الجملة** (مطلوبة): تكلفتك لشراء أو توفير هذا النوع من العناصر
   - **تكلفة التجزئة** (مطلوبة): السعر القياسي إذا تم بيعه بشكل منفصل للعملاء

   > **ⓘ ملاحظة**
   >
   > تعتبر تكاليف الجملة والتجزئة المحددة هنا قيمًا افتراضية عند إنشاء عناصر مخزون جديدة من هذا النموذج. يمكن أن تحتوي عناصر المخزون الفردية على تكاليف مختلفة إذا لزم الأمر.

   تسميات الحقول:

- **سمية نص العنصر 1** (مطلوبة): تسمية للحقل القابل للتخصيص�الأول (تكون افتراضية على "رقم الموديل")
  - أمثلة شائعة: "ICCID" لبطاقات SIM، "عنوان MAC" للمودمات، "رقم الهاتف" للأرقام
- **تسمية نص العنصر 2** (مطلوبة): تسمية للحقل القابل للتخصيص الثاني (تكون افتراضية على "رقم تسلسلي")
  - أمثلة شائعة: "IMSI" لبطاقات SIM، "رقم تسلسلي" للأجهزة، "المنطقة الجغرافية" للأرقام
- **تسميات نص العنصر 3-20** (اختياري): تسميات الحقول الإضافية حسب الحاجة
  - انقر على **إضافة حقل** لإضافة المزيد من الحقول المخصصة
  - أضف فقط الحقول التي ستستخدمها فعليًا لهذا النوع من العناصر

إعدادات الرؤية:

- **السماح لموظفي القائمة المنسدلة**: تمكين الموظفين من اختيار هذا النوع من المخزون في القوائم المنسدلة
- **السماح للعميل بالقائمة المنسدلة**: تمكين العملاء من رؤية/اختيار هذا الن�ع من المخزون (بوابة العملاء)

4. انقر على **حفظ** لإنشاء النموذج

تحرير نماذج المخزون

لتحرير نموذج موجود:

1. انتقل إلى **المخزون** → **النماذج**
2. ابحث عن النموذج في القائمة
3. انقر على زر **تحرير**
4. عدل الحقول حسب الحاجة
5. انقر على **حفظ**

> **⚠WARNING**
>
> يؤثر فقط على العناصر الجديدة التي تم إنشاؤها بعد التغيير. تحتفظ عناصر المخزون الموجودة ببياناتها ولكن ستظهر بالتسميات الجديدة. تغيير تسميات الحقول (مثل، `itemtext1_label`)

> **⚠️ حذر**
> أسماء النماذج المشار إليها في حقول `inventory_items_list` للمنتجات حساسة لحالة الأحرف. سيؤدي إعادة تسمية نموذج إلى كسر الرابط بالمنتجات التي تستخدم الاسم القديم.

أمثلة شائعة على النماذج

نموذج بطاقة SIM:

- العنصر: "SIM Card"
- تكلفة الجملة: 2.50
- تكلفة التجزئة: 10.00
- تسمية نص العنصر 1: "ICCID"
- تسمية نص العنصر 2: "IMSI"
- تسمية نص العنصر 3: "نوع SIM" (مادي/eSIM)

نموذج رقم الهاتف المحمول:

- العنصر: "Mobile Number"
- تكلفة الجملة: 1.00
- تكلفة التجزئة: 0.00
- تسمية نص العنصر 1: "رقم الهاتف"
- تسمية نص العنصر 2: "المنطقة الجغرافية"
- تسمية نص العنصر 3: "نوع الرقم" (محمول/خط أرضي)

نموذج CPE اللاسلكية الثابتة:

- العنصر: "Fixed Wireless CPE"
- تكلفة الجملة: 250.00
- تكلفة التجزئة: 450.00
- تسمية نص العنصر 1: "عنوان MAC"
- تسمية نص العنصر 2: "رقم تسلسلي"
- تسمية نص العنصر 3: "إصدار البرنامج الثابت"
- تسمية نص العنصر 4: "الشركة المصنعة"
- تسمية نص العنصر 5: "النموذج"

نموذج GPON ONT:

- العنصر: "GPON ONT"
- تكلفة الجملة: 45.00
- تكلفة التجزئة: 0.00 (مضمنة مع الخدمة)
- تسمية نص العنصر 1: "رقم تسلسلي"
- تسمية نص العنصر 2: "عنوان MAC"
- تسمية نص العنصر 3: "موقع PON"
- تسمية نص العنصر 4: "النموذج"

إنشاء وإدارة عناصر المخزون

بمجرد تعريف **نموذج المخزون**، يمكن إنشاء عناصر **المخزون** الفردية. يمثل كل عنصر مخزون حالة معينة من نوع عنصر (مثل، مودم معين أو رقم هاتف) يمكن أن يكون:

- **مخصص للعملاء**: ترتبط العناصر بالعملاء لتوفير الخدمة (مثل، تخصيص الأجهزة لاتصال الإنترنت).
- **متابعة المخزون**: يمكن للمشغلين مراقبة المخزون المتاح، مثل العناصر غير المباعة أو غير المخصصة.
- **مباعة أو مفككة**: بمجرد بيعها، يتم وضع علامة على العناصر مع الطوابع الزمنية ذات الصلة (مثل، `sold_date`) ولا يمكن اعتبارها مخزونًا متاحًا بعد الآن.

إدارة المخزون بكفاءة، ويساعد في تخصيص الموارد OmniCRM من خلال هذا النظام، يسهل للعملاء، ويوفر رؤية تفصيلية حول حالة وتاريخ كل عنصر.

يمكن ربط **الخدمات** بعنصر **المخزون** معين، مما يسمح بتتبع سهل لأي العناصر مرتبطة بأي العملاء أو الخدمات.

بتحديث حالة العنصر Ansible بمجرد تخصيص عنصر المخزون لعميل، يمكن أن تقوم برامج وتاريخه لتعكس التخصيص. يضمن ذلك أن يكون لدى المشغلين سجل دقيق حول العناصر التي هي قيد الاستخدام وأيها متاحة للتخصيص، بالإضافة إلى معرفة أي عميل يستخدم أي عنصر.

يمكننا عرض العناصر المخصصة لعميل من صفحة ملف العميل في علامة **المخزون**.

بالنسبة لعناصر المخزون المرتبطة بـ **خدمة**، يمكننا رؤية ذلك من خلال تحرير الخدمة، لرؤية عناصر المخزون المرتبطة.

حقول عنصر المخزون

يحتوي كل عنصر مخزون على معلومات شاملة منظمة في عدة فئات:

معلومات العنصر الأساسية

- **inventory_id** - (يتم إنشاؤه ت��قائيًا) معرف فريد لعنصر المخزون
- **item** - نوع العنصر (مثل، "SIM Card"، "Modem"، "Phone Number") يتطابق مع اسم نموذج المخزون
- **inventory_template_id** - رابط إلى نموذج المخزون الذي يحدد هذا النوع من العنصر
- **customer_id** - (قابل للإلغاء) إذا تم تخصيصه لعميل، معرف العميل
- **service_id** - (قابل للإلغاء) إذا كان مرتبطًا بخدمة معينة، معرف الخدمة
- **item_location** - الموقع المادي أو المنطقي للعنصر:
  - بالنسبة للعناصر المادية: المبنى، المستودع، موقع الرف، اسم المتجر، إلخ.
  - بالنسبة للعناصر الافتراضية: الموقع الجغرافي، موقع كتلة IP، منطقة نطاق الرقم، إلخ.
- **item_state** - (قيم محددة) الحالة الحالية لعنصر المخزون:
  - **جديد** - عنصر جديد تمامًا، غير مستخدم
  - **مستخدم** - تم استخدامه سابقًا ولكنه يعمل
  - **استخدام داخلي** - مخصص للاختبار الداخلي أو استخدام الموظفين
  - **مخصص** - مخصص حاليًا لعميل أو خدمة
  - **تالف** - غير وظيفي، يحتاج إلى ��صلاح أو التخلص منه
  - **خارج الخدمة** - غير متاح مؤقتًا
  - **مفقود** - لا يمكن تحديد موقع العنصر
  - **مسروق** - تم سرقة العنصر

حقول العنصر القابلة للتخصيص (من النموذج)

يدعم نظام المخزون ما يصل إلى 20 حقل نصي قابل للتخصيص يتم تعريف معناه من خلال نموذج المخزون:

- **itemtext1** - أول حقل قابل للتخصيص (مطلوب، التسمية محددة بواسطة `itemtext1_label` للنموذج)
  - مثال: بالنسبة للمودمات، قد يكون "عنوان MAC"
  - مثال: بالنسبة لبطاقات SIM، قد يكون "ICCID"

    - مثال: بالنسبة لأرقام الهواتف، قد يكون "رقم الهاتف"
- **itemtext2** إلى **itemtext20** - حقول إضافية قابلة للتخصيص (اختياري، التسميات محددة بواسطة النموذج)
    - مثال: قد يكون itemtext2 "رقم تسلسلي للمودمات"
    - مثال: قد يكون itemtext2 لبطاقات SIM "IMSI"
    - مثال: قد يكون itemtext3 "إصدار البرنامج الثابت للمودمات"

يحدد كل نموذج مخزون أي من هذه الحقول يتم استخدامها وما تمثله عبر حقول التسميات المقابلة (`itemtext1_label`، `itemtext2_label`، إلخ).

معلومات التكلفة

- **wholesale_cost** - تكلفتك لشراء/توفير هذا العنصر (عائم)
- **retail_cost** - السعر المفروض على العميل إذا تم بيعه بشكل منفصل (عائم)
- **sold_date** - الطابع الزمني عندما تم بيع العنصر أو تخصيصه للعميل

العنوان الفعلي (لأجهزة الشبكة والمواقع)

يستخدم لتتبع مواقع النشر الفعلية، خاصة لأجهزة الشبكة الثابتة (مودمات، ONTs، CPEs) المنازل المارة:

- **address_line_1** - عنوان الشارع، رقم المبنى، رقم الوحدة
- **address_line_2** - معلومات عنوان إضافية (جناح، شقة، طابق)
- **city** - المدينة أو البلدة
- **state** - الولاية، المقاطعة، أو المنطقة
- **zip_code** - الرمز البريدي
- **country** - اسم الدولة

الموقع الجغرافي (يتم ملؤه تلقائيًا من واجهة الويب)

عند إنشاء عناصر المخزون عبر واجهة الويب مع إكمال العنوان، يتم ملء هذه الحقول تلقائيًا:

- **google_maps_place_id** - معرف مكان Google Maps للعنوان
- **plus_code** - رمز Google Maps Plus (Open Location Code) للموقع الدقيق
- **latitude** - خط العرض الجغرافي (درجات عشرية)
- **longitude** - خط الطول الجغرافي (درجات عشرية)

تمكن هذه الحقول من:

- رسم مواقع المخزون على عرض الخريطة
- حسابات القرب لتأهيل الخدمة
- تحليل التغطية لتخطيط الشبكة
- تحسين المسار لتوجيه الفنيين الميدانيين

إدارة الأجهزة وروابط الوصول

يوفر حقل `management_url` وصولًا سريعًا إلى واجهات الأجهزة وروابط التوفير:

- **management_url** - رابط الوصول لعنصر المخزون
  - **أجهزة الشبكة**: رابط واجهة الويب (مثل، `https://192.168.1.1` للموجهات، المفاتيح، ONTs، CPEs)
  - **ملفات تعريف eSIM**: عنوان LPA (مساعد الملف المحلي) لتفعيل eSIM (مثل، `LPA:1$smdp.example.com$ACTIVATION-CODE-HERE`)
  - **حالات استخدام أخرى**: أي رابط يحتاج إلى أن يكون سهل الوصول عبر الجهاز المحمول

  تو ليد رمز QR

  يقوم النظام تلقائيًا بإنشاء **رمز** `management_url`، عند عرض عناصر المخزون مع **قابل للمسح QR**:

  - **عرض تفاصيل عنصر المخزون**: يتم عرض رمز QR 128 بحجم 128x بجانب الرابط
  - **جدول مخزون الخدمة**: يتم عرض رمز QR 64 بحجم 64x للعناصر المخصصة
  - **التنسيق**: يتم عرض كل من رمز QR ورابط قابل للنقر معًا

  حالات الاستخدام الشائعة

  - **فنيي الشبكة**: مسح رمز QR للوصول إلى واجهة إدارة الأجهزة دون كتابة عناوين IP
  - **تفعيل eSIM**: يقوم العملاء بمسح رمز QR من CRM لتثبيت ملف تعريف eSIM على أجهزتهم
  - **خدمة العملاء الذاتية**: توفير وصول سهل لتكوين الأجهزة أو بوابات العملاء
- **management_username** - اسم المستخدم الإداري للوصول إلى الجهاز

- **management_password** - كلمة مرور المسؤول للوصول إلى الجهاز (مشفرة في حالة السكون)

إدارة التكوين

بالنسبة للأجهزة التي تحتوي على ملفات تكوين:

- **config_content** - محتوى ملف التكوين الكامل (مخزن كنص)
  - مفيد للنسخ الاحتياطي، وإصدار النسخ، واستعادة الكوارث
  - يمكن تخزين تكوينات الموجهات، تكوينات المفاتيح، تكوينات CPE، إلخ.
- **config_file_path** - مسار ملف التكوين الخارجي إذا تم تخزينه بشكل منفصل
  - بديل لتخزين التكوين الكامل في قاعدة البيانات
  - مسار إلى مشاركة الشبكة، مستودع التحكم في الإصدار، أو نظام إدارة التكوين

الملاحظات والبيانات الوصفية

- **inventory_notes** - ملاحظات حرة حول عنصر المخزون
  - ملاحظات التثبيت
  - تاريخ الصيانة
  - ميزات أو متطلبات معالجة خاصة
  - معلومات البائع
  - تفاصيل الضمان
- **created** - الطابع الزمني عندما تم إنشاء عنصر المخزون في النظام (يتم تعيينه تلقائيًا)
- **last_modified** - الطابع الزمني لآخر تحديث لعنصر المخزون (يتم تحديثه تلقائيًا)

أمثلة على استخدا�� الحقول

مثال 1: بطاقة SIM المحمولة

```
{
  "inventory_id": 1001,
  "item": "SIM Card",
  "inventory_template_id": 5,
  "itemtext1": "89612345678901234 56",
  "itemtext2": "310120123456789",
  "itemtext3": "Physical",
  "item_location": "Warehouse A, Shelf 3",
  "item_state": "Assigned",
  "customer_id": 456,
  "service_id": 789,
  "wholesale_cost": 2.50,
  "retail_cost": 10.00,
  "sold_date": "2025-01-15T10:30:00Z",
  "inventory_notes": "Activated on 2025-01-15"
}
```


مثال 2: ملف تعريف eSIM المحمولة


```
{
  "inventory_id": 1002,
  "item": "eSIM",
  "inventory_template_id": 6,
  "itemtext1": "89612345678901234 57",
  "itemtext2": "310120123456790",
  "itemtext3": "eSIM",
  "item_location": "Virtual Inventory",
  "item_state": "Assigned",
  "customer_id": 457,
  "service_id": 790,
  "management_url": "LPA:1$smdp.example.com$ACTIVATION-CODE-
ABC123XYZ",
  "wholesale_cost": 0.00,
  "retail_cost": 0.00,
  "sold_date": "2025-01-16T14:20:00Z",
  "inventory_notes": "eSIM profile ready for activation"
}
```


عند عرض عنصر المخزون eSIM هذا، تعرض واجهة المستخدم رمز QR يحتوي على عنوان LPA. يقوم العملاء بمسح رمز QR هذا باستخدام أجهزتهم المحمولة لتثبيت ملف تعريف eSIM.

مثال 3: معدات العميل (CPE) - لاسلكية ثابتة


```
{
  "inventory_id": 2001,
  "item": "Fixed Wireless CPE",
  "inventory_template_id": 10,
  "itemtext1": "AA:BB:CC:DD:EE:FF",
  "itemtext2": "FW2024-12345",
  "itemtext3": "v2.4.1",
  "itemtext4": "Ubiquiti",
  "itemtext5": "LiteBeam AC Gen2",
  "item_location": "Customer Site",
  "item_state": "Assigned",
  "customer_id": 789,
  "service_id": 1234,
  "address_line_1": "123 Main Street",
  "address_line_2": "Apt 4B",
  "city": "Sydney",
  "state": "NSW",
  "zip_code": "2000",
  "country": "Australia",
  "latitude": "-33.8688",
  "longitude": "151.2093",
  "management_url": "https://192.168.100.1",
  "management_username": "admin",
  "management_password": "encrypted_password_here",
  "config_file_path": "/configs/cpe/fw2024-12345.conf",
  "inventory_notes": "Installed 2025-01-10. Customer reports
excellent signal. Pointing: Azimuth 45°, Elevation 15°"
}
```


مثال 4: GPON ONT مع عنوان كامل

```
{
  "inventory_id": 3001,
  "item": "GPON ONT",
  "inventory_template_id": 15,
  "itemtext1": "ALCL12345678",
  "itemtext2": "AA:BB:CC:DD:EE:FF",
  "itemtext3": "OLT-1, PON 3, ONT 42",
  "itemtext4": "Nokia G-010G-A",
  "item_location": "Customer Premises",
  "item_state": "Assigned",
  "customer_id": 321,
  "service_id": 654,
  "address_line_1": "456 Fiber Lane",
  "city": "Melbourne",
  "state": "VIC",
  "zip_code": "3000",
  "country": "Australia",
  "google_maps_place_id": "ChIJ1234567890",
  "plus_code": "4RRH+2C Melbourne VIC",
  "latitude": "-37.8136",
  "longitude": "144.9631",
  "management_url": "https://192.168.1.1",
  "management_username": "admin",
  "config_content": "# ONT Configuration\nwlan-ssid: HomeNetwork\nwlan-password: encrypted...",
  "wholesale_cost": 45.00,
  "retail_cost": 0.00,
  "inventory_notes": "Provisioned 2025-01-20. Optical power: -22dBm"
}
```


ⓘ ملاحظة

تعرض ،(مثل الأمثلة 2، 3، و4 أعلاه) `management_url` عند عرض عناصر المخزون مع واجهة المستخدم تلقائيًا:

- رمز QR أو عنوان LPA قابل للمسح يحتوي على الرابط
- رابط قابل للنقر (لروابط الويب)

حالات الاستخدام:

- **تفعيل eSIM** (المثال 2): يقوم العملاء بمسح رمز QR لتثبيت ملف تعريف eSIM على أجهزتهم
- **الوصول إلى معدات الشبكة** (الأمثلة 3 و4): يقوم الفنيون بمسح للوصول إلى واجهات إدارة الأجهزة دون كتابة عناوين IP يدويًا

مثال 5: رقم الهاتف (المخزون الافتراضي)

```
{
  "inventory_id": 4001,
  "item": "Phone Number",
  "inventory_template_id": 20,
  "itemtext1": "+61412345678",
  "itemtext2": "Melbourne",
  "itemtext3": "Mobile",
  "item_location": "Australia - VIC",
  "item_state": "Assigned",
  "customer_id": 555,
  "service_id": 888,
  "wholesale_cost": 1.00,
  "retail_cost": 0.00,
  "inventory_notes": "Ported from Telstra on 2025-01-05"
}
```

شرح حالات عناصر المخزون

يتتبع حقل `item_state` دورة حياة عناصر المخزون:

- **جديد** → **مخصص** - التدفق العادي عند التوفير لعميل
- **مخصص** → **مستخدم** - بعد إلغاء الخدمة، يمكن إعادة استخدام العنصر
- **جديد** → **استخدام داخلي** - مخصص للاختبار، العروض، أو استخدام الموظفين
- **مخصص** → **تالف** - العنصر فشل، يحتاج إلى RMA أو الخلص
- **أي حالة** → **مفقود** - لا يمكن تحديد موقع العنصر (يؤدي إلى تدقيق)
- **أي حالة** → **مسروق** - تم سرقة العنصر (يؤدي إلى تقرير أمني)
- **تالف/مستخدم** → **جديد** - بعد التجديد أو الإصلاح

يسمح تصفية والبحث عن المخزون حسب الحالة للمشغلين بـ:

- تتبع المخزون المتاح (عناصر جديدة)

- تحديد العناصر المخصصة للعملاء (مخصصة)
- العثور على العناصر المتاحة لإعادة الاستخدام (مستخدمة)
- مراقبة مشكلات المعدات (تالف، خارج الخدمة)
- تدقيق العناصر المفقودة (مفقود، مسروق)

علامات العملاء

العلامات هي روابط ملونة مفيدة يمكن إضافتها إلى عميل للمساعدة في تصنيفهم، على سبيل المثال، قد يكون لدى العميل علامة لـ "تذكرة دعم مفتوحة" أو "فاتورة متأخرة جداً" أو "أحمق".

للتخزين الهيكلي للبيانات الوصفية وبيانات المفتاح والقيمة المخصصة، راجع سمات العملاء.

تظهر العلامات كأقراص على صفحة ملف العميل، ويمكن تخصيص لون القرص جنبًا إلى جنب مع الرابط.

إحدى حالات الاستخدام الشائعة هي وضع علامات على العملاء الذين لديهم تذكرة دعم مفتوحة، بحيث يمكن لفريق الدعم الانتقال بسهولة إلى التذكرة المفتوحة من صفحة ملف العميل.

يمكن إنشاء العلامات في النظام بواسطة مسؤول من خلال واجهة المستخدم أو بواسطة أنظمة الطرف الثالث عبر API ويمكن أن تحتوي على تواريخ بدء وانتهاء، بحيث يمكن إزالتها تلقائيًا بعد فترة معينة.

إدارة العلامات عبر واجهة المستخدم

عرض علامات العملاء

لعرض العلامات لعميل:

1. انتقل إلى صفحة نظرة عامة العميل

2. انقر على علامة **العلامات**
3. سترى قائمة بجميع العلامات النشطة للعميل، تظهر:
    - معاينة العلامة مع اللون المحدد
    - نص العلامة
    - تاريخ التفعيل (عندما تصبح العلامة مرئية)
    - تاريخ إلغاء التفعيل (عندما سيتم إخفاء العلامة)
    - رابط (إذا تم تكوينه)

إنشاء علامة جديدة

لإنشاء علامة جديدة لعميل:

1. انتقل إلى صفحة نظرة عامة العميل
2. انقر على علامة **العلامات**
3. انقر على زر **إضافة علامة**
4. املأ الحقول المطلوبة:
    - **نص العلامة** (مطلوب): النص الذي سيتم عرضه ع��ى العلامة
    - **لون العلامة** (مطلوب): اختر لونًا باستخدام منتقي الألوان أو أدخل رمز سداسي
    - **رابط العلامة** (اختياري): URL الذي سيفتح عند النقر على العلامة
    - **تاريخ التفعيل** (مطلوب): التاريخ الذي يجب أن تبدأ فيه العلامة في العرض
    - **تاريخ إلغاء التفعيل** (مطلوب): التاريخ الذي يجب أن تتوقف فيه العلامة عن العرض (الافتراضي هو 2099-01-01)
5. عاين علامتك في قسم المعاينة
6. انقر على **إنشاء علامة**

تعديل علامة

لتعديل علامة موجودة:

1. انتقل إلى صفحة نظرة عامة العميل
2. انقر على علامة **العلامات**
3. ابحث عن العلامة التي تريد تعديلها في القائمة
4. انقر على زر **تعديل** (قلم رصاص)
5. عدل الحقول حسب الحاجة
6. انقر على **تحديث العلامة**

حذف علامة

لحذف علامة:

1. انتقل إلى صفحة نظرة عامة العميل
2. انقر على علامة **العلامات**
3. ابحث عن العلامة التي تريد حذفها في القائمة
4. انقر على زر **حذف** (سلة المهملات)
5. أكد الحذف في النافذة المنبثقة

مرجع حقل العلامة

تكامل API

يمكن أيضًا إدارة العلامات برمجيًا عبر API:

إنشاء علامة:

```
PUT /crm/tag/
{
  "tag_text": "عميل VIP",
  "tag_hex_color": "FFD700",
  "tag_link": "https://example.com/vip",
  "tag_active_date": "2025-01-01 00:00:00",
  "tag_deactivate_date": "2099-12-31 23:59:59",
  "customer_id": 12
}
```

تحديث علامة:

```
PATCH /crm/tag/tag_id/{tag_id}
{
  "tag_text": "نص العلامة المحدث",
  "tag_hex_color": "FF0000"
}
```

الحصول على العلامات حسب العميل:

```
GET /crm/tag/customer_id/{customer_id}
```

حذف علامة:

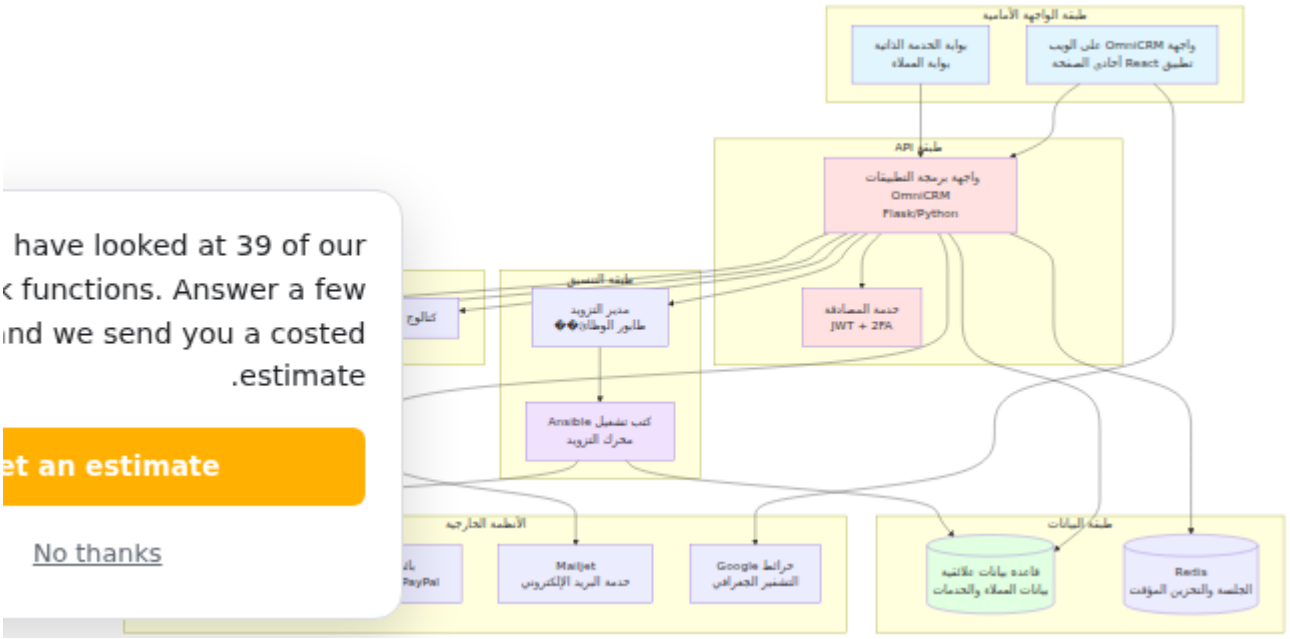
```
DELETE /crm/tag/tag_id/{tag_id}
```

بنية نظام OmniCRM

يوفر هذا المستند نظرة عامة على بنية نظام OmniCRM، بما في ذلك علاقات المكونات وتدفق البيانات.

نظرة عامة على النظام على مستوى عالٍ

OmniCRM هو منصة BSS/OSS شاملة تدمج العديد من المكونات الرئيسية لتوفير إدارة خدمات كاملة لمزودي الاتصالات.



المكونات الأساسية

1. تطبيقات الواجهة الأمامية

واجهة OmniCRM على الويب

- تطبيق React أحادي الصفحة
- واجهة الموظفين لإدارة العملاء، وتزويد الخدمات، والفوترة
- تحديثات حالة التزويد في الوقت الحقيقي
- التحكم في الوصول بناءً على الدور

بوابة الخدمة الذاتية

- بوابة موجهة للعملاء
- إدارة الخدمات وتتبع الاستخدام
- عرض الفواتير والدفع
- قاعدة شفرة مشتركة مع واجهة الموظفين، وجهات نظر مختلفة

2. طبقة API

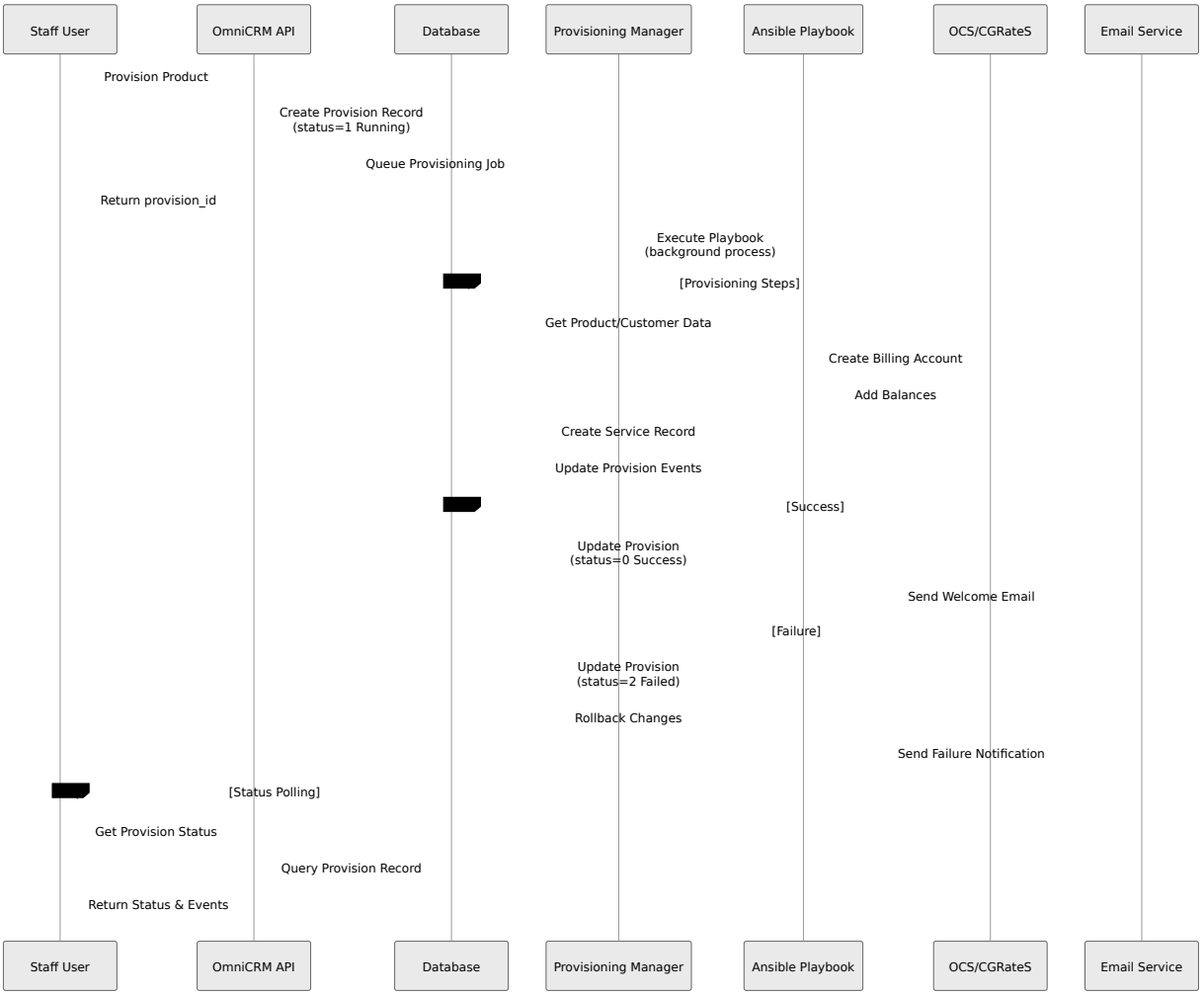
واجهة برمجة التطبيقات OmniCRM (Flask/Python)

- واجهة برمجة تطبيقات RESTful لجميع العمليات
- وثائق OpenAPI/Swagger
- مصادقة تعتمد على JWT
- تحديد المعدل والتخزين المؤقت
- دعم WebSocket للتحديثات في الوقت الحقيقي

المسارات الرئيسية لواجهة API:

- `/crm/customer/*` - إدارة العملاء
- `/crm/service/*` - عمليات الخدمة
- `/crm/product/*` - كتالوج المنتجات
- `/crm/provision/*` - عمليات التزويد
- `/crm/transaction/*` - معاملات الفوترة
- `/crm/invoice/*` - إدارة الفواتير

3. نظام التزويد

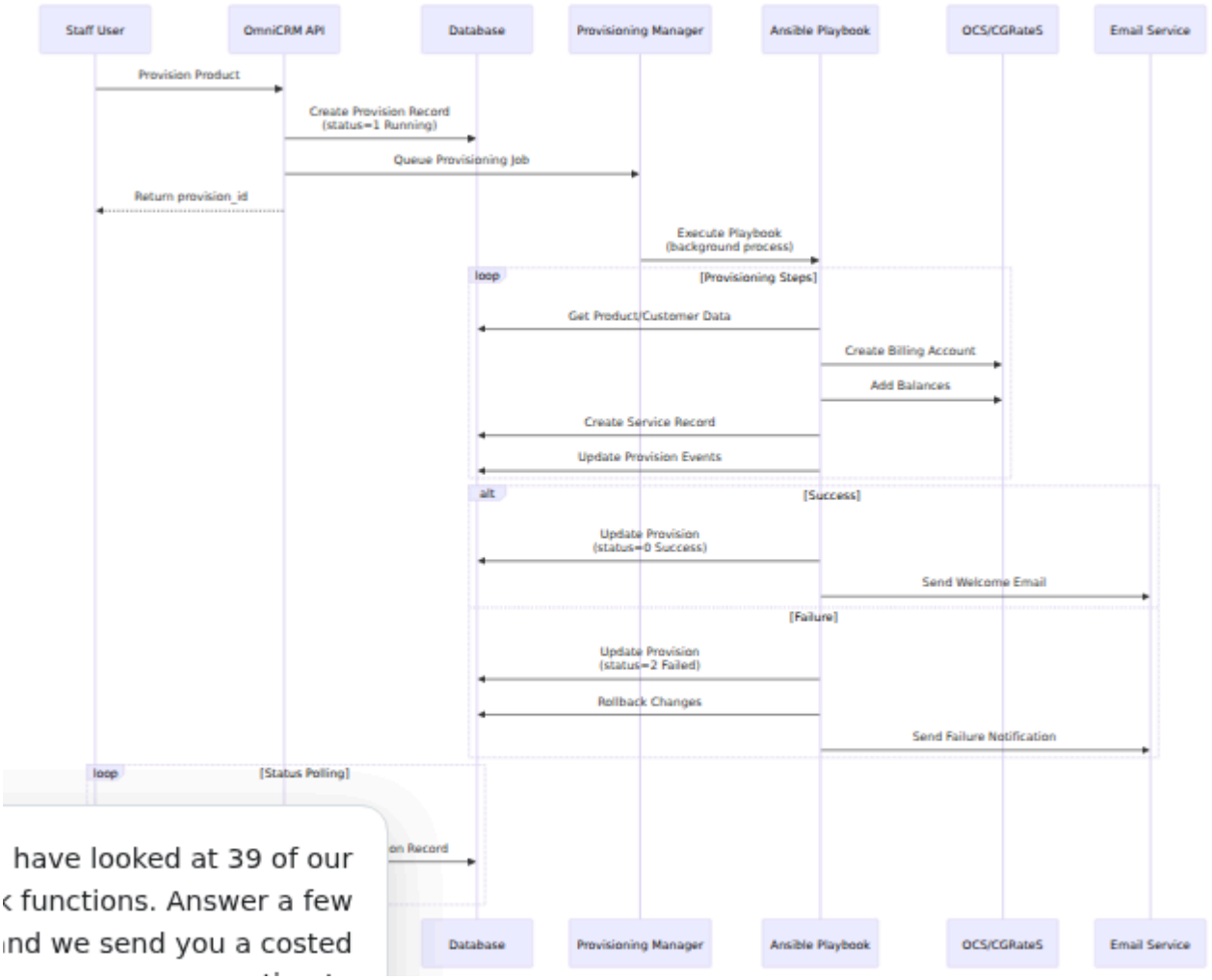


4. الفوترة والتقييم

تكامل OCS/CGRateS

- التحصيل والتقييم في الوقت الحقيقي
- إدارة الحسابات
- تتبع الرصيد (صوت، بيانات، نقدي، SMS)
- خطط العمل للرسوم المتكررة
- إشعارات بناءً على العتبات

سير عمل الفوترة:



5. نموذج البيانات

يستخدم OmniCRM قاعدة بيانات علائقية تحتوي على النماذج الأساسية التالية. لمخططات العلاقات الكيانية المرئية، انظر العملاء، جهات الاتصال، المواقع والخدمات.

العميل والكيانات ذات الصلة

العميل - الكيان المركزي الذي يمثل شركة أو فرد

الحقل	الن��ع	الوصف
customer_id	عدد صحيح (PK)	معرف فريد
customer_name	سلسلة	اسم الشركة أو الفرد
customer_account_type	تعداد	'فردي' أو 'تجاري'
customer_status	تعداد	'مفتوح'، 'مغلق'، 'معلق'، 'أرشي'
customer_payment_type	سلسلة	'مدفوع مسبقًا' أو 'مدفوع لاحقًا'
customer_enabled	منطقي	هل الحساب نشط
tax_identifier	سلسلة	رقم VAT/GST
contract_start_date	تاريخ ووقت	بدء العقد
contract_end_date	تاريخ ووقت	انتهاء العقد

جهة الاتصال - الأشخاص المرتبطون بالعميل

الحقل	النوع	الوصف
contact_id	عدد صحيح (PK)	معرف فريد
customer_id	عدد صحيح (FK)	العميل الرئيسي
contact_firstname	سلسلة	الاسم الأول
contact_lastname	سلسلة	الاسم الأخير
contact_email	سلسلة	عنوان البريد الإلكتروني
contact_phone	سلسلة	رقم الهاتف
contact_types	سلسلة	'أساسي'، 'فوترة'، 'تقني'

الموقع - مواقع تسليم الخدمة الفع��ية

الحقل	النوع	الوصف
site_id	عدد صحيح (PK)	معرف فريد
customer_id	عدد صحيح (FK)	العميل الرئيسي
site_name	سلسلة	اسم الموقع
address_line_1	سلسلة	عنوان الشارع
city, state, zip_code	سلسلة	تفاصيل الموقع
latitude, longitude	عدد عشري	إحداثيات GPS
google_maps_place_id	سلسلة	مرجع خرائط Google
plus_code	سلسلة	رمز الموقع المفتوح

نماذج الخدمة والمنتج

الخدمة - حالات الخدمة النشطة

الحقل	النوع	الوصف
service_id	عدد صحيح (PK)	معرف فريد
customer_id	عدد صحيح (FK)	العميل الرئيسي
product_id	عدد صحيح (FK)	قالب المنتج
site_id	عدد صحيح (FK)	موقع الخدمة
service_name	سلسلة	الاسم المعروض
service_uuid	سلسلة	معرف نظام الفوترة
service_status	تعداد	الحالة الحالية
service_billed	منطقي	توليد المعاملات
wholesale_cost	عدد عشري	تكلفة المزود
retail_cost	عدد عشري	سعر العميل
bundled_parent	عدد صحيح (FK)	الخدمة الرئيسية للحزم

المنتج - قوالب عروض الخدمة

الحقل	النوع	الوصف
product_id	عدد صحيح (PK)	معرف فريد
product_name	سلسلة	الاسم المعروض
product_slug	سلسلة	اسم متوافق مع URL
category	تعداد	'مستقل'، 'حزمة'، 'إضافة'، 'عرض'
provisioning_play	سلسلة	اسم كتاب تشغيل Ansible
provisioning_json_vars	JSON	متغيرات كتاب التشغيل
inventory_items_list	سلسلة	المخزون المطلوب
retail_cost	عدد عشري	السعر الشهري
retail_setup_cost	عدد عشري	رسوم لمرة واحدة
enabled	منطقي	متاح للبيع

نماذج الفوترة

المعاملة - رسوم/ائتمانات فردية

الحقل	النوع	الوصف
transaction_id	عدد صحيح (PK)	معرف فريد
customer_id	عدد صحيح (FK)	العميل الرئيسي
invoice_id	عدد صحيح (FK)	الفاتورة الرئيسية (اختياري)
service_id	عدد صحيح (FK)	الخدمة ذات الصلة
title	سلسلة	وصف المعاملة
retail_cost	عدد عشري	المبلغ (سالب = ائتمان/دفع)
payment_type	سلسلة	مثل `wallet_topup_card`، `wallet_invoice`، `postpaid_deferred_charge`
wallet_ledger_id	عدد صحيح (FK)	يربط إلى wallet_ledger عندما تعكس الصف حركة المحفظة
void	منطقي	معاملة ملغاة

الفاتورة - معاملات مجمعة للفوترة

الحقل	النوع	الوصف
invoice_id	عدد صحيح (PK)	معرف فريد
customer_id	عدد صحيح (FK)	العميل الرئيسي
paid	منطقي	تم استلام الدفع
void	منطقي	فاتورة ملغاة
payment_reference	سلسلة	معرف معاملة Stripe
start_date, end_date	تاريخ	فترة الفوترة
due_date	تاريخ	الموعد النهائي للدفع
retail_cost	عدد عشري	المبلغ الإجمالي

نماذج المخزون

المخزون - الأصول المادية والافتراضية

الحقل	النوع	الوصف
inventory_id	عدد صحيح (PK)	معرف فريد
customer_id	عدد صحيح (FK)	العميل المعين
service_id	عدد صحيح (FK)	الخدمة المرتبطة
inventory_template_id	عدد صحيح (FK)	قالب نوع العنصر
item	سلسلة	نوع العنصر (بطاقة SIM، إلخ، جهاز توجيه.)
item_state	تعداد	'جديد'، 'مخصص'، 'مستخدم'، إلخ.
itemtext1-20	سلسلة	حقول مرنة
management_url	سلسلة	عنوان URL لإدارة المعدات
config_content	نص	ملف التكوين

Inventory_Template - يحدد هيكل عنصر المخزون

الحقل	النوع	الوصف
inventory_template_id	عدد صحيح (PK)	معرف فريد
item	سلسلة	اسم القالب
itemtext1_label	سلسلة	تسمية لحقل itemtext1
itemtext2_label	سلسلة	تسمية لحقل itemtext2

نماذج التزويد

Provision - تتبع وظائف التزويد

الحقل	النوع	الوصف
provision_id	عدد صحيح (PK)	معرف فريد
product_id	عدد صحيح (FK)	المنتج الذي يتم تزويده
customer_id	عدد صحيح (FK)	العميل المستهدف
service_id	عدد ص��يح (FK)	الخدمة التي تم إنشاؤها/تعديلها
provisioning_play	سلسلة	اسم كتاب تشغيل Ansible
provisioning_status	عدد صحيح	نجاح، 1=تشغيل، 2=فشل=0

Provision_Event - خطوات التزويد الفردية

الحقل	النوع	الوصف
provision_event_id	عدد صحيح (PK)	معرف فريد
provision_id	عدد صحيح (FK)	وظيفة التزويد الرئيسية
event_name	سلسلة	اسم المهمة
event_number	عدد صحيح	رقم التسلسل
provisioning_status	عدد صحيح	نجاح، 1=تشغيل، 2=فشل=0
provisioning_result_json	JSON	الإخراج الكامل للمهمة

نماذج المستخدم والأمان

المستخدم - حسابات المستخدمين

الحقل	النوع	الوصف
id	عدد صحيح (PK)	معرف فريد
username	سلسلة	اسم المستخدم لتسجيل الدخول
email	سلسلة	عنوان البريد الإلكتروني
email_verified	منطقي	تم تأكيد البريد الإلكتروني
is_2fa_enabled	منطقي	تم تمكين المصادقة الثنائية
totp_secret	سلسلة	مفتاح TOTP السري

الدور - أدوار المستخدمين

الحقل	النوع	الوصف
id	عدد صحيح (PK)	معرف فريد
name	سلسلة	اسم الدور
description	سلسلة	وصف الدور

الإذن - أذونات دقيقة

الحقل	النوع	الوصف
id	عدد صحيح (PK)	معرف فريد
name	سلسلة	اسم الإذن (resource.action)
description	سلسلة	وصف الإذن

العلاقات:

- لدى المستخدمين العديد من الأدوار (عديد إلى عديد)
- لدى الأدوار العديد من الأذونات (عديد إلى عديد)
- يمكن للمستخدمين الارتباط بجهة اتصال واحدة (للوصول إلى بوابة العملاء)

نقاط التكامل

Stripe بوابة الدفع

- طرق الدفع المعتمدة على الرموز
- تخزين بطاقات متوافقة مع PCI
- دفع الفواتير تلقائيًا
- معالجة المبالغ المستردة
- إشعارات انتهاء صلاحية البطاقة

Mailjet خدمة البريد الإلكتروني

- رسائل بريد إلكتروني معاملات (فواتير، ترحيب، إشعارات)
- مزامنة جهات الاتصال
- إدارة القوالب
- تتبع التسليم

Google خرائط

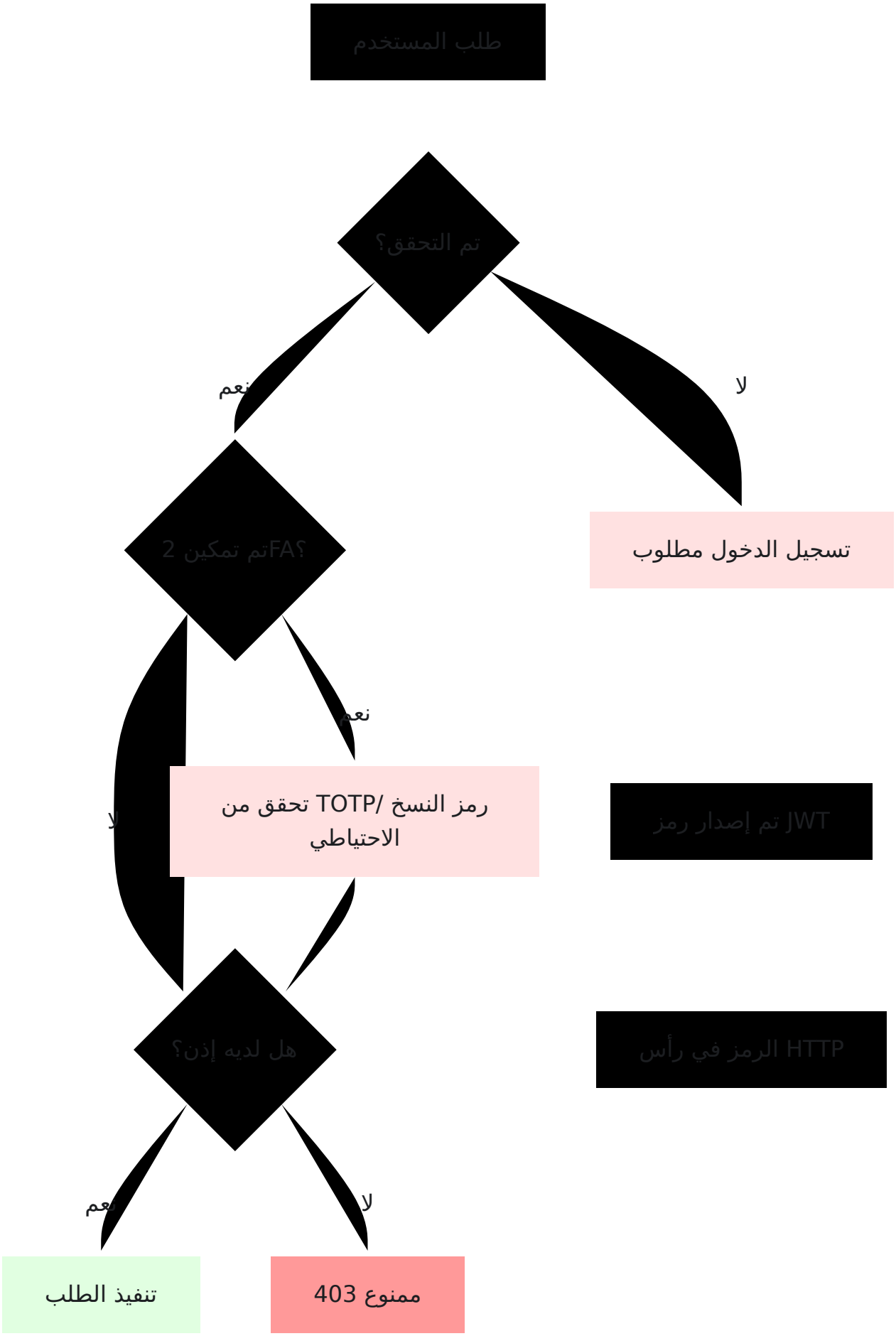
- إكمال العنوان
- التشفير الجغرافي والتشفير العكسي
- توليد رمز زائد
- رسم خرائط مواقع المواقع

OCS/CGRateS

- تزويد الحسابات
- تقييم في الوقت الحقيقي
- إدارة الرصيد
- معالجة CDR

- خطط العمل والجداول الزمنية

بنية الأمان

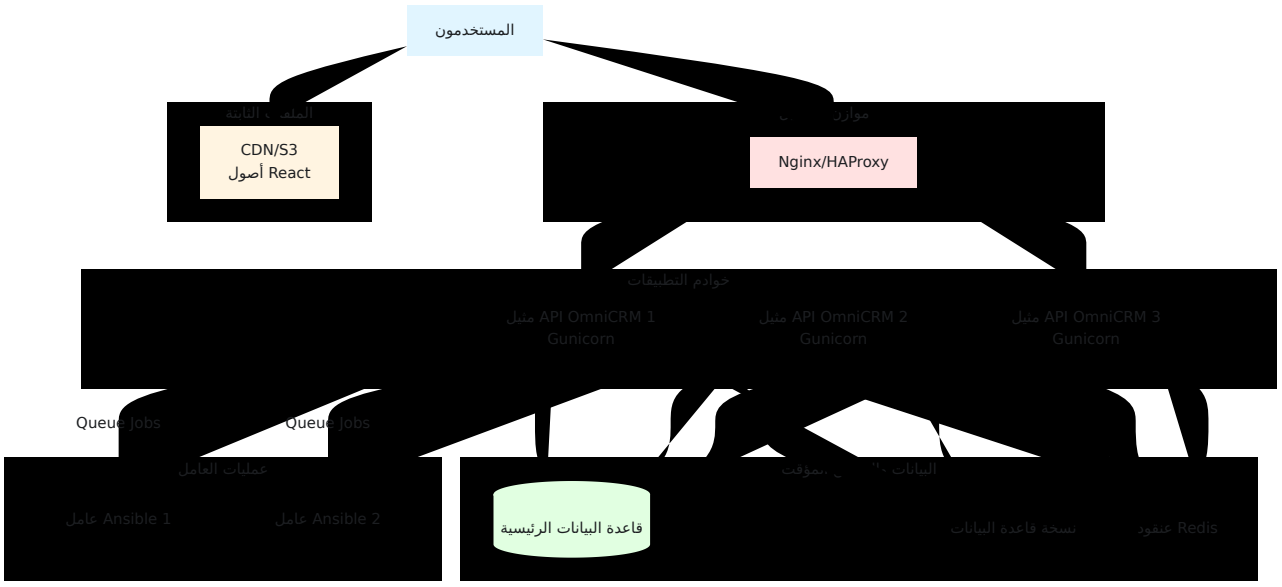

طلب المستخدم
تم التحقق؟
نعم
لا
تسجيل الدخول مطلوب
تم تمكين 2FA؟
نعم
لا
رمز النسخ /TOTP تحقق من الاحتياطي
تم إصدار رمز JWT
هل لديه إذن؟
الرمز في رأس HTTP
نعم
لا
تنفيذ الطلب
ممنوع 403

ميزات الأمان:

- مصادقة تعتمد على JWT
- مصادقة ثنائية (TOTP)
- التحكم في الوصول بناءً على الدور (RBAC)
- الوصول إلى الموارد بن��ءً على الأذونات
- إدارة الجلسات باستخدام Redis
- تجزئة كلمات المرور (bcrypt)
- تحقق من البريد الإلكتروني
- تسجيل التدقيق عبر سجل النشاط

بنية النشر

الإعداد الموصى به للإنتاج:



مجموعة التكنولوجيا

الجزء الخلفي:

- Python 3.x
- إطار عمل Flask
- SQLAlchemy ORM

- ترحيلات Alembic
- للأتمتة Ansible

الواجهة الأمامية:

- React
- إدارة الحالة
- React Router
- لاستدعاءات API Axios

قاعدة البيانات:

- قاعدة بيانات علائقية (مخزن البيانات الرئيسي)
- Redis (التخزين المؤقت والجلسات)

الخدمات الخارجية:

- CGRateS (الفوترة/التقييم)
- Stripe (المدفوعات)
- Mailjet (البريد الإلكتروني)
- خرائط Google (التشفير الجغرافي)

اعتبارات القابلية للتوسع

التوسع الأفقي:

- تصميم API غير متصل يسمح بوجود عدة مثيلات
- موازن التحميل يوزع الطلبات
- لحالة الجلسة المشتركة Redis

توسيع قاعدة البيانات:

- نسخ قراءة لاستعلامات التقارير
- تجميع الاتصالات
- تحسين الاستعلام والفهرسة

التزويد على نطاق واسع:

- معالجة وظائف الخلفية
- عدة عمال Ansible
- إدارة طوابير الوظائف
- منطق إعادة المحاولة للتزويد الفاشل

المراقبة والرؤية

يوفر OmniCRM مقاييس شاملة تعتمد على Prometheus للحصول لمراقبة جميع جوانب النظام. على تفاصيل كاملة، انظر المراقبة والمقاييس.

المقاييس الرئيسية:

- أوقات استجابة API ومعدلات الطلب
- معدلات نجاح/فشل التزويد ومدة الوظائف
- أداء استعلام قاعدة البيانات واستخدام تجمع الاتصالات
- صحة التكامل الخارجي (OCS، Stripe، Mailjet)
- تنفيذ وظائف الخلفية والأخطاء

التسجيل:

- سجلات التطبيق (Flask)
- سجلات التزويد (إخراج Ansible)
- سجل النشاط (مسار التدقيق)
- تتبع الأخطاء والتنبيهات

نقطة نهاية المقاييس: تتعرض جميع المقاييس عند `/crm/metrics` بتنسيق Prometheus. انظر المراقبة والمقاييس لتكوين السحب، وقواعد التنبيه، وأمثلة لوحات المعلومات.

الوثائق ذات الصلة

- كتب تشغيل Ansible - أتمتة التزويد
- نظام التزويد - تفاصيل سير العمل
- المنتجات والخدمات - بنية المنتج
- وثائق API - مرجع API
- RBAC - الأمان والأذونات

تدفقات المصادقة والتحكم الإداري

يوفر OmniCRM ميزات مصادقة شاملة تشمل تسجيل الدخول، المصادقة الثنائية (2FA)، إدارة كلمات المرور، والتحكم الإداري لإدارة أمان المستخدم. تركز هذه الدليل على سير العمل في واجهة المستخدم لكلا من المستخدمين النهائيين والمديرين.

لأذونات RBAC، انظر أيضًا: بوابة الرعاية الذاتية لتسجيل دخول العملاء و الوصول إلى البوابة الموظفين.

نظرة عامة

تشمل مصادقة OmniCRM:

- **تسجيل الدخول باستخدام البريد الإلكتروني/كلمة المرور** - مصادقة قائمة على بيانات الاعتماد القياسية
- **المصادقة الثنائية (2FA)** - عامل ثانٍ اختياري يعتمد على TOTP
- **تذكرني** - جلسة ممتدة تصل إلى 30 يومًا
- **إعادة تعيين كلمة المرور** - استعادة كلمة المرور من خلال الخدمة الذاتية عبر البريد الإلكتروني
- **التحكم الإداري**FA و كلمات المرور 2 - أدوات إدارية لإعادة تعيين
- **تسجيل الدخول الاجتماعي** - تكامل اختياري مع Google وApple وFacebook (إذا تم تمكينه)
- **التنقل القائم على الدور** - توجيه تلقائي بناءً على أدوار المستخدمين

تدفق تسجيل الدخول

تعتبر صفحة تسجيل الدخول نقطة الدخول لجميع المستخدمين (الموظفين والعملاء).

تسجيل الدخول القياسي

عملية تسجيل الدخول:

1. أدخل **عنوان البريد الإلكتروني** (بريد الموظف أو العميل)
2. أدخل **كلمة المرور**
3. اختياري: تحقق من **"تذكرني لمدة 30 يومًا"** لجلسة ممتدة
4. انقر على **"تسجيل الدخول"**

ماذا يحدث بعد ذلك:

- **بدون 2FA:** يتم تسجيل دخول المستخدم على الفور، ويتم التوجيه بناءً على الدور:
  - **العملاء** ← بوابة الرعاية الذاتية (`/self-care`)
  - **الموظفون/المديرون** ← لوحة معلومات العملاء (`/customers`)
  - **CBC وضع** ← واجهة بث الخلايا (`/create-cell-broadcast`)
- **مع تمكين 2FA:** يتم إعادة التوجيه إلى شاشة التحقق من 2FA

ميزة تذكرني:

عند تمكينها، تستمر الجلسة لمدة **30 يومًا** بدلاً من انتهاء صلاحيتها عند إغلاق المتصفح. تستخدم ملفات تعريف الارتباط الآمنة فقط HTTP.

إظهار/إخفاء كلمة المرور:

انقر على **رمز العين (👁)** لتبديل رؤية كلمة المرور.

تسجيل الدخول باستخدام 2FA

إذا كان لدى المستخدم 2FA مفعل، بعد إدخال البريد الإلكتروني/كلمة المرور، تظهر شاشة التحدي 2FA:

استخدام رمز المصادقة:

1. افتح تطبيق المصادقة (Google Authenticator، Authy، إلخ)
2. ابحث عن إدخال OmniCRM
3. أدخل الرمز المكون من 6 أرقام
4. يتم إرسال الرمز تلقائيًا عند إدخال جميع الأرقام الستة
5. إذا كان صالحًا، ي��م تسجيل دخول المستخدم ويتم توجيهه إلى لوحة المعلومات المناسبة

استخدام رمز الاسترداد:

إذا كان تطبيق المصادقة غير متاح:

1. انقر على علامة تبويب **"رمز الاسترداد"**
2. أدخل أحد رموز النسخ الاحتياطي المحفوظة لديك (مثل `fa5b9c2` `3`)
3. انقر على **"تحقق"**
4. يتم استهلاك الرمز (يمكن استخدامه مرة واحدة فقط)

إلغاء:

انقر على **"إلغاء"** للعودة إلى صفحة تسجيل الدخول.

تسجيل الدخول الاجتماعي (اختياري)

إذا تم تمكينه (`VITE_ALLOW_SOCIAL_LOGINS=yes`)، تظهر أزرار تسجيل الدخول الاجتماعي:

> [تسجيل الدخول باستخدام Apple] [تسجيل الدخول باستخدام Google] [تسجيل الدخول باستخدام Facebook]

انقر على أي زر للمصادقة عبر ذلك المزود. حاليًا يعرض رسالة "قريبًا" (تنفيذ تسجيل الدخول الاجتماعي قيد التقدم).

رابط نسيت كلمة المرور

انقر على **"نسيت كلمة المرور؟"** لبدء تدفق إعادة تعيين كلمة المرور.

إعداد المصادقة الثنائية (2FA)

يمكن للمستخدمين تمكين 2FA لتعزيز أمان الحساب. تستخدم 2FA TOTP (كلمة مرور لمرة واحدة تعتمد على الوقت) متوافقة مع تطبيقات المصادقة القياسية.

الوصول إلى إعداد 2FA

من ملف تعريف المستخدم أو الإعدادات:

ملاحظة للعملاء:

2FA مطلوبة فقط للموظفين 2FA. عادة ما تكون 2FA لا يرى مستخدمو دور العميل مطالبات والمستخدمين الإداريين.

الخطوة 1: تأكيد كلمة المرور

> كلمة المرور الحالية
>
> [متابعة] [إلغاء]

أدخل كلمة المرور الحالية الخاصة بك للمتابعة. هذا يتحقق من هويتك قبل تمكين 2FA.

الخطوة 2: مسح رمز الاستجابة السريعة

[تأكيد] [إلغاء]

تعليمات الإعداد:

1. (إذا لم يكن لديك واحد) **قم بتنزيل تطبيق المصادقة**:
   - iOS: متجر تطبيقات Apple → "Google Authenticator"
   - Android: متجر Google Play → "Google Authenticator"
   - بدائل: Authy، Microsoft Authenticator، 1Password
2. **امسح رمز الاستجابة السريعة:**
   - افتح تطبيق المصادقة
   - اضغط على "+" أو "إضافة حساب"
   - اختر "امسح رمز الاستجابة السريعة"
   - وجه الكاميرا نحو رمز الاستجابة السريعة على الشاشة
   - مع رمز مكون من 6 أرقام "OmniCRM" يضيف التطبيق إدخال
3. **احفظ رموز النسخ الاحتياطي:**
   - **حرج:** اكتب أو انسخ هذه الرموز الثمانية
   - خزّنها في مكان آمن (مدير كلمات المرور، خزنة، إلخ)
   - كل رمز يستخدم مرة واحدة فقط
   - يستخدم إذا فقدت الوصول إلى تطبيق المصادقة
   - انقر على **"نسخ الرموز"** لنسخ جميع الرموز إلى الحافظة
4. **تحقق من الإعداد:**
   - أدخل الرمز الحالي المكون من 6 أرقام من تطبيق المصادقة
   - انقر على **"تأكيد"**
   - إذا كان صالحًا، تم الآن تمكين 2FA

الخطوة 3: تم تمكين 2FA

تظهر رسالة النجاح:

2FA. من الآن فصاعدًا، يتطلب تسجيل الدخول كل من كلمة المرور ورمز

تدفق إعادة تعيين كلمة المرور (الخدمة الذاتية)

يمكن للمستخدمين الذين ينسون كلمة الم��ور الخاصة بهم إعادة تعيينها عبر البريد الإلكتروني.

الخطوة 1: طلب رابط إعادة التعيين

من صفحة تسجيل الدخول، انقر على **"نسيت كلمة المرور؟"**

1. أدخل **عنوان البريد الإلكتروني**
2. انقر على **"إرسال رابط إعادة التعيين"**

ماذا يحدث:

- يتحقق النظام مما إذا كان البريد الإلكتروني موجودًا في قاعدة البيانات
- إذا تم العثور عليه، يتم إرسال بريد إلكتروني لإعادة تعيين كلمة المرور عبر Mailjet
- يحتوي البريد الإلكتروني على رابط إعادة تعيين محدود الوقت (عادةً ما ينتهي بعد ساعة)
- تظهر رسالة النجاح: "تم إرسال تعليمات إعادة التعيين إلى بريدك الإلكتروني"

إذا لم يتم العثور على البريد الإلكتروني:

رسالة خطأ: "لا يوجد حساب مرتبط بعنوان البريد الإلكتروني هذا"

الخطوة 2: تحقق من البريد الإلكتروني

يتلقى المستخدم بريدًا إلكترونيًا بعنوان مثل:

مرحبًا [الاسم]،

لقد طلبت إعادة تعيين كلمة المرور لحساب OmniCRM الخاص بك.

انقر على الرابط أدناه لإعادة تعيين كلمة المرور الخاصة بك:
<https://yourcompany.com/reset-password/abc123token456>

ينتهي هذا الرابط خلال ساعة.

إذا لم تكن قد طلبت ذلك، تجاهل هذا البريد الإلكتروني.

انقر على رابط إعادة التعيين للمتابعة.

الخطوة 3: تعيين كلمة مرور جديدة

يفتح رابط إعادة التعيين صفحة إنشاء كلمة المرور:

 (رمز القفل)

كلمة المرور

تأكيد كلمة المرور

[إعادة تعيين كلمة المرور]

1. أدخل **كلمة المرور الجديدة**
2. أعد إدخالها في حقل **تأكيد كلمة المرور**
3. انقر على **"إعادة تعيين كلمة المرور"**

متطلبات كلمة المرور:

- الحد الأدنى للطول (عادةً 8+ أحرف)
- يجب أن تتطابق كلمات المرور

نجاح:

- رسالة النجاح: "تم إعادة تعيين كلمة المرور بنجاح"
- يتم إعادة توجيه المستخدم تلقائيًا إلى صفحة تسجيل الدخول
- يمكن للمستخدم الآن تسجيل الدخول باستخدام كلمة المرور الجديدة

رمز منتهي الصلاحية/غير صالح:

إذا كان رابط إعادة التعيين منتهي الصلاحية أو غير صالح:

> [طلب رابط إعادة تعيين جديد]

التحكم الإداري لإدارة المستخدمين

يمكن للمسؤولين الذين لديهم الأذونات المناسبة إدارة إعدادات مصادقة المستخدم من واجهة إدارة المستخدم.

الوصول إلى إدارة المستخدم

يعرض جدولًا لجميع المستخدمين مع أزرار الإجراءات.

> الاسم البريد الإلكتروني الهاتف الإجراءات جون سميث <john@example.com> +44 123... ✏ 🔒 🔑 جين دو <jane@example.com> +44 456... ✏ 🔒 بوب ويلسون <bob@example.com> +44 789... ✏ 🔒 🔑 ✉

رموز الإجراءات:

- ✏ **تعديل** - تعديل تفاصيل المستخدم، الأدوار، الأذونات
- 🗑 **حذف** - إزالة حساب المستخدم
- 🔑 **إعادة تعيين كلمة المرور** - إنشاء كلمة مرور مؤقتة
- 🔒 **إعادة تعيين 2FA** - تعطيل 2FA للمستخدم (يظهر فقط إذا تم تمكين 2FA)
- ✉ **إرسال بريد الترحيب** - إعادة إرسال بريد الترحيب (يظهر فقط إذا لم يسجل المستخدم الدخول أبدًا)

المسؤول: إعادة تعيين كلمة مرور المستخدم

عندما ينسى المستخدم كلمة المرور ويحتاج المسؤول إلى المساعدة:

الخطوة 1: انقر على رمز إعادة تعيين كلمة المرور (🔑)

تظهر نافذة تأكيد:

> هل أنت متأكد أنك تريد إعادة تعيين كلمة المرور لـ:
>
> المستخدم: جون سميث (<john@example.com>)

> سيتم إنشاء كلمة مرور مؤقتة وعرضها. يجب على المستخدم تغيير هذه الكلمة المرور عند تسجيل الدخول التالي.
>
> [إعادة تعيين كلمة المرور] [إلغاء]

الخطوة 2: تأكيد إعادة التعيين

انقر على **"إعادة تعيين كلمة المرور"**. يقوم النظام بإنشاء كلمة مرور مؤ��تة آمنة.

الخطوة 3: عرض كلمة المرور المؤقتة

> كلمة المرور المؤقتة لجون سميث:
>
> [نسخ كلمة المرور]
>
> ⚠ هام: • أرسل هذه الكلمة المرور إلى المستخدم عبر قناة آمنة • لا ترسل عبر البريد الإلكتروني أو الرسائل غير الآمنة • سيُجبر المستخدم على تغيير كلمة المرور عند تسجيل الدخول التالي
>
> [إغلاق]

إجراء المسؤول:

- انسخ كلمة المرور المؤقتة
- اتصل بالمستخدم أو تواصل عبر طريقة آمنة
- قدم كلمة المرور المؤقتة شفهيًا
- أبلغ المستخدم بتسجيل الدخول وتغيير كلمة المرور

تجربة المستخدم:

عندما يسجل المستخدم الدخول باستخدام كلمة المرور المؤقتة:

1. ينجح تسجيل الدخول
2. يتم إعادة توجيهه على الفور إلى شاشة "تغيير كلمة المرور"
3. يجب تعيين كلمة مرور جديدة قبل الوصول إلى النظام
4. لا يمكن تخطي تغيير كلمة المرور

2 للمستخدم FAالمسؤول: إعادة تعيين

عندما يفقد المستخدم الوصول إلى تطبيق المصادقة وجميع رموز النسخ الاحتياطي:

الخطوة 1: انقر على رمز إعادة تعيين 2FA (🔑)

يظهر 2FA فقط للمستخدمين الذين لديهم 2FA مفعل حاليًا.

تظهر نافذة تأكيد:

الخطوة 2: تأكيد إعادة التعيين

انقر على **"إعادة تعيين 2FA"**

الخطوة 3: تأكيد

رسالة النجاح:

> يمكن لجون سميث الآن تسجيل الدخول باستخدام كلمة المرور فقط. يمكنه إعادة تمكين 2FA من إعدادات المستخدم الخاصة به.

تجربة المستخدم:

- يمكن للمستخدم الآن تسجيل الدخول باستخدام كلمة المرور فقط (لا حاجة لرمز 2FA)
- يختفي رمز درع 2FA (🔑) من صف المستخدم في جدول الإدارة
- يمكن للمستخدم إعادة تمكين 2FA طواعية من إعداداته

ملاحظة أمان هامة:

قبل إعادة تعيين 2FA، يجب على المسؤول:

1. التحقق من هوية المستخدم من خلال وسائل بديلة:
    - التحقق من الهوية الحكومية
    - أسئلة الأمان
    - التحقق من المعاملات الأخيرة
    - التحقق الشخصي (إذا كان ذلك ممكنًا)
2. توثيق إعادة التعيين في ملاحظات العميل
3. إبلاغ المستخدم بإعادة تمكين 2FA بعد استعادة الوصول

المسؤول: إرسال بريد الترحيب

للمستخدمين الذين لم يتلقوا أو فقدوا بريد الترحيب الخاص بهم:

عند توفره:

يظهر رمز الطائرة الورقية (✉) فقط للمستخدمين الذين لم يسجلوا **أبدًا الدخول** (`login_count = 0`).

انقر على رمز إرسال بريد الترحيب (✉)

> إرسال بريد الترحيب إلى:
>
> المستخدم: بوب ويلسون (<bob@example.com>)
>
> سيتضمن البريد الإلكتروني: • رسالة ترحيب • تعليمات تسجيل الدخول • رابط لتعيين كلمة المرور الأولية (إذا كان ذلك ممكنًا) • معلومات الاتصال بالدعم
>
> [إرسال البريد الإلكتروني] [إلغاء]

انقر على "إرسال البريد الإلكتروني"

سالة النجاح:

تم إرسال البريد الإلكتروني عبر Mailjet:

يستخدم القالب: `api_crmCommunicationUserWelcome`

المسؤول: تعديل المستخدم

انقر على **رمز التعديل (✏)** لتعديل تفاصيل المستخدم:

الاسم الأول

الاسم الأخير

البريد الإلكتروني

رقم الهاتف

الأدوار ☑ إداري ☐ وكيل خدمة العملاء_1 ☐ عميل

[حفظ التغييرات] [إلغاء]

الحقول القابلة للتعديل:

- الاسم، البريد الإلكتروني، الهاتف
- **الأدوار** - تعيين/إزالة الأدوار (تؤثر على الأذونات)
- الحالة النشطة/غير النشطة

المسؤول: حذف المستخدم

انقر على **رمز الحذف (🗑)** لإزالة المستخدم:

هل أنت متأكد أنك تريد حذف:

المستخدم: جون سميث (<john@example.com>)

⚠ تحذير: لا يمكن التراجع عن هذا الإجراء.

سيؤدي ذلك إلى حذف: • حساب المستخدم وبيانات الاعتماد • إعدادات 2FA • تاريخ الجلسة

لن يتم حذف بيانات العملاء والمعاملات.

[حذف المستخدم] [إلغاء]

انقر على **"حذف المستخدم"** للتأكيد.

رسالة النجاح:

أفضل الممارسات

للمستخدمين النهائيين

أمان تسجيل الدخول:

- استخدم كلمات مرور قوية وفريدة
- قم بتمكين "تذكرني" فقط على الأجهزة الشخصية
- دائمًا قم بتسجيل الخروج على أجهزة الكمبيوتر المشتركة
- قم بتمكين 2FA لمزيد من الأمان

إدارة 2FA:

- احفظ رموز النسخ الاحتياطي فورًا بعد تمكين 2FA
- خزّن الرموز في مدير كلمات المرور أو مكان آمن
- اختبر رمز النسخ الاحتياطي للتأكد من أنه يعمل
- أعد إنشاء رموز النسخ الاحتياطي إذا استخدمت عدة
- اتصل بالمسؤول إذا فقدت كل من تطبيق المصادقة ورموز النسخ الاحتياطي

إدارة كلمات المرور:

- استخدم مدير كلمات المرور لإنشاء وتخزين كلمات المرور
- لا تشارك كلمات المرور عبر البريد الإلكتروني أو الرسائل
- غير كلمة المرور إذا كنت تشك في تعرضها للاختراق
- استخدم كلمة مرور فريدة لـ OmniCRM (لا تعيد استخدام كلمات المرور)

للمسؤولين

إدارة أمان المستخدم:

- تحقق من هوية المستخدم قبل إعادة تعيين 2FA أو كلمات المرور
- لا ترسل كلمات المرور المؤقتة عبر البريد الإلكتروني
- وثق جميع إعادة تعيين الأمان في ملاحظات المستخدم
- شجع الموظفين على تمكين 2FA
- راقب الأنماط غير العادية لتسجيل الدخول

إعادة تعيين كلمات المرور:

- تواصل مع كلمات المرور المؤقتة عبر الهاتف أو شخصيًا فقط
- أنشئ كلمات مرور مؤقتة قوية (يقوم النظام بذلك تلقائيًا)
- تأكد من أن المستخدم يغير كلمة المرور عند تسجيل الدخول الأول
- لا تعيد تعيين كلمات المرور بلا داعٍ - استخدم تدفق إعادة تعيين البريد الإلكتروني عند الإمكان

إعادة تعيين 2FA:

- اعتبر إعادة تعيين 2FA كإجراءات عالية الأمان
- تحقق من الهوية عبر قنوات م��عددة قبل إعادة التعيين
- وثق سبب إعادة التعيين
- شجع المستخدم على إعادة تمكين 2FA فور استعادة الوصول
- اعتبر فرض 2FA على جميع المستخدمين الإداريين

إدارة المستخدم:

- راجع قائمة المستخدمين بانتظام للبحث عن الحسابات غير النشطة
- أزل المستخدمين الذين غادروا المؤسسة
- تأكد من تعيين الأدوار المناسبة
- راقب المستخدمين الذين لم يسجلوا الدخول أبدًا
- قم بمراجعة أذونات المستخدمين ربع سنويًا

استكشاف الأخطاء وإصلاحها

خطأ "البريد الإلكتروني أو كلمة المرور غير صحيحة"

- **السبب:** بيانات الاعتماد غير صحيحة
- **الإصلاح:**
  - تحقق من أن عنوان البريد الإلكتروني صحيح
  - تحقق من إيقاف تشغيل مفتاح Caps Lock
  - جرب إعادة تعيين كلمة المرور إذا نسيتها
  - اتصل بالمسؤول إذا كان الحساب مقفلاً

رمز 2FA غير مقبول

- **السبب:** مشكلة في مزامنة الوقت أو رمز غير صحيح
- **ال��صلاح:**
  - تأكد من أن وقت الجهاز صحيح (الإعدادات → التاريخ والوقت → تلقائي)
  - انتظر لتحديث الرمز (تتغير الرموز كل 30 ثانية)
  - جرب الرمز التالي الذي يظهر
  - استخدم رمز النسخ الاحتياطي إذا كان تطبيق المصادقة لا يعمل
  - إذا لم تنجح أي من الحلول الأخرى 2FAاتصل بالمسؤول لإعادة تعيين

تذكرني" لا يعمل"

- **السبب:** ملفات تعريف الارتباط معطلة أو تم مسحها
- **الإصلاح:**
  - قم بتمكين ملفات تعريف الارتباط في إعدادات المتصفح
  - لا تقم بمسح ملفات تعريف الارتباط عند إغلاق المتصفح
  - قم بتعطيل ملحقات الخصوصية لنطاق OmniCRM
  - جرب متصفحًا مختلفًا

لم يتم استلام بريد إعادة تعيين كلمة المرور

- **السبب:** البريد الإلكتروني لم يُرسل، فلتر البريد العشوائي، أو البريد الإلكتروني خاطئ
- **الإصلاح:**
  - تحقق من مجلد البريد العشوائي/غير المرغوب فيه
  - تحقق من أن عنوان البريد الإلكتروني صحيح
  - انتظر 5-10 دقائق (قد يتأخر تسليم البريد الإلكتروني)
  - تحقق من أن تكامل Mailjet (المسؤول) يعمل
  - اتصل بالمسؤول لإعادة تعيين كلمة المرور يدويًا

رابط إعادة تعيين كلمة المرور منتهي الصلاحية

- **السبب:** انتهاء صلاحية الرمز (عادةً ما يكون 1 ساعة)
- **الإصلاح:**
  - اطلب إعادة تعيين كلمة مرور جديدة
  - تحقق من البريد الإلكتروني وانقر على الرابط بسرعة
  - اتصل بالمسؤول إذا كانت هناك مشكلات متكررة

2FA لا يمكن تمكين (كلمة مرور غير صحيحة)

- **السبب:** تم إدخال كلمة المرور الحالية بشكل غير صحيح
- **الإصلاح:**
  - تحقق من كلمة المرور الحالية
  - أعد تعيين كلمة المرور أولاً إذا كنت غير متأكد
  - اتصل بالمسؤول للحصول على المساعدة

فقدان تطبيق المصادقة ورموز النسخ الاحتياطي

- **السبب:** فقدان الهاتف/إعادة ضبطه، عدم حفظ رموز النسخ الاحتياطي
- **الإصلاح:**
  - اتصل بالمسؤول على الفور
  - سيقوم المسؤول بالتحقق من الهوية وإعادة تعيين 2FA
  - سجل الدخول باستخدام كلمة المرور فقط بعد إعادة التعيين
  - واحفظ رموز النسخ الاحتياطي هذه المرة 2FAأعد تمكين

خطأ "فشل في إعادة تعيين 2FA" :المسؤول

- **السبب:** أذونات غير كافية
- **الإصلاح:**
  - تأكد من أن لديك دور المسؤول
  - تحقق من أذونات API
  - اتصل بالمسؤول عن النظام

المسؤول: لم يتم إنشاء كلمة مرور مؤقتة

- **السبب:** خطأ في الأذونات API أو مشكلة في
- **الإصلاح:**
  - قم بتحديث الصفحة وحاول مرة أخرى
  - تحقق من أذونات المسؤول
  - تحقق من سجلات API للبحث عن الأخطاء
  - تأكد من أن قاعدة البيانات متاحة

اعتبارات الأمان

إدارة الجلسات:

- تنتهي الجلسات بعد فترة من عدم النشاط
- تذكرني" تمدد الجلسة إلى 30 يومًا"
- يتم تخزين الجلسات كملفات تعريف الارتباط فقط HTTP (غير متاحة لـ JavaScript)
- يضمن العلم الآمن إرسال ملفات تعريف الارتباط فقط عبر HTTPS

أمان كلمة المرور:

- يتم تجزئة كلمات المرور باستخدام خوارزميات قياسية في الصناعة
- لا يتم تخزين كلمات المرور النصية العادية أبدًا
- تنتهي صلاحية كلمات المرور المؤقتة تلقائيًا بعد الاستخدام الأول
- يتم تتبع محاولات تسجيل الدخول الفاشلة (احتمالية تحديد المعدل)

أمان 2FA:

- يتم تشفير أسرار TOTP في قاعدة البيانات
- يتم إنشاء رموز الاستجابة السريعة على جانب العميل عند الإمكان
- يتم تجزئة رموز النسخ الاحتياطي قبل التخزين
- كل رمز نسخ احتياطي يستخدم مرة واحدة فقط

إجراءات المسؤول:

- يتم تسجيل إعادة تعيين 2FA في سجل النشاط
- تخلق إعادة تعيين كلمات المرور مسار تدقيق
- تتطلب إجراءات المسؤول أذونات الدور المناسبة
- يتم تسجيل عناوين IP للأحداث الأمنية

الوثائق ذات الصلة

- `2fa` - مرجع API 2FA (مركز على API) مفصل ل
- `rbac` - التحكم في الوصول القائم على الدور والأذونات
- `administration_configuration` - تكوين بريد Mailjet لإعادة تعيين كلمة المرور
- `integrations_mailjet` - تكوين قالب البريد الإلكتروني
- `customer_care` - بوابة الرعاية الذاتية للعملاء

إنشاء عميل

0fT52ZvoZBE

العملاء، جهات الاتصال، المواقع والخدمات

لدينا نموذج بسيط لـ **عميل** تحت هذا العميل، يمكن أن يكون لديه عدة **جهات اتصال** وعدة **مواقع**، **خدمات**، إلخ.

العميل هو شركة أو فرد لديه علاقة معنا، نرسل له فاتورة / حساب.

جهة الاتصال هي شخص يعمل مع العميل، بالنسبة للفرد، من المحتمل أن يكون نفس العميل، شخص واحد، ولكن قد يكون لدينا أفراد من العائلة أو جهات اتصال أخرى، ولكل جهة اتصال نوع، على سبيل المثال جهة اتصال فواتير، جهة اتصال تقنية، إلخ، مما يؤثر على كيفية تعاملنا مع جهة الاتصال.

الموقع هو موقع مادي حيث نقدم الخدمات، يمكن أن يكون منزلاً أو مكتبًا أو موقعًا آخر. وهذا يسمح لنا بوجود عدة مواقع لعميل واحد، على سبيل المثال، عميل لديه مكاتب متعددة، ومعرفة الخدمات المرتبطة بكل موقع.

الخدمة هي شيء نقوم بفرض رسوم على العميل من أجله، يمكن أن تكون خدمة إنترنت منزلية، خدمة موبايل، أو حتى خدمات مجردة مثل تأجير شبكة فرعية أو توفير كهرباء مقاسة لرف. كل خدمة مرتبطة بعميل وموقع، ويمكن أن يكون لها عدة رسوم مرتبطة بها.

كما أن العملاء لديهم سجل النشاط، وهو سجل لجميع التغييرات التي تم إجراؤها، علامات، سمات لتخزين بيانات التعريف المخصصة، عناصر الجرد ومعلومات مالية مثل المعاملات، الفواتير و طرق الدفع.

بمجرد أن نقوم بإنشاء عميل، يمكننا بعد ذلك إضافة خدمة لذلك العميل، وهي الشيء الذي نفرض عليه رسومًا.

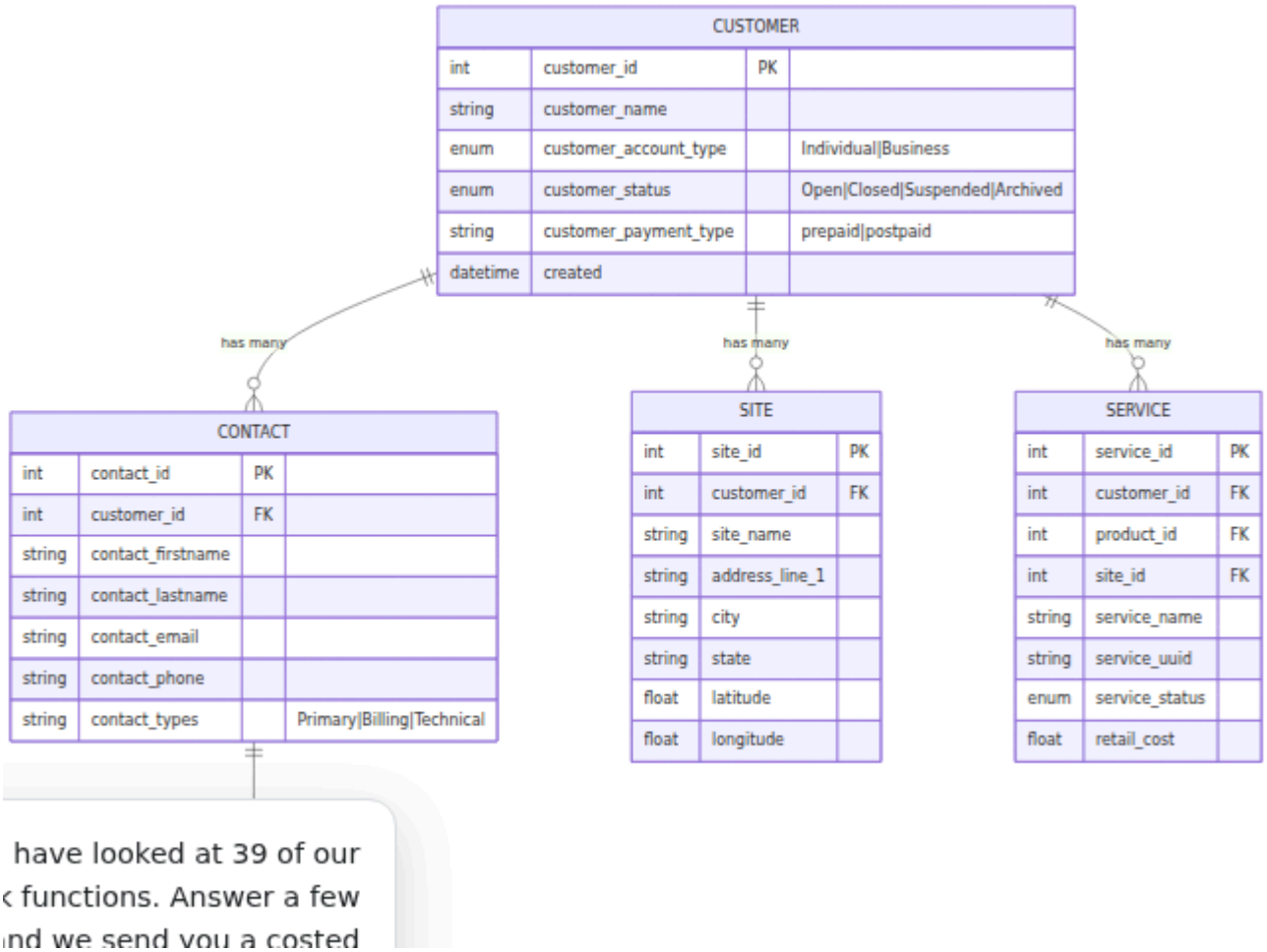
للحصول على معلومات حول إنشاء عميل، انظر إنشاء عميل.

نظرة عامة على نموذج البيانات

يستخدم OmniCRM نموذج بيانات علائقي منظم حول العملاء وخدماتهم. يتم تقسيم النموذج إلى أقسام مركزة أدناه.

العلاقات الأساسية للعميل

العميل هو الكيان المركزي، مع جهات الاتصال والمواقع والخدمات ذات الصلة.

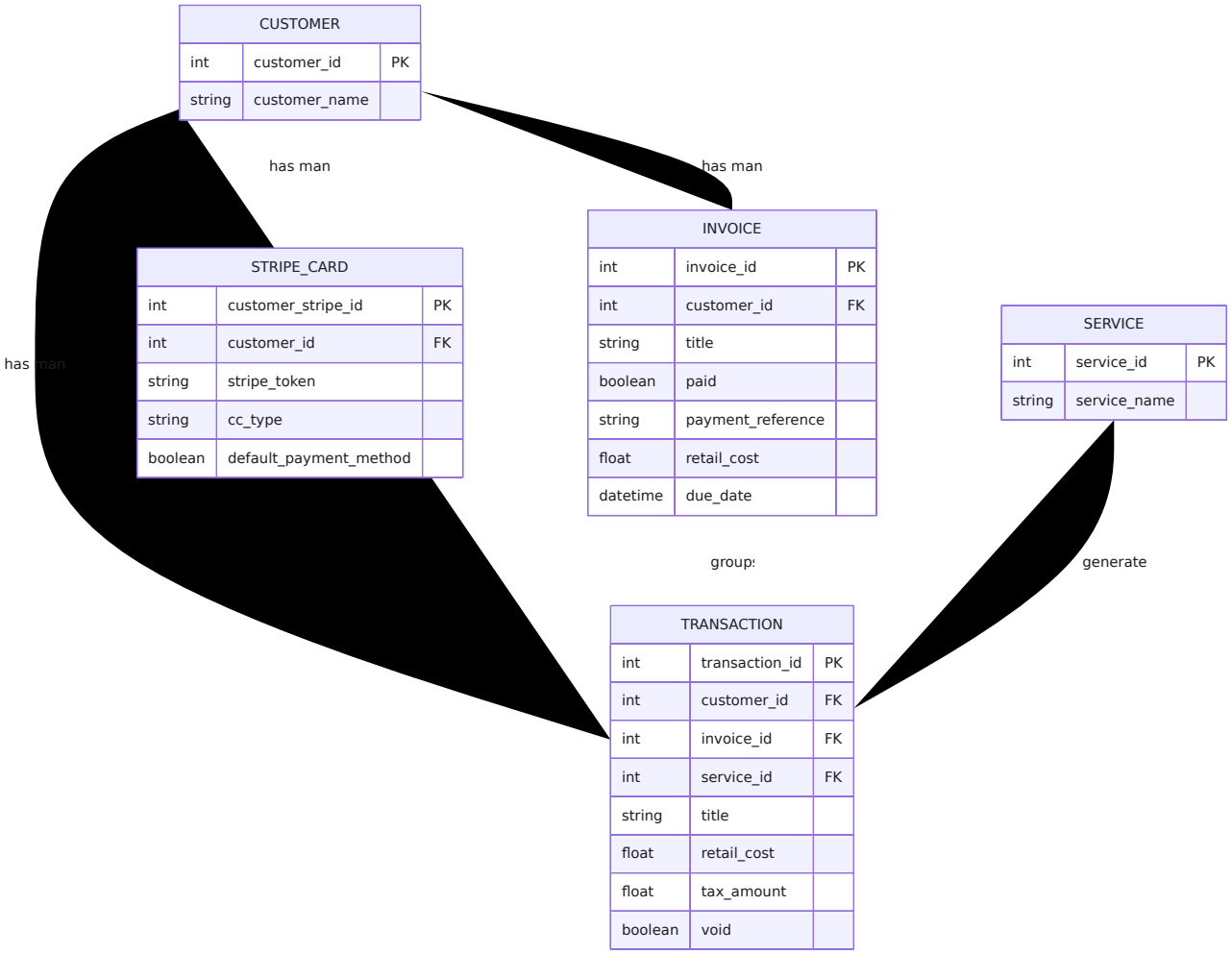


نقاط رئيسية:

- يمكن أن يكون لدى عميل واحد عدة جهات اتصال (فواتير، تقنية، إلخ)
- يمكن أن يكون لدى عميل واحد عدة مواقع (مكاتب فرعية، مواقع)
- يتم تقديم الخدمات إلى المواقع
- يمكن أن تكون جهات الاتصال لديها وصول إلى البوابة عبر حسابات المستخدمين المرتبطة

البيانات المالية والفوترة

تتبع المعاملات والفواتير جميع الأنشطة المالية.

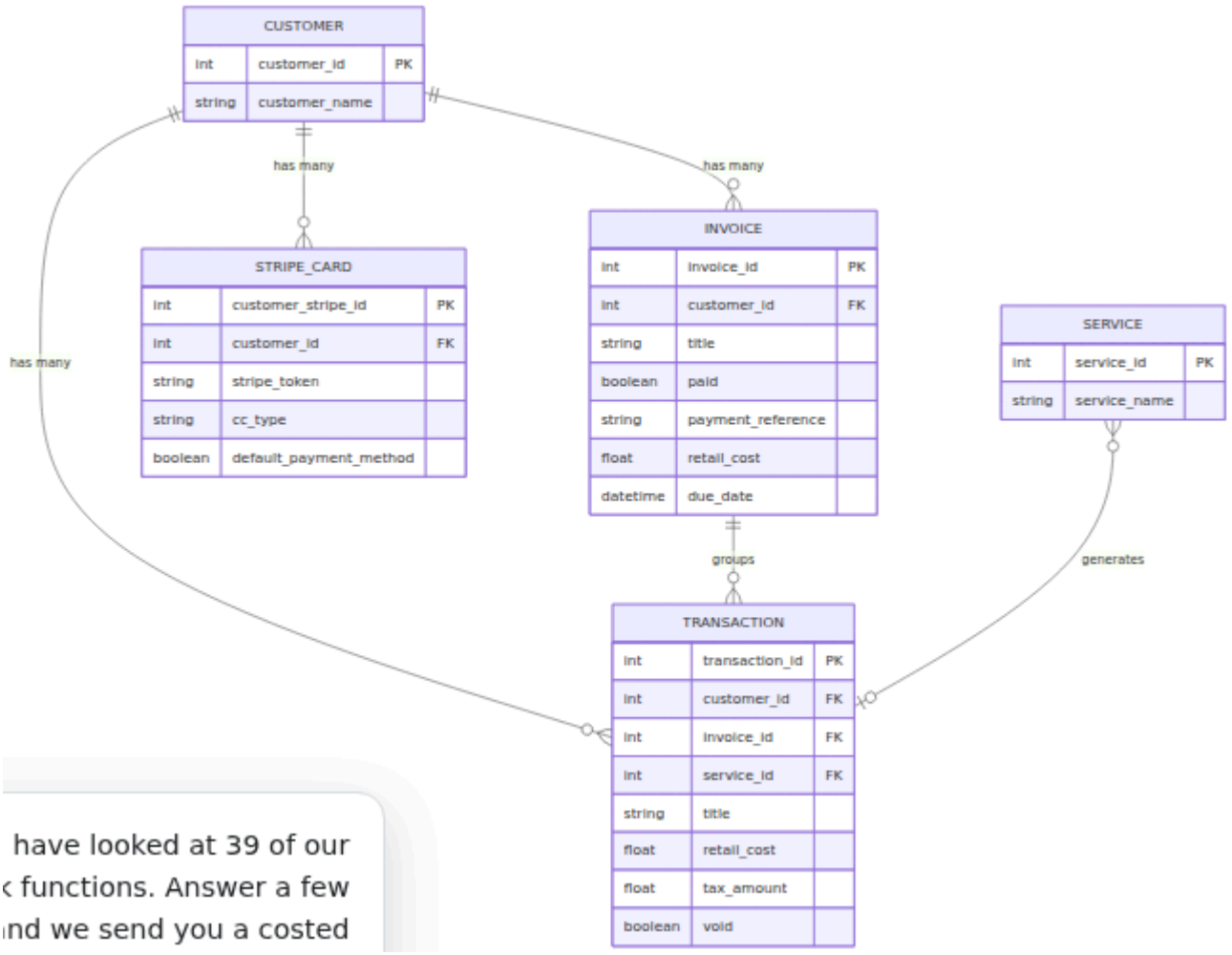


نقاط رئيسية:

- يمكن أن تكون المعاملات مستقلة أو مجمعة في فواتير
- تقوم الخدمات تلقائيًا بإنشاء معاملات شهرية
- وتخزينها بشكل آمن Stripe يتم توكين بطاقات
- تجمع الفواتير عدة معاملات للفوترة

المنتج والتوفير

تحدد المنتجات عروض الخدمة؛ التوفير ينشئ الخدمات الفعلية.

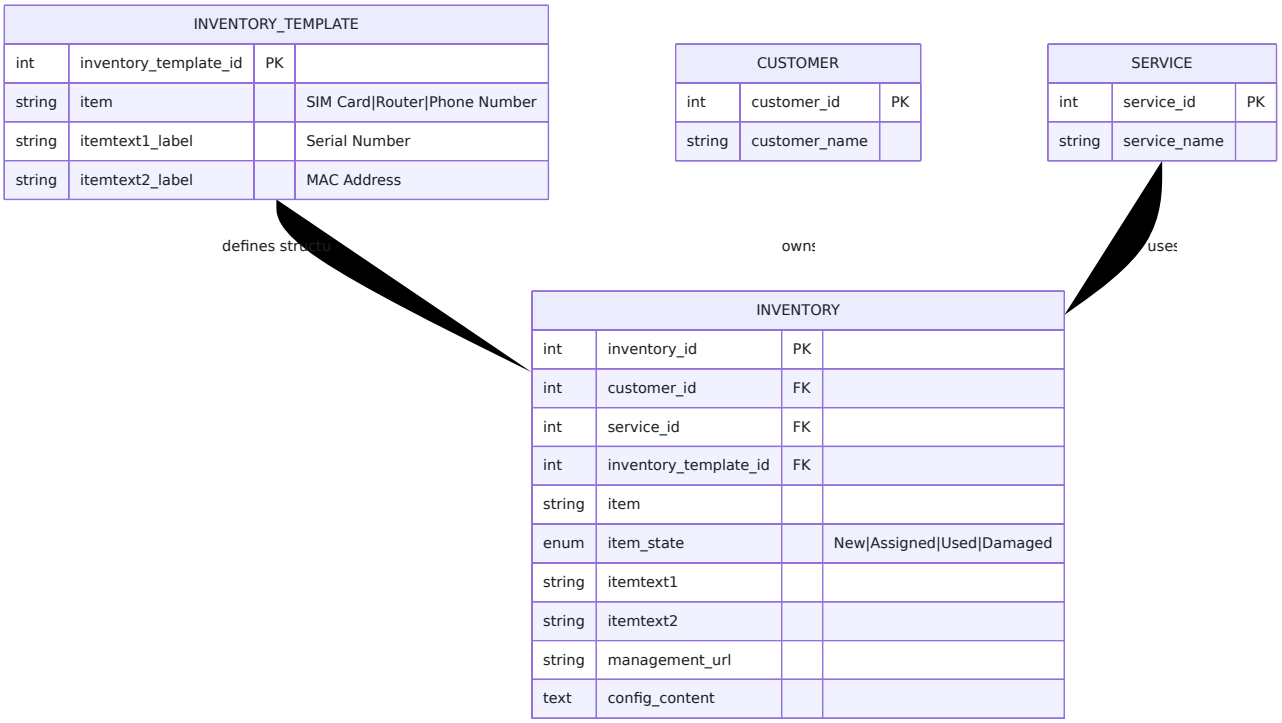


نقاط رئيسية:

- المنتجات هي قوالب؛ الخدمات هي حالات نشطة
- يقوم التوفير بإنشاء أو تعديل الخدمات عبر كتب اللعب Ansible
- تحتوي كل وظيفة توفير على عدة أحداث لتتبع التقدم
- يمكن لوظيفة توفير واحدة إنشاء خدمات متعددة (حزم)

نظام الجرد

ت��يع الأصول المادية والافتراضية المخصصة للعملاء.

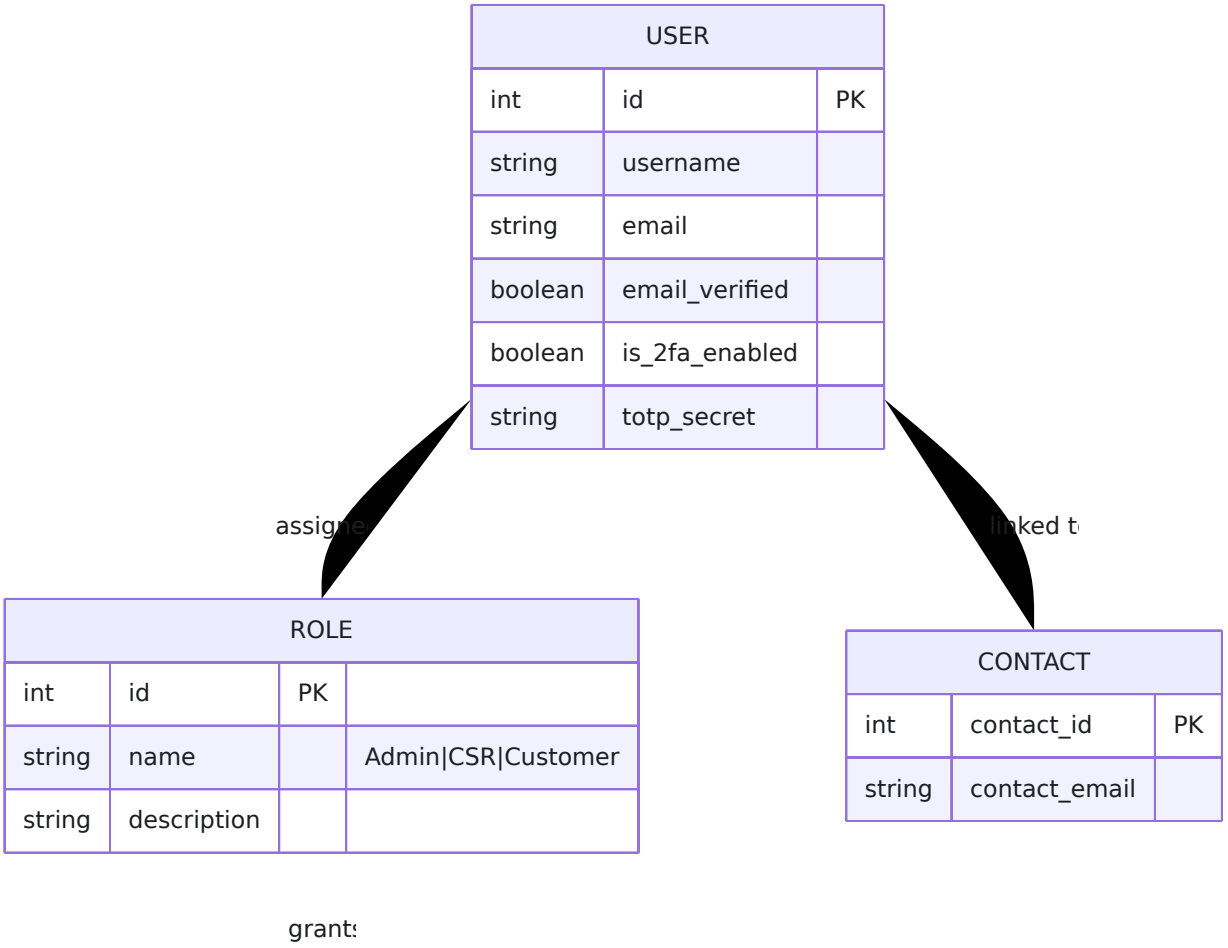


نقاط رئيسية:

- تحدد قوالب الجرد الهيكل (الحقول) لكل نوع عنصر
- تتكيف حقول itemtext1-20 المرنة مع أنواع الجرد المختلفة
- يمكن تخصيص العناصر للعملاء وربطها بالخدمات
- تخزن تكوينات المعدات وبيانات الاعتماد الإدارية

المستخدم والأمان

حسابات المستخدمين مع التحكم في الوصول بناءً على الأدوار.



نقاط رئيسية:

- يمكن أن يكون لدى المستخدمين أدوار متعددة؛ الأدوار لديها أذونات متعددة
- اختيارية لكل مستخدم (2FA) المصادقة الثنائية
- المستخدمون من الموظفين مستقلون؛ المستخدمون من العملاء مرتبطون بجهات الاتصال
- تتحكم الأذونات الدقيقة في الوصول إلى الموارد

قائمة العملاء

توفر قائمة العملاء جدولًا قابل��ا للبحث والتصفية لجميع العملاء في النظام.

المميزات:

- **بحث** - تصفية العملاء حسب الاسم أو المعرف
- **إجراءات جماعية** - تحديد عدة عملاء لعمليات الدفعة
- **ترقيم الصفحات** - التنقل عبر قوائم العملاء الكبيرة
- **إجراءات سريعة** - عرض أو حذف العملاء مباشرة من القائمة

العميل - التفاصيل

لا يحتوي كائن العميل نفسه على الكثير من المعلومات، إنه مجرد اسم ومرجع لجهات الاتصال والمواقع.

قد تتضمن عملية النشر الخاصة بك حقولًا إضافية أو تخصيصات، لكن كائن العميل الأساسي بسيط جدًا.

في صفحة النظرة العامة يوجد أيضًا رسم بياني يظهر متوسط الإيرادات لكل مستخدم (`ARPU`) للعميل، وهو إجمالي الإيرادات مقسومًا على عدد الخدمات، ومقارنة بكيفية مقارنة هذا العميل بمتوسط ARPU لجميع العملاء في النظام.

تكون خيارات حالة العميل مصممة بناءً على الاحت��اجات المحددة لعملك، ولكن عادةً ما تشمل خيارات مثل نشط، غير نشط، معلق، إلخ، مع قواعد مختلفة في كل منها تتحكم في سلوك العميل في النظام في تلك الحالة.

يمكن حذف العميل فقط إذا لم يكن لديه خدمات نشطة، فواتير غير مدفوعة أو معاملات غير مفوترة. إذا كان لدى العميل أي من هذه، ستحتاج إلى إغلاق الخدمات النشطة والتأكد من دفع المدفوعات قبل أن تتمكن من حذف العميل، مما سيؤدي بدوره إلى أرشفة العميل وجميع البيانات المرتبطة، والتي يمكن استعادتها لاحقًا إذا لزم الأمر.

الموقع - التفاصيل

المواقع هي مواقع مادية حيث يتم تقديم الخدمات، ويمكن أن يكون لديها عدة خدمات مرتبطة بها.

تستخدم بشكل أساسي للعملاء التجاريين، حيث قد يكون لدى عميل واحد عدة مواقع، على سبيل المثال، عميل لديه مكاتب متعددة.

يسمح لنا وجود عدة مواقع بتتبع الخدمات المرتبطة بكل موقع، على سبيل المثال إذا طلب عميل خدمة جديدة لمكتب جديد، نحتاج إلى التأكد من أننا نقدم الخدمة الصحيحة للموقع الصحيح. وهذا يسمح لنا بتتبع الخدمات حسب الموقع وفوترتها بشكل منفصل إذا لزم الأمر.

تكامل خرائط Google والتشفير الجغرافي

لضمان دقة بيانات العنوان والموقع الجغرافي. يقوم واجهة **Google** كل موقع متكامل مع **خرائط** المستخدم تلقائيًا بتشفير العناوين وإنشاء بيانات الموقع.

كيف يعمل تشفير العنوان:

عند إضافة أو تعديل موقع، يوفر الواجهة طريقتين لتحديد الموقع:

1. **بحث عن العنوان (موصى به)**
    - استخدم شريط البحث في أعلى النموذج
    - اكتب عنوانًا واقتراحات الإكمال التلقائي من خرائط Google
    - اختر العنوان الصحيح من القائمة المنسدلة
    - يقوم النظام تلقائيًا بملء:
        - اسم المكان من خرائط Google - **اسم الموقع**
        - قم الشارع واسم الشارع - **خط العنوان 1**
        - وحدة/رقم الشقة - **خط العنوان 2**
        - المحلية - **المدينة**
        - المنطقة الإدارية - **الولاية/المنطقة**
        - الرمز البريدي - **الرمز البريدي**
        - اسم الدولة - **الدولة**
        - الإحداثيات الدقيقة - **خط العرض وخط الطول**
        - رمز موقع مفتوح مكون من 11 حرفًا (على سبيل المثال، `C3MFJV8+2F` 8) - **رمز زائد**
        - معرف المكان الفريد - **معرف مكان خرائط Google**
2. **علامة خريطة قابلة للسحب (يدوي)**
    - اسحب الدبوس على الخريطة إلى الموقع الدقيق
    - يقوم النظام بإجراء **تشفير عكسي** للحصول على العنوان من الإحداثيات
    - يتم ملء نفس الحقول تلقائيًا بناءً على موقع الدبوس
    - مفيد للمناطق الريفية أو عندما يكون العنوان غير دقيق

توليد رمز زائد:

يتم توليد رموز الزائد تلقائيًا من خط العرض/خط الطول باستخدام مكتبة رمز الموقع المفتوح. رمز زائد هو رمز قصير (11 حرفًا) يمثل موقعًا دقيقًا في أي مكان في العالم.

مثال:

> خط العنوان 1: "123 Main Street" ✓ الحقول المملوءة تلقائيًا: ✓ اسم الموقع: "123 Main Street" ✓ ✓ المدينة: "لندن" ✓ الولاية: "غرينتش" ✓ الدولة: "المملكة المتحدة" ✓ خط العرض: 51.5074 ✓ خط الطول: -0.1278 ✓ رمز الرمز البريدي: "SW1A 1AA" معرف مكان Google: "ChIJdd4hrwug2EcRmSrV3Vo6lll" ✓ زائد: "9C3XGPHC+3Q"

متطلبات التحقق:

قبل حفظ موقع، يتحقق النظام من:

- **خط العرض وخط الطول** يجب أن يتم تعيينها (إما عبر البحث أو سحب العلامة)
- **الدولة** يجب أن تكون مملوءة (إذا لم يتم تعيينها `VITE_DEFAULT_COUNTRY` تعود إلى)
- **رمز زائد** يجب أن يكون 12 حرفًا (11 + 1 للتعبئة)

إذا فشل التحقق، سترى خطأ:

ردود الفعل المرئية:

تظهر الواجهة ردود فعل في الوقت الفعلي:

أو

كيف يتم استخدام بيانات الموقع

تستخدم بيانات الموقع المشفرة (خط العرض، خط الطول، رمز ز��ئد) في جميع أنحاء OmniCRM لـ:

1. تقديم الخدمة والتركيب

- **الفنيون الميدانيون** - الوصول إلى رمز زائد للتنقل إلى الموقع الدقيق للموقع
- **جدولة التركيب** - تعيين الفنيين بناءً على القرب الجغرافي
- **نشر المعدات** - ضمان تسليم المعدات الصحيحة إلى الموقع الصحيح

2. إشعارات الانقطاع

- **تنبيهات محاطة جغرافيًا** - إذا كان هناك انقطاع في الشبكة في منطقة معينة، استعلام عن المواقع بواسطة دائرة lat/long
- **الاتصالات المستهدفة** - إرسال إشعارات الانقطاع فقط للعملاء المتأثرين عبر Mailjet
- **صفحات الحالة** - عرض خريطة الانقطاع مع المواقع المتأثرة

مثال:

> استعلام: SELECT * FROM Customer_Site
> WHERE distance(latitude, longitude, 51.5074, -0.1278) < 5
>
> النتيجة: 47 موقعًا متأثرًا الإجراء: إرسال إشعار انقطاع إلى 47 عميلًا

3. التقارير والتحليلات

- **الإيرادات الجغرافية** - الإيرا��ات حسب المدينة، الولاية، المنطقة
- **خرائط كثافة الخدمة** - خريطة حرارية لمواقع الخدمة
- **تخطيط التوسع** - تحديد المناطق التي تعاني من نقص الخدمة

4. العملاء التجاريون متعدد المواقع

- **إدارة المواقع** - تتبع أي الخدمات في أي المواقع
- **فواتير منفصلة** - فاتورة حسب الموقع إذا لزم الأمر
- **تعيين الخدمة** - ربط الخدمات بمواقع محددة أثناء التوفير

المواقع الريفية والنائية

بالنسبة للعملاء في المناطق الريفية حيث قد لا توجد عناوين شوارع أو تكون غير دقيقة:

1. **استخدام سحب الخريطة**
   - تكبير المنطقة التقريبية
   - سحب الدبوس إلى العقار/المبنى الدقيق
   - يقوم النظام بإنشاء رمز زائد لذلك الموقع الدقيق
2. **مشاركة رمز زائد**
   - مشاركة رمز زائد مع العميل (على سبيل المثال، `C3MFJV8+2F` `8`)
   - يمكن للعميل إدخال هذا في خرائط Google لرؤية الموقع الدقيق
   - يست��دم الفنيون الميدانيون رمز زائد للتنقل
3. **ملاحظات العنوان**
   - استخدام حقل "ملاحظة العنوان" للتوجيهات الإضافية
   - مثال: "انعطف يسارًا عند الحظيرة الحمراء، 500 متر بعد شبكة الماشية"
   - الملاحظات مرئية لفرق التركيب

تلميح

يمكنك سحب الدبوس على الخريطة إلى الموقع الصحيح إذا لم يكن العنوان دقيقًا. سيقوم النظام بتشفير الموقع عكسيًا وملء جميع حقول العنوان تلقائيًا.

جهة الاتصال - التفاصيل

جهات الاتصال هم الأشخاص المرتبطون بالعميل. يمكن أن يكونوا جهات اتصال فواتير، جهات اتصال تقنية، أو أنواع أخرى، ولكل جهة اتصال نوع يؤثر على كيفية تعاملنا مع جهة الاتصال.

يمكن أن يكون لدينا عدة جهات اتصال لعميل واحد، على سبيل المثال، عميل لديه عدة جهات اتصال فواتير، أو عميل لديه عدة جهات اتصال تقنية.

مثال جيد سيكون شركة مع مزود خدمة مُدار، يتولى الجانب الفني من الأمور، وجهة اتصال فواتير منفصلة تتولى الجانب المالي من الأمور، أو عائلة حيث يمتلك كل فرد جهة اتصال خاصة به ولكن ليس جميعهم مخولين لإجراء تغييرات.

وبالمثل، قد نرغب فقط في إرسال إشعارات الانقطاع إلى جهة الاتصال التقنية، أو إرسال الفواتير فقط إلى جهة الاتصال المالية، ونوع جهة الاتصال يسمح لنا بالتحكم في ذلك.

المنطق الدقيق لكيفية استخدام أنواع جهات الاتصال متروك لعملك، لكن الفكرة الأساسية هي أن كل جهة اتصال لديها نوع يؤثر على كيفية تعاملنا معها، وأن كل شخص مرتبط بالعميل هو جهة اتصال.

يتم مزامنة جهات الاتصال مع تكامل Mailjet، مما يسمح لنا بإرسال حملات بريد إلكتروني مستهدفة بناءً على ��وع جهة الاتصال، الموقع، الإنفاق الشهري، أو الخدمات المشتراة، وإدارة جميع قوالب البريد الإلكتروني المستخدمة في الاتصالات المعاملات.

التنقل

تم تصميم OmniCRM من الألف إلى الياء ليكون استجابة وبديهي.

ستساعدك هذه الدليل في التنقل في النظام والعثور على المعلومات التي تحتاجها.

بمجرد تسجيل الدخول، ستظهر لك شريط التنقل على الجانب الأيسر من الشاشة الأقسام الرئيسية للنظام، والأقسام الفرعية داخل كل منها.

في أعلى يمين الشاشة، سترى قائمة المستخدم، التي تتيح لك تسجيل الخروج، تغيير كلمة المرور الخاصة بك، أو الوصول إلى إعدادات المستخدم الخاصة بك.

لديك روابط لجميع تطبيقات الويب الأكثر استخدامًا في مؤسستك في شريط تطبيقات الويب (يمكن تخصيص هذا وفقًا لاحتياجات مؤسستك).

أي تنبيهات لديك معلقة مرئية في أعلى يمين الشاشة، ويمكنك النقر على التنبيه لعرض المزيد من المعلومات.

يمكننا تغيير لغة النظام من خلال النقر على قائمة اللغة في أعلى يمين الشاشة.

إذا كنت أمير الظلام، يمكنك التبديل إلى الوضع الداكن من خلال النقر على أيقونة القمر في أعلى يمين الشاشة.

المدفوعات والفواتير والمعاملات

معاملات العملاء

أي شيء يكلف المال في النظام يتم تسجيله كمعاملة تحت العميل.

كل معاملة لها مبلغ مالي لتكلفة الجملة وتكلفة التجزئة، ووصف لما تتعلق به المعاملة.

يمكن أن يتم إنشاء المعاملات تلقائيًا بواسطة النظام، على سبيل المثال، عندما يتم توفير خدمة، يتم إنشاء معاملة لتكلفة الإعداد، وعندما يتم فوتره خدمة، يتم إنشاء معاملة لتكلفة التجزئة.

يمكن أيضًا إنشاء المعاملات يدويًا، على سبيل المثال، إذا تم منح العميل ائتمان، يتم إنشاء معاملة لمبلغ الائتمان، أو إذا تم فرض رسوم تركيب، يتم إنشاء معاملة لرسوم التركيب.

تُجمع المعاملات معًا لتشكيل الفواتير، والتي تُرسل إلى العميل للدفع.

الوصول إلى المعاملات

يمكن عرض المعاملات على مستوى النظام أو لكل عميل:

عرض لكل عميل:

1. انتقل إلى **العملاء → [اختر العميل]**
2. انقر على علامة التبويب **الفوترة**
3. عرض قائمة المعاملات في البطاقة الأولى

عرض على مستوى النظام:

1. انتقل إلى **الفوترة → المعاملات** (من القائمة الرئيسية)
2. عرض جميع المعاملات عبر جميع العملاء

أدوات إحصائيات المعاملات

في أعلى صفحة المعاملات، تعرض أربع بطاقات إحصائية ملخصات مالية:

وصف الأدوات:

- **إجمالي المعاملات** - مجموع جميع تكاليف التجزئة للمعاملات (على مر الزمن)
- **إجمالي المعاملات غير المفوترة** - مجموع المعاملات التي لم يتم تضمينها بعد في فاتورة
- **إجمالي المعاملات لهذا الشهر** - مجموع المعاملات التي تم إنشاؤها في هذا الشهر الميلادي
- **إجمالي المعاملات الشهر الماضي** - مجموع المعاملات التي تم إنشاؤها الشهر الماضي

تنسيق القيم:

- القيم التي تزيد عن 1,000: تعرض مع لاحقة "k" (1.5$ ،مثلk)
- القيم التي تزيد عن 1,000,000: تعرض مع لاحقة "M" (2.3$ ،مثلM)
- القيم التي تزيد عن 1,000,000,000: تعرض مع لاحقة "B" (1.1$ ،مثلB)

قائمة المعاملات

تعرض جدول المعاملات جميع المعاملات مع الأعمدة التالية:

وصف الأعمدة:

- **ID** - معرف المعاملة الفريد
- **التاريخ** - تاريخ إنشاء المعاملة
- **العنوان** - اسم المعاملة القصير
- **الوصف** - وصف تفصيلي لما تتعلق به المعاملة
- **المبلغ** - تكلفة التجزئة (إيجابي للرسوم، سلبي للاعتمادات)
- **الفاتورة** - معرف الفاتورة إذا تم فوتره المعاملة (رابط قابل للنقر)
- **الحالة** - علامة تحقق إذا تم فوترتها، شَر��ة إذا لم يتم فوترتها بعد

الإجراءات لكل صف:

كل صف يحتوي على قائمة إجراءات (⋮) مع الخيارات:

- **عرض التفاصيل** - يفتح نافذة تفاصيل المعاملة
- **تنزيل فاتورة PDF** - تنزيل PDF (فقط إذا تم فوترتها)
- **إلغاء المعاملة** - وضع علامة على المعاملة كملغاة (فقط إذا لم يتم فوترتها)

أنواع المعاملات

تقع المعاملات في فئتين رئيسيتين:

معاملات الخصم (الرسوم)

مبالغ إيجابية تزيد من رصيد العميل المستحق:

- **رسوم إعداد الخدمة** - رسوم لمرة واحدة عند توفير الخدمة

- **رسوم الخدمة الشهرية** - رسوم متكررة للخدمات
- **رسوم التركيب** - رسوم لزيارات الفنيين الميدانيين
- **رسوم المعدات** - رسوم للمودمات، أجهزة التوجيه، بطاقات SIM
- **رسوم الدفع المتأخر** - غرامات على الفواتير المتأخرة
- **رسوم يدوية** - رسوم مخصصة يضيفها الموظفون

معاملات الائتمان (المدفوعات/الاستردادات)

مبالغ س��بية تقلل من رصيد العميل المستحق:

- **المدفوعات النقدية** - دفع العميل نقدًا
- **مدفوعات البطاقة** - دفع العميل بواسطة بطاقة ائتمان/خصم
- **مدفوعات التحويل البنكي** - دفع العميل عبر التحويل البنكي
- **اعتمادات الحساب** - اعتمادات حسن النية، تعويض
- **الاستردادات** - الأموال المعادة للعميل
- **الخصومات** - خصومات ترويجية أو ولاء

إضافة معاملة يدويًا

انقر على **"+ إضافة معاملة"** لفتح نافذة إضافة المعاملة.

معاملة خصم (رسوم):

معاملة ائتمان (مدفوعات/استردادات):

وصف الحقول:

- **نوع المعاملة** - اختر خصم (رسوم) أو ائتمان (مدفوعات/استردادات)
- **نوع الائتمان** - إذا تم اختيار ائتمان، اختر طريقة الدفع (نقدي، بطاقة، تحويل بنكي)
- **العنوان** - اسم قصير للمعاملة (مطلوب)
- **الوصف** - شرح تفصيلي (اختياري)
- **تكلفة التجزئة** - المبلغ الذي يدفعه العميل (مطلوب، رقم إيجابي)
- **تكلفة الجملة** - تكلفتك (اختياري، لتتبع الهامش)
- **نسبة الضريبة** - معدل الضريبة المطبق على هذه المعاملة (اختياري، الافتراضي هو ضريبة المنتج أو 0%)
- **الخدمة** - ربط المعاملة بخدمة معينة (اختياري)
- **الموقع** - ربط المعاملة بموقع معين (اختياري)
- **تاريخ المعاملة** - تاريخ المعاملة (الافتراضي هو اليوم)

التحقق:

- العنوان وتكلفة التجزئة مطلوبان
- يجب أن تكون تكلفة التجزئة رقمًا إيجابيًا
- إذا تم اختيار نوع الائتمان، يجب اختيار نوع ائتمان

ماذا يحدث:

1. يتم إنشاء المعاملة في قاعدة البيانات
2. تظهر في قائمة معاملات العميل
3. يتم تضمينها في عدد "المعاملات غير المفوترة"
4. متاحة للتضمين في إنشاء الفاتورة التالية
5. تم إنشاء إدخال في سجل النشاط

البحث وتصفية المعاملات

البحث

استخدم شريط البحث للعثور على المعاملات. يتم البحث عبر:

- معرف المعاملة
- العنوان
- الوصف
- معرف الفاتورة

الفلاتر

تطبيق الفلاتر لتضييق قائمة المعاملات:

الفلاتر المتاحة:

- **حالة الإلغاء** - الكل، ملغاة، غير ملغاة
- **حالة الفاتورة** - الكل، مفوترة، غير مفوترة

إجراءات الفلتر:

- **تطبيق الفلاتر** - تطبيق الفلاتر المحددة على القائمة
- **إعادة تعيين الفلاتر** - مسح جميع الفلاتر وعرض جميع المعاملات

الفرز

انقر على أي رأس عمود للفرز:

- **ID** - فرز حسب معرف المعاملة (الأحدث/الأقدم)

- **التاريخ** - فرز حسب تاريخ المعاملة
- **العنوان** - فرز أبجديًا
- **المبلغ** - فرز حسب تكلفة التجزئة (الأعلى/الأدنى)
- **الفاتورة** - فرز حسب معرف الفاتورة

انقر مرة أخرى لعكس اتجاه الفرز (تصاعدي ↔ تنازلي).

إلغاء المعاملات

يمكن إلغاء المعاملات المضافة عن طريق الخطأ (وضع علامة عليها كملغاة).

المتطلبات:

- يجب ألا تكون المعاملة مفوترة
- بمجرد فوترتها، لا يمكن إلغاء المعاملات (يجب استردادها بدلاً من ذلك)

كيفية الإلغاء:

1. ابحث عن المعاملة في القائمة
2. انقر على قائمة الإجراءات (⋮)
3. اختر "**إلغاء المعاملة**"
4. أكد في النافذة

ماذا يحدث:

- يتم وضع علامة على المعاملة كـ `void = true`
- لم تعد تظهر في قائمة المعاملات الافتراضية
- مستبعدة من إنشاء الفواتير
- يمكن عرضها عن طريق تصفية "المعاملات الملغاة"
- يتم خصمها من إجمالي "المعاملات غير المفوترة"

المعاملات المرتبطة بالمحفظة (تم تعيينه `wallet_ledger_id`):

- يتم عكس رصيد المحفظة قبل الإلغاء (خصم لإلغاء ائتمان، ائتمان لإلغاء خصم)
- **إضافة أموال بطاقة الشحن** (`payment_type=wallet_topup_card`): تم خص�� المحفظة واسترداد البطاقة
- **شحن نقدي / دفع فاتورة المحفظة / خطوط محفظة أخرى:** تم تعديل المحفظة فقط؛ **لا استرداد للبطاقة**
- **محفظة غير كافية:** يفشل الإلغاء إذا كانت الأموال المشحونة قد تم إنفاقها بالفعل (تظل المعاملة نشطة)

في `wallet_ledger_id` **رسوم الخدمة / التوفير** من الالتقاط الهجين عادةً لا تحتوي على اليوم؛ من المخطط الحصول على ائتمان `void=true` خط الفوترة. إلغاء تلك الخطوط يضع فقط كامل للمحفظة لاستردادات الخدمة في فرع الوثائق المالية.

API: `DELETE /crm/transaction/transaction_id/{transaction_id}`. انظر دليل نظام الدفع: إلغاء المعاملة.

ملاحظة: الإلغاء ليس هو نفسه الاسترداد. الإلغاء يعني "هذه المعاملة لم يكن ينبغي أن توجد أبدًا." الاسترداد يعني "عكس معاملة صالحة."

الضريبة على المعاملات

يمكن أن تتضمن المعاملات ضريبة، والتي يتم حسابها تلقائيًا بناءً على إعدادات ضريبة المنتج أو يتم تحديدها يدويًا لكل معاملة.

سلوك الضريبة:

- **معاملات الخصم (الرسوم)**: - يتم تطبيق الضريبة على الرسوم بناءً على
  - **نسبة ضريبة المنتج** - إذا كانت المعاملة مرتبطة بمنتج، يتم تطبيق نسبة ضريبة المنتج تلقائيًا
  - **تجاوز يدوي** - يمكن للموظفين تجاوز نسبة الضريبة عند إنشاء معاملة
  - **مبلغ الضريبة**: - يتم حسابه كالتالي: `retail_cost × (tax_percentage / 100)`
  - **تنسيق العرض** - يظهر كالتالي: `10.00$ (%10)` في قوائم المعاملات
- **معاملات الائتمان (المدفوعات/الاستردادات)** - لا يتم تطبيق ضريبة على الاعتمادات
  - يتم إخفاء حقل نسبة الضريبة للمعاملات الائتمانية
  - يتم تعيين الضريبة تلقائيًا إلى 0% لجميع المدفوعات والاستردادات
  - تقلل الاعتمادات من الرصيد المستحق للعميل دون آثار ضريبية

مثال على حساب الضريبة:

- المنتج: خطة موبايل مع 10% ضريبة، 50.00$ تكلفة التجزئة
- حساب الضريبة التلقائي: 50.00$ × 0.10 = 5.00$
- العرض: 5.00$ (10%)

صفر ضريبة (NIL/معفى):

- يمكن أن تكون المنتجات معفاة من الضريبة عن طريق تعيين نسبة الضريبة إلى 0

- الافتراضي للضريبة هو 0% إذا لم يتم تحديده
- تظهر المعاملات المعفاة من الضريبة كـ "-" في عمود الضريبة

عرض تفاصيل المعاملة

انقر على معاملة لعرض التفاصيل الكاملة:

المعاملات المفوترة مقابل غير المفوترة

المعاملات غير المفوترة:

- لم يتم تضمينها بعد في أي فاتورة
- متاحة لإنشاء الفاتورة التالية
- يمكن إلغاؤها
- تُحسب ضمن إجمالي "المعاملات غير المفوترة"
- الحالة تظهر شَرطة (-)

المعاملات المفوترة:

- تم تضمينها في فاتورة
- لا يمكن إلغاؤها (يجب استردادها إذا لزم الأمر)
- معرف الفا��ورة قابل للنقر (يرتبط بتفاصيل الفاتورة)

- الحالة تظهر علامة تحقق (✓)
- لا يمكن تعديلها

إنشاء الفاتورة:

عند إنشاء فاتورة لعميل:

1. يجد النظام جميع المعاملات غير المفوترة لذلك العميل
2. يمكن تصفية حسب نطاق التاريخ
3. يتم تضمين المعاملات في فاتورة جديدة
4. يتم ملء حقل `invoice_id` للمعاملة
5. يتم الآن وضع علامة على المعاملة كـ "مفوتر"

انظر `payments_invoices` لتفاصيل إنشاء الفاتورة.

سير العمل الشائعة

سير العمل 1: ائتمان يدوي لانقطاع الخدمة

1. يتصل العميل: "كانت الخدمة متوقفة لمدة يومين"
2. يقرر الموظف منح ائتمان بقيمة 10 جنيهات إسترلينية
3. انتقل إلى علامة التبويب **الفوترة** للعميل
4. انقر على "**+ إضافة معاملة**"
5. اختر نوع المعاملة **ائتمان**
6. اختر نوع ائتمان **مدفوعات نقدية**
7. أدخل العنوان: "ائتمان انقطاع الخدمة"
8. أدخل الوصف: "تعويض عن انقطاع لمدة يومين 8-9 يناير"
9. أدخل تكلفة التجزئة: 10.00
10. اختر الخدمة المتأثرة من القائمة المنسدلة
11. انقر على "**إضافة معاملة**"
12. تظهر المعاملة بمبلغ -10.00£
13. سيتم تضمينها في الفاتورة التالية كائتمان

سير العمل 2: رسوم تركيب يدوية

1. يقوم الفني الميداني بتركيب الخدمة
2. يحتاج الموظف إلى فرض رسوم تركيب بقيمة 75 جنيهًا إسترلينيًا
3. انتقل إلى علامة التبويب **الفوترة** للعميل
4. انقر على "**إضافة معاملة +**"
5. اختر نوع المعاملة **خصم**
6. أدخل العنوان: "رسوم التركيب"
7. أدخل الوصف: "زيارة فني ميداني لتركيب الألياف"
8. أدخل تكلفة التجزئة: 75.00
9. أدخل تكلفة الجملة: 45.00 (اختياري، لتتبع الهامش)
10. اختر الخدمة التي تم تركيبها
11. اختر الموقع الذي تم التركيب فيه
12. انقر على "**إضافة معاملة**"
13. تظهر المعاملة في قائمة غير مفوترة
14. سيتم تضمينها في الفاتورة التالية

سير العمل 3: إلغاء معاملة مكررة

1. يلاحظ الموظف معاملة مكررة
2. تحقق من أن المعاملة لم يتم فوترتها بعد
3. انقر على قائمة الإجراءات (⋮) على المعاملة المكررة
4. اختر "**إلغاء المعاملة**"
5. أكد في النافذة
6. تتم إزالة المعاملة من القائمة
7. ينخفض إجمالي غير المفوتر وفقًا لذلك

سير العمل 4: العثور على المعاملات لفاتورة

1. تحتاج إلى إنشاء فاتورة شهرية
2. انقر على **فلتر الفاتورة: "غير مفوترة**"
3. انقر على **تطبيق الفلاتر**
4. عرض جميع المعاملات غير المفوترة
5. لاحظ المبلغ الإجمالي من الأدوات

6. انتقل لإنشاء الفاتورة
7. اختر نطاق التاريخ (مثل، 1-31 يناير)
8. يتم تضمين المعاملات في النطاق في الفاتورة

استكشاف الأخطاء وإصلاحها

لا يمكن إلغاء المعاملة

- **السبب:** تم فوترتها بالفعل
- **الإصلاح:** المعاملة جزء من تاريخ الفواتير. إذا كانت هناك حاجة لاسترداد، قم بإنشاء معاملة ائتمان بدلاً من ذلك.
- **السبب:** فشل عكس شحن المحفظة (رصيد المحفظة غير كافٍ)
- **الإصلاح:** أنفق العميل الأموال المشحونة بالفعل. استخدم خصم/تعديل يدوي للمحفظة أو تدفقات عكس الوثائق المالية؛ لا يمكن إلغاء خط الشحن حتى تغطي المحفظة المبلغ.
- **السبب:** فشل استرداد البطاقة (شحن أموال فقط)
- **الإصلاح:** يتم التراجع عن عكس المحفظة تلقائيًا. حل مشكلة بائع الدفع وحاول إلغاء المعاملة مرة أخرى.

ظهور معاملات مكررة

- **السبب:** تم فرض رسوم على الخدمة عدة مرات أو خطأ في التوفير
- **الإصلاح:** إلغاء المعاملة المكررة إذا لم يتم فوترتها. إذا تم فوترتها، قم بإصدار ائتمان.

المعاملة لا تظهر في القائمة

- **السبب:** تم تطبيق فلاتر أو تم إلغاء المعاملة
- **الإصلاح:** انقر على "إعادة تعيين الفلاتر" لعرض جميع المعاملات. لرؤية المعاملات الملغاة، قم بتصفية "ملغاة: ملغاة".

إجمالي غير المفوتر لا يتطابق مع المتوقع

- **السبب:** تم فوترت بعض المعاملات بالفعل، أو تم استبعاد المعاملات الملغاة

- **الإصلاح:** تطبيق فلتر "فاتورة: غير مفوترة" لرؤية غير المفوتر فقط. تحقق من المعاملات الملغاة بشكل منفصل.

لا يمكن إضافة معاملة (حقل العميل معطل)

- **السبب:** عرض صفحة معاملات محددة للعميل
- **الإصلاح:** تم اختيار العميل مسبقًا. إذا كنت بحاجة إلى إضافة معاملة لعميل مختلف، انتقل إلى صفحة المعاملات على مستوى النظام.

الوثائق ذات الصلة

- `payments_invoices` - إنشاء وإدارة الفواتير
- `payments_process` - معالجة المدفوعات مقابل الفواتير
- `basics_payment` - نظرة عامة على طرق الدفع
- `csa_activity_log` - عرض تاريخ المعاملات في سجل النشاط

فواتير العملاء

تُجمع معًا لتشكيل فاتورة، والتي تُرسل إلى العميل للدفع المعاملات.

تحتوي الفواتير على تاريخ بدء وتاريخ انتهاء، وهو الفترة التي تغطيها الفاتورة، وتاريخ استحقاق، وهو التاريخ الذي تكون فيه الفاتورة مستحقة للدفع.

يمكن أن يتم إنشاء الفواتير تلقائيًا بواسطة النظام، على سبيل المثال، عندما يتم فوتره خدمة، يتم إنشاء فاتورة لتكلفة التجزئة، أو يمكن إنشاؤها يدويًا، على سبيل المثال، إذا طلب العميل نسخة من فاتورة، أو إذا تم فوتره العميل لرسوم لمرة واحدة.

تكون فواتير العملاء مُصممة بالكامل باستخدام Mailjet، ويمكن تخصيصها لتشمل شعار الشركة، العنوان، وتفاصيل الدفع، ويمكن إرسالها إلى العميل عبر البريد الإلكتروني، أو تنزيلها كملف PDF.

تخصيص قوالب الفواتير

يستخدم OmniCRM قوالب HTML مع Jinja2 لتوليد الفواتير. يمكنك تخصيص تصميم الفاتورة بالك��مل، العلامة التجارية، الألوان، والتخطيط.

موقع قالب الفاتورة

تُخزن قوالب الفواتير في `OmniCRM-API/invoice_templates/`

القوالب الافتراضية:

- `sample_invoice_template.html` - قالب فاتورة عينة
- `cifi_invoice_template.html` - مثال على قالب بديل

التكوين:

يتم تحديد قالب الفاتورة النشط في `OmniCRM-API/crm_config.yaml`:

```
invoice:
  template_filename: 'sample_invoice_template.html'
```

المتغيرات المتاحة في القالب

تتوفر لقوالب الفواتير المتغيرات التالية من Jinja2:

معلومات الفاتورة:

- `{{ invoice_number }}` - معرف الفاتورة الفريد (مثل، `INV-2025-001234`)
- `{{ date }}` - تاريخ إصدار الفاتورة (بتنسيق ISO: `2025-01-10T12:00:00`)
- `{{ due_date }}` - تاريخ استحقاق الدفع (مثل، `10-02-2025`)
- `{{ start_date }}` - تاريخ بدء فترة الفوترة
- `{{ end_date }}` - تاريخ انتهاء فترة الفوترة
- `{{ total_amount }}` - المبلغ الإجمالي للفاتورة قبل الضريبة (رقم)
- `{{ total_tax }}` - المبلغ الإجمال�� للضريبة المحسوبة من جميع المعاملات (رقم)

معلومات العميل:

- `{{ client.name }}` - الاسم الكامل للعميل أو اسم الشركة
- `{{ client.address.address_line_1 }}` - عنوان السطر 1
- `{{ client.address.address_line_2 }}` - عنوان السطر 2
- `{{ client.address.city }}` - المدينة
- `{{ client.address.state }}` - الولاية/المقاطعة
- `{{ client.address.zip_code }}` - الرمز البريدي
- `{{ client.address.country }}` - الدولة

عناصر خط المعاملة:

تكرار المعاملات باستخدام:

```
{% for sub_transaction in transactions %}
    <tr>
        <td>{{ sub_transaction.transaction_id }}</td>
        <td>{{ sub_transaction.created.split("T")[0] }}</td>
        <td>{{ sub_transaction.title }}</td>
        <td>{{ sub_transaction.description }}</td>
        <td>${{ "%.2f"|format(sub_transaction.retail_cost) }}</td>
    </tr>
{% endfor %}
```

حقول المعاملة:

- `sub_transaction.transaction_id` - معرف المعاملة
- `sub_transaction.created` - تاريخ/وقت المعاملة
- `sub_transaction.title` - عنوان المعاملة
- `sub_transaction.description` - وصف تفصيلي
- `sub_transaction.retail_cost` - مبلغ خط ��لمعاملة
- `sub_transaction.tax_percentage` - نسبة الضريبة المطبقة (مثل، 10 لـ 10%)
- `sub_transaction.tax_amount` - مبلغ الضريبة المحسوب بالدولار

عرض الضريبة في القوالب:

```
<td>
    {% if sub_transaction.tax_amount and
sub_transaction.tax_amount > 0 %}
        ${{ "%.2f"|format(sub_transaction.tax_amount) }} ({{
sub_transaction.tax_percentage }}%)
    {% else %}
        -
    {% endif %}
</td>
```

إنشاء قالب فاتورة مخصص

الخطوة 1: نسخ القالب الحالي

```
cd OmniCRM-API/invoice_templates/
cp sample_invoice_template.html your_company_invoice_template.html
```

الخطوة 2: تخصيص HTML/CSS

قم بتحرير `your_company_invoice_template.html` لتتناسب مع علامتك التجارية:

مجالات التخصيص الرئيسية:

1. **شعار الشركة والعلامة التجارية**

```
<!-- استبدل برابط شعارك -->
![شركتك](https://yourcompany.com/logo.png)

<!-- تحديث اسم الشركة -->
<h1>اسم شركتك</h1>
```

2. **نظام الألوان**

```
<style>
    /* لون العلامة التجارية الرئيسي */
    .navbar {
        background: linear-gradient(to bottom right,
#your-color-1, #your-color-2);
    }

    /* رؤوس الجدول */
    .table thead th {
        background-color: #your-brand-color !important;
        color: white !important;
    }

    /* الأزرار والروابط */
    .btn-primary {
        background-color: #your-brand-color;
    }
</style>
```

3. **معلومات الشركة في التذييل**

```
<footer>
    <p>اسم شركتك</p>
    <p>123 شارع الأعمال، المدينة، الدولة</p>
    <p>الهاتف: +1-555-123-4567 | البريد الإلكتروني:
billing@yourcompany.com</p>
    <p>الضريبة: رقم ABN/12345678900</p>
</footer>
```

4. **تعليمات الدفع**

```
<div class="payment-info">
    <h3>طرق الدفع</h3>
    <p><strong>الدفع عبر الإنترنت:</strong> على
https://yourcompany.com/pay</p>
    <p><strong>التحويل البنكي:</strong></p>
    <ul>
        <li>اسم الحساب: Your Company Ltd</li>
        <li>BSB: 123-456</li>
        <li>رقم الحساب: 987654321</li>
        <li>المرجع: {{ invoice_number }}</li>
    </ul>
</div>
```

5. **الش��وط والأحك��م**

```
<div class="terms">
    <h4>شروط الدفع</h4>
    <p>الدفع مستحق خلال 30 يومًا من تاريخ الفاتورة.</p>
    <p>رسوم الدفع المتأخر: 2% شهريًا على الأرصدة المتأخرة.
</p>
    <p>للاستفسارات المتعلقة بالفوترة:
billing@yourcompany.com</p>
</div>
```

الخطوة 3: تحديث التكوين

قم بتحرير `OmniCRM-API/crm_config.yaml`:

```
invoice:
  template_filename: 'your_company_invoice_template.html'
```

الخطوة 4: إعادة تشغيل API

```
cd OmniCRM-API
sudo systemctl restart omnicrm-api
```

الخطوة 5: اختبار إنشاء الفاتورة

1. انتقل إلى عميل لديه معاملات
2. أنشئ فاتورة اختبار
3. قم بتنزيل PDF للتحقق من التنسيق
4. أرسل الفاتورة إلى نفسك لاختبار تسليم البريد الإلكتروني

تخصيص متقدم

المحتوى الشرطي:

استخدم الشروط في Jinja2 لإظهار/إخفاء المحتوى:

```
{% if total_amount > 1000 %}
    <div class="high-value-notice">
        <p><strong>ملاحظة:</strong> رصيد كبير - خطة الدفع متاحة عند الطلب.</p>
    </div>
{% endif %}

{% if client.address.country == "Australia" %}
    <p>شامل GST: ${{ "%.2f"|format(total_amount * 0.10) }}</p>
{% endif %}
```

دعم متعدد اللغات:

إنشاء قوالب محددة للغة:

```
invoice_template_en.html
invoice_template_es.html
invoice_template_fr.html
```

تكوين بناءً على تفضيل لغة العميل.

حسابات مخصصة:

```
<!-- عرض المجموع الفرعي وتفاصيل الضريبة -->
<tr>
    <td colspan="4" class="text-right"><strong>المجموع الفرعي:
</strong></td>
    <td>${{ "%.2f"|format(total_amount) }}</td>
</tr>
<tr>
    <td colspan="4" class="text-right"><strong>الضريبة:</strong>
</td>
    <td>${{ "%.2f"|format(total_tax) }}</td>
</tr>
<tr>
    <td colspan="4" class="text-right"><strong>الإجمالي:</strong>
</td>
    <td>${{ "%.2f"|format(total_amount + total_tax) }}</td>
</tr>
```

ملاحظة: يتم حساب متغير `total_tax` تلقائيًا عن طريق جمع `tax_amount` من جميع المعاملات في الفاتورة. يتم حساب ضريبة كل معاملة بناءً على حقل `tax_percentage` الخاص بها، والذي يكون افتراضيًا هو 0% إذا لم يتم تحديده `tax_percentage` للمنتج أو.

رمز QR للدفع:

تم توليد رمز QR للدفع عبر الهاتف المحمول:

```
<div class="qr-payment">
    ![Scan to Pay](https://api.qrserver.com/v1/create-qr-code/?
size=150x150&data={{ payment_url }})
    <p>امسح باستخدام هاتفك للدفع على الفور</p>
</div>
```

أفضل الممارسات لتنسيق PDF

تستخدم OmniCRM **WeasyPrint** لتحويل HTML إلى PDF. اتبع هذه الإرشادات:

CSS المدعومة:

- معظم خصائص CSS 2.1
- CSS3 محدودة (flexbox، بعض التحولات)
- خطوط الويب عبر `@font-face`

غير مدعوم:

- JavaScript
- CSS Grid (استخدم الجداول بدلاً من ذلك)
- الرسوم المتحركة المعقدة
- بعض خصائص CSS الحديثة

حجم الصفحة والهوامش:

```
@page {
    size: A4;
    margin: 1cm;
}

body {
    font-family: Arial, sans-serif;
    font-size: 10pt;
}
```

تنسيق محدد للطباعة:

```
@media print {
    .no-print {
        display: none;
    }

    .page-break {
        page-break-after: always;
    }
}
```

تخطيط الجدول:

```
.table {
    table-layout: fixed;
    width: 100%;
}

.table th, .table td {
    word-wrap: break-word;
    padding: 4px;
}
```

تضمين الخطوط:

للحصول على خطوط مخصصة، استخدم خطوط آمنة على الويب أو قم بالتضمين:

```
@font-face {
    font-family: 'YourFont';
    src: url('https://yourcompany.com/fonts/yourfont.woff2')
format('woff2');
}

body {
    font-family: 'YourFont', Arial, sans-serif;
}
```

اختبار قوالب الفواتير

قائمة التحقق للاختبار:

1. **الفحص البصري:**
    - يظهر الشعار بشكل صحيح
    - تتطابق الألوان مع إرشادات العلامة التجارية
    - النص مقروء (ليس صغيرًا جدًا)
    - تتماشى الجداول بشكل صحيح
    - جميع الأقسام موجودة
2. **دقة البيانات:**
    - تفاصيل العميل صحيحة
    - مجموع المعاملات صحيح
    - تواريخ بتنسيق صحيح
    - جميع المتغيرات تستبدل بشكل صحيح
3. **جودة PDF:**
    - حجم الملف معقول (<5MB)
    - الصور واضحة وحادة
    - لا يوجد قطع أو تجاوز نص
    - فواصل الصفحات في الأماكن المناسبة
4. **الفواتير متعددة الصفحات:**
    - تتكرر العناوين في كل صفحة
    - تظهر أرقام الصفحات
    - قوائم المعاملات الطويلة تتوزع بشكل صحيح
5. **تسليم البريد الإلكتروني:**
    - يتم إرفاق PDF بالبريد الإلكتروني
    - حجم الملف تحت حد Mailjet (15MB)
    - يظهر في Gmail وOutlook وApple Mail

أمر الاختبار (التوليد اليدوي):

يمكنك اختبار توليد الفاتورة عبر API:

```
curl -X GET "http://localhost:5000/crm/invoice/{invoice_id}/pdf" \
     -H "Authorization: Bearer YOUR_TOKEN" \
     --output test_invoice.pdf
```

مشكلات القالب الشائعة

المتغيرات لا تستبدل:

- **السبب:** خطأ مطبعي في اسم المتغير أو بيانات مفقودة
- **الحل:** تحقق من التهجئة بدقة (حساسة لحالة الأحرف)، تحقق من وجود البيانات في قاعدة البيانات

معطل PDF تخطيط:

- **السبب:** خاصية CSS غير مدعومة
- **الحل:** استخدم خصائص CSS 2.1، اختبر مع WeasyPrint متوافق مع CSS

الصور لا تظهر:

- **السبب:** روابط نسبية أو موارد خارجية محجوبة
- **الحل:** استخدم روابط HTTPS المطلقة، تأكد من أن الصور متاحة للجمهور

الجداول تتجاوز الصفحة:

- **السبب:** عرض الأعمدة الثابتة واسع جدًا
- **الحل:** استخدم عرض النسب المئوية، `table-layout: fixed`

الخطوط لا تظهر:

- **السبب:** الخط غير مضمن أو غير متاح
- **الحل:** استخدم خطوط آمنة على الويب (Arial، Times New Roman، إلخ) أو قم بالتضمين بشكل صحيح للخطوط المخصصة

فشل توليد PDF:

- **السبب:** أخطاء في بناء جملة HTML أو تعطل WeasyPrint
- **الحل:** تحقق من سجلات HTML، تبسيط التخطيطات، WeasyPrint المعقدة

تخزين PDF للفواتير

نظام تخزين PDF الزائد، تتضمن OmniCRM لتحسين الأداء وتقليل توليد PDF للفواتير. عند إنشاء PDF للفاتورة لأول مرة، يتم تخزينه في قاعدة البيانات للطلبات اللاحقة.

كيف يعمل تخزين PDF:

1. **الطلب الأول** - عندما يتم طلب PDF للفاتورة (تنزيل أو بريد إلكتروني)، يقوم النظام:
   - إنشاء PDF من قالب الفاتورة
   - ترميز PDF كـ Base64
   - حساب تجزئة SHA256 لمحتوى PDF
   - تخزينه في جدول `Invoice_PDF_Cache` مع:
     - مرجع ID الفاتورة
     - بيانات PDF (مشفرة بـ Base64)
     - اسم الم��ف
     - تجزئة المحتوى (للتحقق من السلامة)
     - طابع زمني للإنشاء
2. **الطلبات اللاحقة** - عندما يتم طلب نفس الفاتورة مرة أخرى:
   - يتحقق النظام من invoice_id المخزن بواسطة PDF
   - إذا كان التخزين موجودًا وصالحًا، يتم إرجاع PDF المخزن على الفور
   - يتم تحديث طابع `last_accessed` لتتبع استخدام التخزين
3. **إبطال التخزين** - يتم إبطال PDFs المخزنة عندما:
   - يتم تعديل الفاتورة (إضافة/إزالة معاملات، تغيير التفاصيل)
   - يتم تحديث قالب الفاتورة
   - يتم تفعيل مسح التخزين يدويًا

الفوائد:

- **الأداء** - تسليم PDF فوري للطلبات المتكررة (لا تأخير في إعادة التوليد)
- **الاتساق** - نفس PDF لجميع تنزيلات الفاتورة (ما لم يتم تعديل الفاتورة)
- **تحميل الخادم** - يقلل من استخدام وحدة المعالجة المركزية من توليد PDF
- **تجربة المستخدم** - يظهر مؤشر تحميل أثناء التوليد الأولي، الطلبات اللاحقة تكون فورية

إدارة التخزين:

للفواتير تلقائيًا بواسطة النظام. يمكن مسح الإدخالات القديمة أو غير PDF يتم إدارة تخزين المستخدمة بشكل دوري بناءً على:

- العمر (على سبيل المثال، إزالة إدخالات التخزين التي تزيد عن 90 يومًا)
- أنماط الوصول (إزالة الإدخالات التي لم يتم الوصول إليها خلال 30 يومًا)
- حدود التخزين (تنفيذ حدود حجم التخزين إذا لزم الأمر)

سلوك API:

عند تنزيل فاتورة عبر API أو واجهة المستخدم:

- الطلب الأول: يظهر مؤشر تحميل أثناء توليد PDF، ثم يتم تخزينه
- الطلبات اللاحقة: تنزيل فوري من التخزين
- ضربة التخزين/عدم الضربة شفافة للمستخدم

مهم: عند تحديث قالب الفاتورة، قم بمسح التخزين لضمان استخدام الفواتير الجديدة التصميم المحدث:

```
-- مسح جميع PDFs المخزنة للفواتير (تشغيل في MySQL)
DELETE FROM Invoice_PDF_Cache;
```

أو تحديث `crm_config.yaml` لإبطال التخزين تلقائيًا عند تغيير القالب.

الوصول إلى الفواتير

يمكن عرض الفواتير على مستوى النظام أو لكل عميل:

عرض لكل عميل:

1. انتقل إلى **العملاء** → **[اختر عميل]**
2. انقر على علامة التبويب **الفوترة**
3. عرض قائمة الفواتير في البطاقة الثالثة

عرض على مستوى النظام:

1. انتقل إلى **الفوترة** → **الفواتير** (من القائمة الرئيسية)
2. عرض جميع الفواتير عبر جميع العملاء

أدوات إحصائيات الفواتير

في أعلى صفحة الفواتير، تعرض أربع بطاقات إحصائية ملخصات مالية.

وصف الأدوات:

- **إجمالي الفواتير** - مجموع جميع تكاليف الفواتير (على مر الزمن) وعدد الفواتير المرسلة
- **الفواتير غير المدفوعة** - مجموع الفواتير التي لم يتم دفعها بعد وعدد الفواتير غير المدفوعة
- **الفواتير هذا الشهر** - مجموع الفواتير التي تم إنشاؤها هذا الشهر الميلادي مع العدد
- **الفواتير الشهر الماضي** - مجموع الفواتير التي تم إنشاؤها الشهر الماضي الميلادي مع العدد

تنسيق القيم:

- القيم التي تزيد عن 1,000: عرض كلاحقة "k" (على سبيل المثال، 1.5$k)
- القيم التي تزيد عن 1,000,000: عرض كلاحقة "M" (على سبيل المثال، 2.3$M)
- القيم التي تزيد عن 1,000,000,000: عرض كلاحقة "B" (على سبيل المثال، 1.1$B)

مؤشرات الاتجاه:

- تعرض الأدوات لـ "هذا الشهر" و "الشهر الماضي" نسبة التغيير
- سهم أخضر لأعلى: زيادة من الفترة السابقة
- سهم أحمر لأسفل: انخفاض من الفترة السابقة

- سهم رمادي لليمين: لا تغيير

قائمة الفواتير

تعرض جدول الفواتير جميع الفواتير مع الأعمدة التالية:

وصف الأعمدة:

- **ID** - معرف الفاتورة الفريد
- **العنوان** - عنوان/وصف الفاتورة
- **الفترة** - فترة الفوترة (تاريخ البدء - تاريخ الانتهاء) أو "N/A" للفواتير ذات الاستخدام الواحد
- **تاريخ الاستحقاق** - تاريخ استحقاق الدفع
- **تاريخ الإنشاء** - تاريخ إنشاء الفاتورة
- **المبلغ** - إجمالي مبلغ الفاتورة (تكلفة التجزئة)
- **الحالة** - مدفوعة، غير مدفوعة، أو مستردة
- **الإجراءات** - الإجراءات المتاحة (تختلف حسب الحالة)

أيقونات الإجراءات:

- **⬇ (تنزيل)** - تنزيل PDF الفاتورة

- 🗑 **(حذف)** - إبطال الفاتورة (فقط إذا لم يتم دفعها)
- 💳 **(دفع)** - دفع الفاتورة عبر الإنترنت (فقط إذا كانت غير مدفوعة)
- ✉ **(بريد إلكتروني)** - إرسال بريد إلكتروني بالفاتورة إلى العميل
- 💸 **(استرداد)** - استرداد دفعة Stripe (فقط للفواتير المدفوعة عبر Stripe)

توليد فاتورة

انقر على **"+ توليد فاتورة مؤقتة"** لإنشاء فاتورة جديدة.

وصف الحقول:

- **البحث عن العملاء** - اختر عميلًا (يظهر فقط في العرض على مستوى النظام، مُعبأ مسبقًا في عرض العميل)
- **العنوان** - عنوان/اسم الفاتورة (اختياري، الافتراضي هو "فاتورة لـ [الفترة]")
- **تاريخ البدء** - بداية فترة الفوترة (الافتراضي هو 14 يومًا مضت)
- **تاريخ الانتهاء** - نهاية فترة الفوترة (الافتراضي هو اليوم)
- **تاريخ الاستحقاق** - الموعد النهائي للدفع (الافتراضي هو اليوم)
- **معاينة المعاملة** - تعرض جميع المعاملات غير المفوترة في نطاق التاريخ مع القدرة على تضمين/استبعاد معاملات معينة

اختيار المعاملة:

- ✓ **(علامة زرقاء)** - انقر لاستبعاد معاملة من الفاتورة

- انقر لتضمين معاملة تم استبعادها سابقًا - **(علامة حمراء) ×**
- **تحديد الكل** - تضمين جميع المعاملات المعروضة
- **مسح الكل** - استبعاد جميع المعاملات
- تظهر المعاملات المستبعدة بلون رمادي مع نص مخطوط
- يتم تحديث الإجماليات في الوقت الحقيقي أثناء تحديد/إلغاء تحديد المعاملات

ما يحدث:

1. يجد النظام جميع المعاملات غير المفوترة للعميل ضمن نطاق التاريخ
2. يعرض معاينة المعاملة مع القدرة على تضمين/استبعاد معاملات فردية
3. يظهر حساب في الوقت الحقيقي للمجموع الفرعي، الضريبة، والإجمالي بناءً على المعاملات المحددة
4. يتم إضافة المعاملات المحددة (المتضمنة) فقط إلى الفاتورة
5. يتم توليد PDF الفاتورة وتخزينه
6. يتم وضع علامة على المعاملات المحددة على أنها مفوترة (تم تعبئته `invoice_id`)
7. تبقى المعاملات المستبعدة غير مفوترة ومتاحة للفواتير المستقبلية
8. تظهر الفاتورة في القائمة بحالة "غير مدفوعة"

حالات الاستخدام النموذجية:

الفوترة الشهرية: تعيين تاريخ البدء إلى الأول من الشهر، وتاريخ الانتهاء إلى آخر يوم من الشهر، تظهر المعاينة جميع المعاملات غير المفوترة من تلك الفترة. اختر الكل أو استبعد يدويًا معاملات معينة.

فاتورة خاصة بالخدمة: استخدم نفس نطاق التاريخ، ثم استبعد يدويًا المعاملات غير المرغوب فيها (على سبيل المثال، استبعاد المعاملات غير المحمولة لإنشاء فاتورة خاصة بالمحمول فقط).

فاتورة ذات استخدام واحد: تعيين كل من تاريخ البدء والانتهاء إلى نفس اليوم، تظهر المعاينة فقط المعاملات من ذلك التاريخ. استبعد أي رسوم غير ذات صلة بهذه الفاتورة المحددة.

عرض تفاصيل الفاتورة

انقر على أي صف فاتورة في الجدول لعرض تفاصيل الفاتورة الكاملة بما في ذلك جميع المعاملات، والإجماليات، والإجراءات المتاحة.

نافذة تفاصيل الفاتورة:

- **معلومات الفاتورة** - تظهر معرف الفاتورة، العنوان، التواريخ، حالة الدفع، وحالة الإبطال
- **قائمة المعاملات** - تعرض جميع المعاملات المضمنة في الفاتورة مع:
  - تاريخ المعاملة
  - العنوان والوصف
  - تكلفة التجزئة
  - مبلغ الضريبة والنسبة (مُنسق كـ `$10.00 (%10)`)
  - تظهر المعاملات المعفاة من الضريبة "-" في عمود الضريبة
- **ملخص الإجماليات** - حساب في الوقت الحقيقي يظهر:
  - عدد المعاملات
  - المجموع الفرعي (مجموع جميع تكاليف التجزئة)
  - الضريبة (مجموع جميع مبالغ الضرائب)
  - إجمالي الفاتورة (المجموع الفرعي + الضريبة)
- **أزرار الإجراءات** - نفس الإجراءات المتاحة كما في الجدول:
  - **تنزيل PDF** - تنزيل PDF الفاتورة (متاح دائمًا)
  - **إرسال بريد إلكتروني** - إرسال الفاتورة إلى العميل (للفواتير غير الملغاة)
  - **دفع الفاتورة** - معالجة الدفع (فواتير غير مدفوعة، غير ملغاة فقط)
  - **استرداد** - استرداد دفعة Stripe (فواتير Stripe فقط المدفوعة)
  - **حذف** - إبطال الفاتورة (فواتير غير مدفوعة، غير ملغاة فقط)

تنزيل PDFs للفواتير

في نافذة تفاصيل الفاتورة **"PDF"** انقر على **أيقونة التنزيل (⬇)** في الجدول أو زر **"تنزيل** لتنزيل فاتورة كـ PDF.

عملية التنزيل:

1. انقر على أيقونة التنزيل بجوار الفاتورة
2. يظهر مؤشر تحميل أثناء التوليد (في المرة الأولى فقط)
3. يطلب المتصفح حفظ الملف: `Invoice_01234.pdf`
4. أو يتم حفظه في مجلد التنزيلات PDF يفتح

سلوك تخزين PDF:

- **أول تنزيل** - يتم توليد PDF من القالب، ويتم تخزينه في قاعدة البيانات (قد يستغرق 2-3 ثوانٍ)
- **التنزيلات اللاحقة** - تنزيل فوري من التخزين
- **إبطال التخزين** - يتم مسح التخزين إذا تم تعديل الفاتورة أو تحديث القالب

استكشاف مشكلات التنزيل:

- **المؤشر لا يتوقف أبدًا** - تحقق من وحدة التحكم في المتصفح، قد يكون API معطلًا
- **فارغ أو تالف PDF** - تحقق من قالب الفاتورة بحثًا عن أخطاء في بناء الجملة
- **فشل التنزيل** - تحقق من إعدادات حظر النوافذ المنبثقة، جرب متصفحًا مختلفًا

دفع الفواتير

انقر على **أيقونة الدفع (💳)** لدفع فاتورة عبر الإنترنت.

عملية الدفع:

1. انقر على أيقونة الدفع على فاتورة غير مدفوعة
2. تفتح نافذة الدفع مع تفاصيل الفاتورة
3. اختر طريقة الدفع:
    - **معاملة Stripe** - خصم بطاقة الائتمان المحفوظة (متاحة لجميع المستخدمين)
    - **نقدًا** - دفع نقدي يدوي (للموظفين فقط)
    - **استرداد** - تطبيق الاسترداد كدفع (للموظفين فقط)
    - **معاملة POS** - محطة نقطة البيع (للموظفين فقط)
    - **تحويل بنكي** - تحويل بنكي يدوي (للموظفين فقط)
4. إذا تم اختيار Stripe:
    - اختر بطاقة من طرق الدفع المحفوظة
    - يتم اختيار البطاقة الافتراضية مسبقًا
    - انقر لاختيار بطاقة مختلفة
5. إذا تم اختيار طريقة أخرى:
    - أدخل رقم المرجع (اختياري)
6. انقر على **"دفع الفاتورة"** لمعالجة
7. يقوم النظام بمعالجة الدفع:
    - **Stripe** - يخصم Stripe البطاقة عبر واجهة برمجة تطبيقات
    - **طرق أخرى** - ينشئ معاملة سلبية لمبلغ الدفع
8. تتغير حالة الفاتورة إلى "مدفوعة"
9. يتم عرض إشعار النجاح

رعاية ذاتية مقابل دفع الموظفين:

:doc:`بوابة الرعاية الذاتية <self_care_portal>` (العملاء):

- Stripe متاح فقط دفع
- يجب أن يكون لديك طريقة دفع محفوظة
- يظهر تحذير إذا لم توجد طرق دفع
- يتم توفير رابط لإضافة طريقة دفع

بوابة الموظفين (المسؤولون):

- جميع طرق الدفع متاحة
- يمكنهم وضع علامة على الفاتورة كمدفوعة يدويًا (نقدًا، POS، تحويل بنكي)
- يمكنهم إدخال أرقام المرجع للتتبع

تحذير طريقة الدفع:

إذا لم يكن لدى العميل أي طرق دفع محفوظة، يتم عرض تحذير يدعوهم لإضافة طريقة دفع قبل أن يتمكنوا من دفع الفواتير.

إرسال الفواتير عبر البريد الإلكتروني

انقر على **أيقونة البريد الإلكتروني (✉)** لإرسال الفاتورة إلى العميل.

ما يحدث:

1. انقر على أيقونة البريد الإلكتروني بجوار الفاتورة

2. يسترجع النظام PDF (أو يولدها إذا لم تكن مخزنة) الفاتورة من التخزين
3. يرسل البريد الإلكتروني عبر Mailjet باستخدام قالب `api_crmCommunicationCustomerInvoice`
4. يتضمن البريد الإلكتروني:
    - PDF الفاتورة كمرفق
    - اسم العميل
    - رقم الفاتورة وتاريخ الاستحقاق
    - المبلغ الإجمالي المستحق
    - رابط لدفع الفاتورة عبر الإنترنت
    - رابط لعرض/تنزيل الفاتورة
5. إشعار النجاح: "تم إرسال بريد الفاتورة بنجاح"

المستلمون عبر البريد الإلكتروني:

يتم إرسال البريد الإلكتروني إلى جميع جهات اتصال العميل من نوع "الفوترة" أو جهة الاتصال الرئيسية إذا لم توجد جهة اتصال فوترة.

متغيرات قالب البريد الإلكتروني:

- `{{ var:customer_name }}` - الاسم الكامل للعميل
- `{{ var:invoice_number }}` - معرف الفاتورة
- `{{ var:invoice_date }}` - تاريخ إصدار الفاتورة
- `{{ var:due_date }}` - تاريخ استحقاق الدفع
- `{{ var:total_amount }}` - المبلغ الإجمالي المستحق
- `{{ var:invoice_url }}` - رابط لعرض/تنزيل PDF
- `{{ var:pay_url }}` - رابط لدفع الفاتورة عبر الإنترنت

استكشاف مشكلات البريد الإلكتروني:

- **لم يتم إرسال البريد الإلكتروني** - تحقق من بيانات اعتماد واجهة برمجة تطبيقات Mailjet في `crm_config.yaml`
- **العميل لا يتلقى** - تحقق من عناوين البريد الإلكتروني لجهات اتصال العميل
- **PDF لا يتم إرفاقه** - تحقق من نجاح توليد PDF (حاول التنزيل أولاً)

إبطال الفواتير

انقر على **أيقونة الحذف (🗑)** لإبطال فاتورة.

المتطلبات:

- يجب أن تكون الفاتورة **غير مدفوعة**
- لا يمكن إبطال الفواتير المدفوعة (يجب استردادها بدلاً من ذلك)

كيفية الإبطال:

1. حدد فاتورة غير مدفوعة في القائمة
2. انقر على أيقونة الحذف (🗑)
3. أكد في النافذة:

ما يحدث:

- يتم وضع علامة على الفاتورة كـ `void = true`
- يتم فصل جميع المعاملات عن الفاتورة (`invoice_id` تم تعيينه إلى null)
- تصبح المعاملات "غير مفوترة" مرة أخرى
- يمكن تضمين المعاملات في فاتورة جديدة
- تظهر الفاتورة في القائمة مع بادئة "إبطال:" في العنوان
- يتم تعطيل إجراءات الفاتورة (لا تنزيل، دفع، أو بريد إلكتروني)
- يمكن عرضها عن طريق تصفية الفواتير "المبطلّة"

ملاحظات مهمة:

- الإبطال ليس هو نفسه الاسترداد
- **الإبطال** = "يجب ألا توجد هذه الفاتورة أبدًا" (خطأ في الفوترة، تكرار)
- **الاسترداد** = "عكس فاتورة مدفوعة صحيحة" (إرجاع المال إلى العميل)

استرداد الفواتير

انقر على **أيقونة الاسترداد (□)** لاسترداد فاتورة مدفوعة.

المتطلبات:

- يجب أن تكون الفاتورة **مدفوعة**
- يجب أن تكون الفاتورة مدفوعة عبر **Stripe**
- يجب أن تحتوي الفاتورة على `payment_reference` صالح (معرف نية الدفع في Stripe)
- متاحة فقط لمستخدمي الموظفين (ليس للرعاية الذاتية)

كيفية الاسترداد:

1. حدد فاتورة Stripe المدفوعة
2. انقر على أيقونة الاسترداد (□)
3. تفتح نافذة تأكيد الاسترداد:

4. انقر على **"تأكيد الاسترداد"**

5. يقوم النظام بمعالجة استرداد Stripe:
    - يستدعي واجهة برمجة تطبيقات Stripe لاسترداد الدفع
    - ينشئ معاملة استرداد في Stripe
    - يحدث الفاتورة بـ `refund_reference`
6. تتغير حالة الفاتورة إلى "مستردة"
7. يتم عرض إشعار النجاح

ما يحدث بعد الاسترداد:

- تبقى الفاتورة في النظام (غير ملغاة)
- تظهر الحالة "مستردة"
- تبقى المعاملات مرتبطة بالفاتورة
- يتلقى العميل استردادًا إلى طري��ة الدفع الأصلية (3-7 أيام عمل)
- معاملة الاسترداد Stripe يظهر لوحة معلومات

قيود الاسترداد:

- لا يمكن استرداد الفواتير المدفوعة نقدًا، أو عبر POS، (يتطلب عكس يدوي)
- لا يمكن الاسترداد الجزئي (مبلغ الفاتورة الكامل فقط)
- لا يمكن الاسترداد مرتين

البحث وتصفية الفواتير

البحث

استخدم شريط البحث للعثور على الفواتير. البحث عبر:

- معرف الفاتورة
- عنوان الفاتورة
- اسم العميل (عرض على مستوى النظام فقط)

الفلاتر

تطبيق الفلاتر لتضييق قائمة الفواتير:

الفلاتر المتاحة:

- **حالة الإبطال** - الكل، مبطل، غير مبطل
- **حالة الدفع** - الكل، مدفوعة، غير مدفوعة بعد

إجراءات الفلترة:

- **تطبيق الفلاتر** - تطبيق الفلاتر المحددة على القائمة
- **إعادة تعيين الفلاتر** - مسح جميع الفلاتر وعرض جميع الفواتير

الفرز

انقر على أي رأس عمود للفرز:

- **ID** - الفرز حسب معرف الفاتورة (الأحدث/الأقدم)
- **العنوان** - الفرز أبجديًا
- **تاريخ الاستحقاق** - الفرز حسب تاريخ الاستحقاق
- **تاريخ الإنشاء** - الفرز حسب تاريخ الإنشاء
- **المبلغ** - الفرز حسب تكلفة التجزئة (الأعلى/الأدنى)
- **الحالة** - الفرز حسب حالة الدفع (المدفوعة أولاً أو غير المدفوعة أولاً)

انقر مرة أخرى لعكس اتجاه الفرز (تصاعدي ↔ تنازلي).

الصفحات

تصفح قوائم الفواتير الكبيرة مع عناصر التحكم في الصفحات التي تظهر الصفحة الحالية، وإجمالي الصفحات، ومحدد العناصر لكل صفحة (10، 25، 50، أو 100 عنصر).

سير العمل الشائع للفواتير

سير العمل 1: الفوترة الشهرية مع معاينة المعاملة

1. يأتي نهاية الشهر (على سبيل المثال، 31 يناير)
2. انتقل إلى **الفوترة** ← **الفواتير**
3. انقر على **"+ توليد فاتورة مؤقتة"**
4. اختر العميل (أو قم بذلك لكل عميل إذا كنت تفوّتر العديد من العملاء)

5. تعيين التواريخ:
    - تاريخ البدء: 01-01-2025
    - تاريخ الانتهاء: 31-01-2025
    - تاريخ الاستحقاق: 15-02-2025 (15 يومًا من الآن)
    - العنوان: "خدمات يناير 2025" (اختياري)
6. يظهر قسم **معاينة المعاملة** مع جميع المعاملات غير المفوترة من يناير
7. مراجعة المعاينة:
    - يتم تضمين جميع المعاملات بشكل افتراضي
    - تحقق من الإجماليات: المجموع الفرعي، الضريبة، وإجمالي الفاتورة
    - تحقق من صحة جميع الرسوم
8. انقر على **"توليد الفاتورة"** (يظهر الزر عدد المعاملات، على سبيل المثال، "توليد الفاتورة (15)")
9. تم إنشاء الفاتورة مع جميع المعاملات المحددة
10. انقر على صف الفاتورة لعرض التفاصيل والتحقق
11. انقر على زر **"إرسال بريد إلكتروني"** في نافذة التفاصيل أو أيقونة البريد الإلكتروني في الجدول
12. يتلقى العميل بريد الفاتورة مع PDF ورابط الدفع

سير العمل 2: الفوترة الانتقائية للمعاملات

1. العميل لديه خدمات متعددة (موبايل + إنترنت) ورسوم متنوعة
2. يريد فواتير منفصلة لكل خدمة
3. **إنشاء الفاتورة الأولى (خدمات الموبايل):**
    - انقر على **"+ إنشاء فاتورة أولية"**
    - العنوان: "خدمات الموبايل - يناير 2025"
    - البداية/النهاية: 1-31 يناير
    - تاريخ الاستحقاق: 15 فبراير
    - في معاينة المعاملات، **استبعد** جميع المعاملات غير المتعلقة بالموبايل:
        - انقر على زر **X** بجوار معاملات الإنترنت
        - انقر على زر **X** بجوار الرسوم المتنوعة
        - تبقى فقط معاملات خدمات الموبايل محددة
    - تحقق من أن الإجماليات تعكس فقط خدمات الموبايل
    - انقر على **"إنشاء فاتورة"** (يظهر عدد معاملات الموبايل)
4. **إنشاء الفاتورة الثانية (خدمات الإنترنت):**

- انقر على **"+ إنشاء فاتورة أولية"** مرة أخرى
- العنوان: "خدمات الإنترنت - يناير 2025"
- البداية/النهاية: 1-31 يناير (نفس الفترة)
- في معاينة المعاملات:
  - معاملات الموبايل التي تم إصدار فاتورة بها (لا تظهر)
  - استبعد الرسوم المتنوعة باستخدام زر **X**
  - تبقى فقط معاملات خدمات الإنترنت
- انقر على **"إنشاء فاتورة"**

5. **إنشاء الفاتورة الثالثة (الرسوم الإضافية):**
   - انقر على **"+ إنشاء فاتورة أولية"** مرة أخرى
   - العنوان: "الرسوم الإضافية - يناير 2025"
   - تظهر فقط الرسوم المتنوعة غير المفوترة في المعاينة
   - انقر على **"تحديد الكل"** لتضمين الكل
   - انقر على **"إنشاء فاتورة"**
6. أرسل جميع الفواتير الثلاثة إلى العميل

سير العمل 3: استبعاد المعاملات المتنازع عليها أو المعلقة

1. وصول نهاية فترة الفوترة
2. انتقل إلى علامة تبويب **الفوترة** للعميل
3. انقر على **"+ إنشاء فاتورة أولية"**
4. حدد تواريخ فترة الفوترة
5. تظهر معاينة المعاملات 20 معاملة
6. العميل قد اعترض على رسم واحد وآخر قيد التحقيق
7. في معاينة المعاملات:
   - حدد المعاملة المتنازع عليها (مثل "رسم تجاوز البيانات")
   - انقر على زر **X** لاستبعادها
   - حدد المعاملة المعلقة (مثل "رسم التركيب")
   - انقر على زر **X** لاستبعادها
   - يتم تحديث عدد المعاملات: "18 معاملة محددة"
   - يتم إعادة حساب الإجماليات تلقائيًا
8. راجع الإجماليات المحدثة (تستبعد المبالغ المتنازع عليها)

9. انقر على **"إنشاء فاتورة (18)"**
10. تم إنشاء الفاتورة مع المعاملات المعتمدة فقط
11. تبقى المعاملات المتنازع عليها/المعلقة غير مفوترة للدورة التالية من الفوترة

سير العمل 4: مراجعة سريعة للفواتير وتعديلها

1. يقوم الموظف بإنشاء فاتورة شهرية
2. تظهر معاينة المعاملات إجماليًا مرتفعًا غير متوقع
3. راجع كل معاملة في المعاينة:
    - لاحظ رسمًا مكررًا لنفس الخدمة
    - انقر على **X** لاستبعاد النسخة المكررة
    - لاحظ معاملة اختبار لا ينبغي فوترة
    - انقر على **X** لاستبعاد معاملة الاختبار
4. يتم تحديث الإجماليات في الوقت الفعلي
5. تحقق من أن الإجمالي الجديد يتطابق مع المبلغ المتوقع
6. انقر على **"إنشاء فاتورة"** مع المعاملات المصححة
7. ارجع وألغِ/احذف المعاملات المستبعدة إذا لزم الأمر
8. أرسل الفاتورة إلى العميل بثقة

سير العمل 5: فاتورة تركيب لمرة واحدة

1. يقوم الفني الميداني بإكمال التركيب
2. يضيف الموظف معاملة التركيب يدويًا
3. انتقل إلى علامة تبويب **الفوترة** للعميل
4. انقر على **"إنشاء فاتورة أولية +"**
5. حدد التواريخ:
    - تاريخ البدء: اليوم
    - تاريخ الانتهاء: اليوم
    - تاريخ الاستحقاق: اليوم + 7 أيام
    - العنوان: "خدمات التركيب"
6. تظهر معاينة المعاملات فقط معاملات اليوم
7. تحقق من ظهور رسم التركيب
8. استبعد أي رسوم متكررة باستخدام زر **X** (إذا كانت موجودة)
9. انقر على **"إنشاء فاتورة"**

10. أرسل إلى العميل على الفور
11. يدفع العميل عبر الإنترنت عبر Stripe

سير العمل 6: مراجعة الفاتورة قبل الاتصال بالعميل

1. يتصل العميل بسؤال حول الفوترة
2. ينتقل الموظف إلى قائمة فواتير العميل
3. **انقر على صف الفاتورة** لفتح نافذة تفاصيل الفاتورة
4. راجع معلومات الفاتورة:
    - معرف الفاتورة، التواريخ، الحالة
    - جميع المعاملات المدرجة مع الوصف
    - تفصيل الضرائب لكل معاملة
    - المجموع الفرعي، الضريبة، والمجموع الكلي
5. أجب عن أسئلة العميل بالتفاصيل الدقيقة
6. إذا طلب العميل PDF، انقر على زر **"تنزيل PDF"** في النافذة
7. إذا طلب العميل إعادة إرسال البريد الإلكتروني، انقر على زر **"إرسال بريد إلكتروني"**
8. أغلق النافذة عند الانتهاء

سير العمل 7: تصحيح خطأ الفوترة

1. يبلغ العميل عن رسم غير صحيح
2. ينقر الموظف على صف الفاتورة لعرض التفاصيل
3. يراجع قائمة المعاملات في نافذة تفاصيل الفاتورة
4. يحدد المعاملة غير الصحيحة
5. الفاتورة غير مدفوعة، لذا يمكن إلغاؤها
6. انقر على زر **"حذف"** في تذييل النافذة
7. أكد الإلغاء
8. تصبح المعاملات غير مفوترة مرة أخرى
9. يعدل الموظف أو يزيل المعاملة غير الصحيحة من قائمة المعاملات
10. إنشاء فاتورة جديدة مع المعاملات المصححة:
    - استخدم معاينة المعاملات لاستبعاد المعاملة المصححة إذا لزم الأمر
    - تضمين الرسوم الصالحة فقط
11. أرسل الفاتورة المصححة إلى العميل

سير العمل 8: معالجة مدفوعات متعددة

1. يأتي العميل بالنقد لدفع فواتير متعددة
2. انتقل إلى علامة تبويب **الفوترة** للعميل
3. عرض الفواتير غير المدفوعة
4. انقر على صف الفاتورة الأولى لعرض التفاصيل
5. تحقق من المبلغ والمعاملات
6. انقر على زر **"دفع الفاتورة"** في تذييل النافذة
7. اختر طريقة الدفع **"نقد"**
8. أدخل المرجع: "تم الدفع نقدًا 15-01-2025"
9. انقر على **"دفع الفاتورة"**
10. تغلق النافذة، ويتم وضع علامة على الفاتورة كـ "مدفوعة"
11. كرر لبقية الفواتير
12. جميع الفواتير الآن موضوعة كـ "مدفوعة"

سير العمل 9: التعامل مع طلب استرداد

1. يطلب العميل استردادًا للمبلغ المدفوع الزائد
2. يتحقق الموظف من أن الفاتورة قد تم دفعها عبر Stripe
3. انتقل إلى الفاتورة في القائمة
4. انقر على صف الفاتورة لعرض التفاصيل
5. تحقق من معلومات الدفع والمبلغ
6. انقر على زر **"استرداد"** في تذييل النافذة (يظهر فقط لفواتير Stripe)
7. أكد الاسترداد
8. يقوم النظام بمعالجة استرداد Stripe
9. تتغير حالة الفاتورة إلى "تم استردادها"
10. يتلقى العميل الاسترداد في غضون 3-7 أيام عمل
11. يتابع الموظف مع العميل لتأكيد الاستلام

استكشاف الأخطاء وإصلاحها

لا يمكن إنشاء الفاتورة - لا توجد معاملات

- **السبب:** لا توجد معاملات غير مفوترة في النطاق الزمني المحدد

- **الحل:** تحقق من قائمة المعاملات، تحقق من وجود المعاملات وأنها ليست مفوترة بالفعل. ضبط النطاق الزمني أو إزالة الفلتر.

فشل إنشاء PDF الفاتورة

- **السبب:** خطأ في بناء جملة القالب، تعطل WeasyPrint، أو بيانات العميل مفقودة
- **الحل:** تحقق من قالب HTML للفواتير بحثًا عن أخطاء، تحقق من ملء حقول عنوان العميل، راجع سجلات API.

فشل الدفع مع خطأ Stripe

- **السبب:** تم رفض البطاقة، عدم كفاية الأموال، انتهاء صلاحية البطاقة، أو مشكلة في واجهة برمجة التطبيقات Stripe
- **الحل:** جرب طريقة دفع مختلفة، تحقق من صحة البطاقة، تحقق من لوحة معلومات Stripe لمعرفة سبب الرفض.

لا يمكن إلغاء الفاتورة

- **السبب:** تم دفع الفاتورة بالفعل
- **الحل:** لا يمكن إلغاء الفواتير المدفوعة. إذا كان الاسترداد مطلوبًا، استخدم وظيفة الاسترداد Stripe لفواتير أو أنشئ معاملة ائتمانية يدويًا.

البريد الإلكتروني للفواتير لا يُرسل

- **السبب:** بيانات اعتماد واجهة برمجة التطبيقات Mailjet غير صحيحة، العميل ليس لديه جهة اتصال فوترة، أو قالب البريد الإلكتروني مفقود
- **الحل:** تحقق من تكوين Mailjet في `crm_config.yaml`، تحقق من جهات اتصال العميل، تحقق من وجود قالب البريد الإلكتروني للفواتير.

زر الاسترداد لا يظهر

- **السبب:** تم دفع الفاتورة نقدًا/نقاط البيع/تحويل بنكي (ليس Stripe)، أو الفاتورة لم تُدفع
- **الحل:** يظهر زر الاسترداد فقط لمدفوعات Stripe. بالنسبة لطرق الدفع الأخرى، أنشئ معاملة ائتمانية يدوية.

تنزيل PDF يظهر تصميم القالب القديم

- **السبب:** تم تخزين PDF في ذاكرة التخزين المؤقت قبل تحديث القالب

- الفاتورة PDF: امسح ذاكرة التخزين المؤقت لـ **الحل:** `DELETE FROM Invoice_PDF_Cache WHERE invoice_id = X;`

لا يمكن للعميل دفع الفاتورة (لا توجد طرق دفع)

- **السبب:** لا توجد طرق دفع محفوظة في بوابة الخدمة الذاتية
- **الحل:** يجب على العميل إضافة بطاقة ائتمان في صفحة **طرق الدفع** قبل دفع الفواتير.

تم إنشاء فواتير متعددة لنفس الفترة

- **السبب:** قام الموظف بإنشاء فاتورة مرتين، أو تتداخل النطاقات الزمنية
- **الحل:** ألغِ الفاتورة المكررة. ضبط النطاقات الزمنية لمنع التداخل. استخدم معاينة المعاملات لضمان مجموعات المعاملات الفريدة.

لا تظهر أي معاملات في معاينة المعاملات

- **السبب:** جميع المعاملات في النطاق الزمني تم إصدار فواتير بها بالفعل أو لا توجد معاملات
- **الحل:** تحقق من صحة النطاق الزمني. تحقق من قائمة المعاملات للتأكد من وجود المعاملات. قم بتصفية الفواتير لمعرفة أي فاتورة تحتوي على المعاملات.

لا يمكن استبعاد المعاملة من إنشاء الفاتورة

- **السبب:** تم إصدار الفاتورة بالفعل أو مشكلة في المتصفح
- **الحل:** تحقق من أن المعاملة تظهر في المعاينة مع علامة الاختيار. قم بتحديث الصفحة وحاول مرة أخرى. امسح ذاكرة التخزين المؤقت للمتصفح إذا استمرت المشكلة.

إجمالي الفاتورة لا يتطابق مع المبلغ المتوقع

- **السبب:** تم تضمين معاملات غير متوقعة، لم يتم حساب الضريبة، أو تم احتساب المعاملات المستبعدة
- **الحل:** راجع معاينة المعاملات بعناية. تحقق من تكلفة التجزئة والضريبة لكل معاملة. تحقق من أن المعاملات المستبعدة مظللة. تحقق من شارة عدد المعاملات على زر إنشاء الفاتورة.

زر إنشاء الفاتورة معطل

- **السبب:** لم يتم اختيار أي معاملات أو نطاق زمني غير صالح

- **الحل:** تأكد من تضمين معاملة واحدة على الأقل (وليس مستبعدة). تحقق من أن تاريخ البدء قبل تاريخ الانتهاء. تحقق من تعيين تاريخ الاستحقاق.

نافذة تفاصيل الفاتورة لا تفتح

- **السبب:** خطأ في JavaScript أو الصفحة لم يتم تحميلها بالكامل
- **الحل:** قم بتحديث الصفحة. تحقق من وحدة التحكم في المتصفح بحثًا ع�ن الأخطاء. جرب متصفحًا مختلفًا. تحقق من الاتصال بالإنترنت.

الضريبة على المعاملة لا تظهر في تفاصيل الفاتورة

- **السبب:** المعاملة لديها ضريبة 0% أو tax_amount فارغ
- **الحل:** تحقق من أن المعاملة تحتوي على tax_percentage محدد. تحقق من أنه تم حساب tax_amount عند إنشاء المعاملة. قم بتحديث المعاملة إذا لزم الأمر.

أزرار الإجراءات مفقودة في نافذة تفاصيل الفاتورة

- **السبب:** تم إلغاء الفاتورة أو أن المستخدم يفتقر إلى الأذونات
- **الحل:** تظهر الفواتير الملغاة فقط زر تنزيل PDF. تحقق من تحقق من حالة الفاتورة. تحقق من دور المستخدم وأذوناته.

الوثائق ذات الصلة

- `integrations_mailjet` - تسليم الفواتير عبر البريد الإلكتروني والقوالب
- `administration_configuration` - تكوين قالب الفاتورة
- `payments_transaction` - إنشاء المعاملات التي تظهر في الفواتير
- `payments_process` - معالجة مدفوعات الفواتير
- `payment_system_guide` - مرجع واجهة برمجة التطبيقات للدفع وتكوين البائع

إدارة طرق الدفع

يسمح نظام طرق الدفع في OmniCRM للعملاء والموظفين بإدارة بطاقات الدفع بأمان باستخدام **معالجة الدفع متعددة البائعين** (Stripe، PayPal، إلخ). تمكّن طرق الدفع من الفوترة التلقائية للخدمات، والمدفوعات لمرة واحدة، والرسوم المتكررة دون تخزين بيانات البطاقة الحساسة في OmniCRM.

انظر أيضًا: دليل نظام الدفع، نظرة عامة على الفوترة، معالجة الدفع، الفواتير.

نظرة عامة

يوفر نظام طرق الدفع:

- **تخزين آمن للبطاقات** (Stripe، PayPal)، ولا يتم تخزينها أبدًا في OmniCRM - يتم ترميز البطاقات بواسطة بائعي الدفع
- **دعم متعدد البائعين** - تدعم طرق الدفع من Stripe وPayPal
- **بطاقات متعددة** - يمكن للعملاء تخزين طرق دفع متعددة
- **اختيار افتراضي** - تعيين طريقة الدفع المفضلة للرسوم التلقائية
- **تتبع انتهاء الصلاحية** - مراقبة وتحديث البطاقات التي على وشك الانتهاء
- **خدمة ذاتية** - يمكن للعملاء إدارة بطاقاتهم الخاصة عبر بوابة الخدمة الذاتية
- **إدارة الموظفين** - يمكن لموظفي الدعم إضافة/إزالة البطاقات نيابة عن العملاء

طرق الدفع المدعومة:

- بطاقات الائتمان (Visa، Mastercard، American Express، Discover)
- بطاقات الخصم
- بطاقات مسبقة الدفع (إذا كانت مدعومة من شبكة البطاقة)

لا يتم تخزينها في OmniCRM:

يتم ترميز تفاصيل البطاقة بواسطة بائعي الدفع وتخزينها بأمان. يخزن OmniCRM فقط:

- بائع الدفع (stripe، paypal)
- علامة البطاقة (Visa، Mastercard، إلخ.)
- آخر 4 أرقام
- شهر/سنة انتهاء الصلاحية

- اسم/لقب حامل البطاقة
- رمز طريقة الدفع المحدد من البائع

الوصول إلى طرق الدفع

من صفحة العميل:

1. انتقل إلى **العملاء** ← **[اختر عميلًا]**
2. انقر على علامة تبويب **الفوترة**
3. انتقل إلى قسم **طرق الدفع**

أو مباشرة:

من لوحة معلومات البطاقات التي على وشك الانتهاء:

عرض جميع العملاء الذين لديهم بطاقات على وشك الانتهاء:

هذا يعرض قائمة على مستوى النظام للبطاقات التي ستنتهي خلال 60 يومًا القادمة.

قائمة طرق الدفع

تظهر جدول طرق الدفع جميع البطاقات المخزنة لعميل:

وصف الأعمدة:

- **اللقب** - اسم ودود للبطاقة (مثل "بطاقة شخصية"، "بطاقة عمل")
- **المصدر** - علامة البطاقة وآخر 4 أرقام
- **انتهاء الصلاحية** - شهر/سنة انتهاء الصلاحية (بتنسيق MM/YYYY)
- **تاريخ الإضافة** - تاريخ إضافة البطاقة إلى الحساب
- **افتراضي** - علامة الاختيار تشير إلى طريقة الدفع الافتراضية للرسوم التلقائية

الإجراءات لكل بطاقة:

كل صف لديه قائمة إجراءات (⋮) مع الخيارات:

- **تعيين كافتراضي** - اجعل هذه هي طريقة الدفع الافتراضية
- **حذف** - إزالة البطاقة من الحساب

إضافة طريقة دفع

انقر على **"إضافة طريقة دفع"** لفتح نافذة الدفع الآمنة.

الخطوة 1: إدخال تفاصيل البطاقة

تظهر نموذج الدفع الآمن (مدعوم من Stripe Elements أو PayPal SDK):

الحقول المطلوبة:

- **معلومات البطاقة**، CVC (تم التحقق منه بواسطة Stripe) - رقم البطاقة، تاريخ الانتهاء
- **اسم حامل البطاقة** - الاسم الموجود على البطاقة
- **الدولة/المنطقة** - الدولة التي يتم الفوترة إليها

الحقول الاختيارية:

- **لقب البطاقة** - تسمية ودية لتمييز بين البطاقات

الأمان:

- آمن مستضاف من قبل البائع iframe يتم إدخال تفاصيل البطاقة مباشرة في (Stripe Elements / PayPal SDK)
- أو يخزن أرقام البطاقة الكاملة OmniCRM لا يرى
- بواسطة بائع الدفع PCI DSS يتم التعامل مع الامتثال لمعيار
- تمنع التحقق الفوري الأرقام غير الصحيحة للبطاقات

الخطوة 2: التقديم والترميز

عند النقر على **"إضافة طريقة دفع"**:

1. **التحقق من جانب العميل:**
   - يتحقق بائع الدفع من تنسيق رقم البطاقة
   - يتحقق من أن تاريخ انتهاء الصلاحية في المستقبل
   - يتحقق من تنسيق CVC
2. **الترميز:**
   - يتم إرسال تفاصيل البطاقة مباشرة إلى بائع الدفع (ليس OmniCRM)
   - ينشئ البائع رمزًا آمنًا (مثل `pm_1A2B3C4D` لStripe)
   - يتم إرجاع الرمز إلى OmniCRM
3. **معالجة الخادم:**
   - يخزن OmniCRM الرمز في سجل العميل مع معرف البائع
   - يخزن آخر 4 أرقام، العلامة، تاريخ الانتهاء، واسم البائع للعرض
   - لا تلمس أبدًا رقم البطاقة الكامل خوادم OmniCRM

الخطوة 3: التأكيد

تظهر رسالة النجاح:

> تم إضافة بطاقة Visa إلى حسابك الخاصة بك التي تنتهي بـ 1234.

تظهر البطاقة الجديدة في جدول طرق الدفع.

اختيار افتراضي تلقائي:

- إذا كانت هذه هي البطاقة الأولى للعميل، يتم تعيينها تلقائيًا كافتراضية
- إذا كان لدى العميل بطاقات بالفعل، تتم إضافة البطاقة الجديدة كغير افتراضية
- يمكن للعميل تغيير الافتراضي بعد الإضافة

تعيين طريقة الدفع الافتراضية

تستخدم طريقة الدفع الافتراضية لـ:

- رسوم الخدمة المتكررة التلقائية
- مدفوعات الفواتير
- الشحنات وإعادة الشحن
- المعاملات لمرة واحدة (ما لم يُحدد خلاف ذلك)

لتغيير الافتراضي:

1. حدد البطاقة التي تريد تعيينها كافتراضية في جدول طرق الدفع
2. انقر على **قائمة الإجراءات (⋮)** بجوار البطاقة
3. اختر **"تعيين كافتراضي"**
4. تظهر رسالة التأكيد

   > أصبحت Visa الخاصة بك التي تنتهي بـ 5678 الآن طريقة الدفع الافتراضية.

تتحرك علامة الاختيار إلى البطاقة الجديدة المحددة.

مؤشر بصري:

تظهر البطاقات الافتراضية:

في عمود الافتراضي، عادةً مع شارة علامة اختيار خضراء.

حذف طريقة دفع

قم بإزالة البطاقات التي انتهت صلاحيتها، أو فقدت، أو لم تعد مطلوبة.

الخطوة 1: بدء الحذف

1. ابحث عن البطاقة التي تريد حذفها في جدول طرق الدفع
2. انقر على أيقونة **سلة المهملات** في ذلك الصف

إذا كان لدى العميل طريقة دفع واحدة نشطة فقط، تكون أيقونة سلة المهملات **معطلة**. يظهر عند التمرير فوقها تلميح:

> لا يمكن حذف طريقة الدفع الوحيدة لديك. أضف طريقة دفع أخرى أولاً.

أضف طريقة دفع ثانية قبل أن تتمكن من إزالة الحالية.

الخطوة 2: تأكيد الحذف

عندما يكون الحذف مسموحًا، تظهر نافذة تأكيد:

> هل أنت متأكد أنك تريد حذف هذه الطريقة للدفع؟
>
> [حذف طريقة الدفع] [إلغاء]

إذا كانت الطريقة التي يتم حذفها هي الافتراضية الحالية وتوجد طرق نشطة أخرى، تلاحظ النافذة أن طريقة نشطة أخرى سيتم ترقيتها إلى الافتراضية بعد الحذف.

انقر على **"حذف طريقة الدفع"** للتأكيد.

الخطوة 3: اكتمال الحذف

رسالة النجاح:

تمت إزالة البطاقة من الجدول وتم حذفها من بائع الدفع.

قيود مهمة:

- **لا يمكن حذف الطريقة الوحيدة النشطة** - يتم تعطيل عنصر التحكم في الحذف (مع تلميح). ترفض واجهة برمجة التطبيقات أيضًا الطلب. أضف طريقة دفع أخرى أولاً.
- **حذف الافتراضي عندما توجد طرق أخرى** - يتم تلقائيًا ترقية طريقة دفع نشطة أخرى قابلة للاستخدام إلى الافتراضية.
- **لا يوجد تراجع** - الحذف دائم؛ يجب على العميل إعادة إضافة الطريقة إذا لزم الأمر

إدارة البطاقات التي على وشك الانتهاء

يتتبع OmniCRM تواريخ انتهاء صلاحية البطاقات ويقدم أدوات لتحديث البطاقات التي على وشك الانتهاء بشكل استباقي.

لوحة معلومات البطاقات التي على وشك الانتهاء

انتقل إلى **الفوترة → البطاقات التي على وشك الانتهاء** لرؤية قائمة على مستوى النظام:

> العميل تاريخ انتهاء البطاقة الأيام حتى الإجراء جون سميث Visa **1234 02/2025 12** يومًا تحديث جين دو MC5678 03/2025 45 يومًا تحديث شركة أكمي Amex 9012 01/2025 تحديث منتهية الصلاحية

الفلاتر:

- **نطاق انتهاء الصلاحية** - الأيام الـ 30/60/90 القادمة أو المنتهية بالفعل
- **نوع العميل** - فردي مقابل تجاري
- **نوع الخدمة** - تصفية حسب الخدمة التي تتطلب طريقة الدفع

الإجراءات:

- **تحديث** - يفتح صفحة طرق الدفع الخاصة بالعميل لإضافة بطاقة جديدة
- **إخطار** - إرسال تذكير عبر البريد الإلكتروني للعميل (Mailjet إذا تم تكوين)

إشعارات انتهاء الصلاحية

إذا تم تكوين Mailjet، يتم إرسال رسائل بريد إلكتروني تلقائية:

- **قبل 60 يومًا من انتهاء الصلاحية** - التذكير الأول
- **قبل 30 يومًا من انتهاء الصلاحية** - التذكير الثاني
- **قبل 7 أيام من انتهاء الصلاحية** - التحذير النهائي
- **عند انتهاء الصلاحية** - إشعار بانتهاء صلاحية البطاقة

يمكن للعملاء النقر على رابط في البريد الإلكتروني لتحديث طريقة الدفع الخاصة بهم عبر بوابة الخدمة الذاتية.

متغيرات قالب البريد الإلكتروني:

تستقبل قوالب Mailjet:

- اسم العميل
- علامة البطاقة وآخر 4 أرقام
- تاريخ انتهاء الصلاحية
- رابط إلى صفحة طرق الدفع في الخدمة الذاتية

انظر `integrations_mailjet` لتكوين قالب البريد الإلكتروني.

تحديث بطاقة على وشك الانتهاء

سير العمل الموصى به:

1. يتلقى العميل إشعار انتهاء الصلاحية عبر البريد الإلكتروني
2. يقوم العميل بتسجيل الدخول إلى بوابة الخدمة الذاتية

3. ينتقل إلى **الفوترة ← طرق الدفع**
4. ينقر على **"إضافة طريقة دفع"**
5. يدخل تفاصيل البطاقة الجديدة (نفس البطاقة مع تاريخ انتهاء صلاحية محدث، أو بطاقة بديلة)
6. يحدد البطاقة الجديدة كافتراضية
7. يحذف البطاقة القديمة/المنتهية

سير عمل الموظف:

إذا اتصل العميل بالدعم:

1. يفتح الموظف حساب العميل
2. ينتقل إلى **الفوترة ← طرق الدفع**
3. يضيف بطاقة جديدة نيابة عن العميل (يقدم العميل التفاصيل عبر الهاتف)
4. يحدد البطاقة الجديدة كافتراضية
5. يحذف البطاقة المنتهية
6. يؤكد مع العميل

> **⚠ WARNING**
> لا تطلب أبدًا من العملاء إرسال تفاصيل البطاقة عبر البريد الإلكتروني أو الرسائل النصية. استخدم دائمًا:
>
> - بوابة الخدمة الذاتية الآمنة للخدمة الذاتية
> - الهاتف مع الموظف الذي يدخل التفاصيل مباشرة في النظام
> - شخصيًا في موقع البيع بالتجزئة

ماذا يحدث عندما تنتهي صلاحية البطاقات

عندما تصل بطاقة الدفع إلى تاريخ انتهاء صلاحيتها ولا يتم تحديثها:

التأثيرات الفورية:

1. **فشل المدفوعات التلقائية**
   - يرفض بائع الدفع المعاملات مع البطاقات المنتهية
   - تفشل تجديدات الخدمة الشهرية في المعالجة
   - تفشل الشحنات التلقائية

- تفشل المدفوعات التلقائية للفواتير

2. **إشعارات العملاء**
   - يحاول النظام فرض رسوم على البطاقة
   - يتم إرسال إشعار بفشل الدفع
   - يتم إرسال بريد إلكتروني "تحديث طريقة الدفع" مع رابط إلى بوابة الخدمة الذاتية
3. **تغييرات حالة الخدمة**
   - **الخدمات المدفوعة لاحقًا** - قد تستمر مؤقتًا مع رصيد مستحق
   - **الخدمات المدفوعة مسبقًا** - تعليق الخدمة عند نفاد الرصيد
   - **الخدمات المتجددة تلقائيًا** - يفشل التجديد، قد تنتهي الخدمة

الإجراءات اللاحقة:

اليوم 1-3 (فترة السماح):

- تستمر الخدمة بشكل طبيعي
- يتلقى العميل أول إشعار بفشل الدفع
- يحاول النظام إعادة المحاولة (حسب التكوين)

اليوم 4-7:

- محاولة دفع ثانية (إذا تم تكوينها)
- يتم إرسال بريد إلكتروني تحذيري
- قد يتصل خدمة العملاء بالعميل

اليوم 8-14:

- قد يتم تعليق الخدمة بسبب عدم الدفع
- حالة التعليق تمنع الاستخدام ولكن تحافظ على الحساب
- يمكن للعميل استعادة الخدمة عن طريق تحديث طريقة الدفع ودفع الرصيد المستحق

اليوم 15+:

- قد يتم إنهاء الخدمة بسبب عدم الدفع
- يتم وضع المخزون (المعدات، بطاقات SIM) للعودة
- تم إرسال إشعار نهائي
- تم إحالة الحساب إلى التحصيل (إذا كان ذلك مناسبًا)

منع انقطاع الخدمة:

لتجنب انقطاع الخدمة:

- تحديث البطاقات **قبل 30 يومًا من انتهاء الصلاحية**
- إضافة طرق دفع متعددة كاحتياطي
- تفعيل تنبيهات فشل الدفع
- مراقبة لوحة معلومات البطاقات المنتهية أسبوعيًا

استعادة الخدمة بعد انتهاء الصلاحية:

إذا تم تعليق الخدمة بسبب بطاقة منتهية الصلاحية:

1. إضافة طريقة دفع جديدة وصالحة
2. تعيينها كطريقة الدفع الافتراضية
3. دفع الرصيد المستحق (إن وجد)
4. الاتصال بالدعم لإعادة تفعيل الخدمة
5. استعادة الخدمة خلال دقائق إلى ساعات

أمان طريقة الدفع

التوكن

تستخدم OmniCRM توكن البائع لضمان الأمان:

1. **يدخل العميل البطاقة** → ترسل مباشرة إلى خوادم بائع الدفع
2. **يحقق البائع ويقوم بالتوكن** → ينشئ توكن فريد
3. **يتم تخزين التوكن في OmniCRM** → لا يتم تخزين رقم البطاقة بالكامل
4. **معالجة الدفع** → يتم إرسال التوكن إلى البائع، ويقوم البائع بخصم البطاقة

ما تخزنه OmniCRM:

```
{
  "vendor": "stripe",
  "vendor_payment_method_id": "pm_1A2B3C4D5E6F",
  "payment_type": "card",
  "brand": "visa",
  "last4": "1234",
  "exp_month": 12,
  "exp_year": 2026,
  "name": "John Smith",
  "nickname": "Personal Card",
  "is_default": true
}
```

ما لا تخزنه OmniCRM:

- رقم البطاقة بالكامل
- رمز CVV/CVC
- بيانات الشريط المغناطيسي
- أرقام PIN

الامتثال لمعايير PCI

من خلال استخدام نماذج الدفع المستضافة من قبل البائع:

- OmniCRM **مخفض** - بيانات البطاقة لا تلمس خوادم **نطاق PCI**
- آمن للبائع iframe **حقول مستضافة من قبل البائع** - إدخال البطاقة يحدث في
- **لا يتم تخزين البطاقة** - يتم استخدام التوكن بدلاً من بيانات البطاقة الخام
- HTTPS/TLS **نقل آمن** - جميع الاتصالات عبر

راجع دليل نظام الدفع للحصول على تفاصيل أمان بائع الدفع.

سير العمل الشائعة

سير العمل 1: إضافة العميل لأول طريقة دفع

السيناريو: عميل جديد يسجل للحصول على الخدمة

1. يقوم العميل بإنشاء حساب
2. يختار خطة الخدمة
3. يُطلب منه إضافة طريقة دفع أثناء الخروج
4. يدخل تفاصيل البطاقة في نافذة Stripe
5. يتم توكن البطاقة وحفظها
6. يتم تعيينها تلقائيًا كطريقة الدفع الافتراضية
7. يتم توفير الخدمة
8. يتم معالجة أول عملية خصم

سير العمل 2: تحديث العميل لبطاقة منتهية

السيناريو: بطاقة الائتمان على وشك الانتهاء

1. يتلقى العميل إشعارًا عبر البريد الإلكتروني (قبل 60 يومًا من انتهاء الصلاحية)
2. يقوم بتسجيل الدخول إلى بوابة الخدمة الذاتية
3. يتنقل إلى **الفوترة ← طرق الدفع**
4. يستعرض البطاقة الحالية التي ستنتهي في 12/2025
5. ينقر على **"إضافة طريقة دفع"**
6. يدخل بطاقة بديلة تنتهي في 12/2028
7. يعين البطاقة الجديدة كطريقة الدفع الافتراضية
8. يحذف البطاقة القديمة
9. يتم إرسال بريد إلكتروني للتأكيد

سير العمل 3: مساعدة الموظف للعميل عبر الهاتف

السيناريو: يتصل العميل: "تم رفض بطاقتي"

1. يتصل العميل بالدعم
2. يتحقق الموظف من الهوية (أسئلة الأمان)
3. يتحقق الموظف من طرق الدفع: البطاقة منتهية في 01/2025
4. الموظف: "بطاقتك منتهية. هل لديك بطاقة جديدة؟"
5. يقدم العميل تفاصيل البطاقة الجديدة عبر الهاتف
6. يتنقل الموظف إلى **العملاء ← [العميل] ← الفوترة**
7. ينقر على **"إضافة طريقة دفع"**

8. يدخل تفاصيل البطاقة كما يقرأها العميل
9. يعين البطاقة الجديدة كطريقة الدفع الافتراضية
10. يحذف البطاقة المنتهية
11. يعيد محاولة الدفع الفاشل
12. يؤكد مع العميل: "تم الدفع بنجاح، وتم استعادة الخدمة"

سير العمل 4: عميل تجاري لديه بطاقات متعددة

السيناريو: ترغب الشركة في بطاقات مختلفة لأغراض مختلفة

1. يضيف العميل التجاري البطاقة الرئيسية (فيزا تنتهي بـ 1111)
2. يعينها كطريقة الدفع الافتراضية لرسوم الخدمة الشهرية
3. يضيف بطاقة احتياطية (ماستركارد تنتهي بـ 2222) للتعبئة
4. يضيف بطاقة شراء (أميكس تنتهي بـ 3333) لشراء المعدات
5. عند إجراء التعبئة، يختار ماستركارد يدويًا عند الخروج
6. لا تزال فيزا الافتراضية تستخدم للفوترة الشهرية التلقائية

سير العمل 5: إدارة البطاقات المنتهية (الإدارة)

السيناريو: إدارة انتهاء الصلاحية بشكل استباقي

1. يتنقل المدير إلى **الفوترة ← البطاقات المنتهية**
2. الفلاتر: "الـ 30 يومًا القادمة"
3. يرى 15 عميلًا لديهم بطاقات منتهية
4. يحدد الكل ← **"إرسال رسائل تذكير عبر البريد الإلكتروني"**
5. ترسل Mailjet رسائل بريد إلكتروني مخصصة لكل عميل
6. يقوم العملاء بتحديث بطاقاتهم عبر الخدمة الذاتية
7. يراجع المدير القائمة بعد أسبوع
8. يتصل بالعملاء المتبقين الذين لم يقوموا بالتحديث
9. يساعد في تحديث البطاقات عبر الهاتف

استكشاف الأخطاء وإصلاحها

تم رفض البطاقة" عند إضافة طريقة الدفع"

- **السبب:** رفضت Stripe (نقص في الأموال، منع الاحتيال، رفض المصدر) البطاقة
- **الإصلاح:**
  - جرب بطاقة مختلفة
  - اتصل بمصدر البطاقة لـ[illegible]فويض [illegible]معاملة
  - تأ[illegible]د من أن البطاقة تدعم المشتريات عبر الإنترنت
  - تحقق من أن عنوان الفوترة يتطابق مع البطاقة المسجلة

خطأ في إضافة طريقة الدفع" (خطأ عام)"

- **السبب:** خطأ في واجهة برمجة تطبيقات بائع الدفع أو مشكلة في الشبكة
- **الإصلاح:**
  - قم بتحديث الصفحة وحاول مرة أخرى
  - تحقق من اتصال الإنترنت
  - تحقق من أن تكوين بائع الدفع صحيح في إعدادات النظام
  - تحقق من وحدة التحكم في المتصفح للحصول على رسالة خطأ محددة
  - جرب متصفحًا مختلفًا (قم بتعطيل حظر الإعلانات)

لا يمكن حذف طريقة الدفع (الزر معطل)

- **السبب:** إنها الطريقة الوحيدة النشطة للعميل
- **الإصلاح:**
  - أضف طريقة دفع جديدة أولاً، ثم احذف القديمة
  - مرر فوق رمز سلة المهملات المعطل للحصول على شرح الأداة

تظهر البطاقة كمنتهية ولكن ليست في قائمة "البطاقات المنتهية"

- **السبب:** انتهت صلاحية البطاقة مؤخرًا، لم يتم تحديث الذاكرة المؤقتة
- **الإصلاح:**
  - قم بتحديث الصفحة
  - تحقق من الفلاتر على لوحة معلومات البطاقات المنتهية
  - قد تنتقل البطاقات المنتهية إلى عرض مختلف

لا تظهر البطاقة الجديدة على الفور

- **السبب:** لم يتم تحديث الصفحة بعد إضافة البطاقة
- **الإصلاح:**
  - يجب أن يتم تحديث جدول طرق الدفع تلقائيًا

    - إذا لم يكن كذلك، قم بتحديث المتصفح يدويًا
    - تحقق مما إذا كان هناك خطأ حدث أثناء عملية الإضافة

لن يتم تحميل نافذة الدفع

- **السبب:** لم يتم تحميل SDK بائع الدفع، مشكلة في مفتاح API، أو إضافة متصفح تمنع التحميل
- **الإصلاح:**
    - تحقق من وحدة التحكم في المتصفح للحصول على أخطاء
    - قم بتعطيل حظر الإعلانات وحماية التتبع
    - تحقق من تكوين بائع الدفع في إعدادات النظام
    - تأكد من تحميل سكربت SDK (تحقق من علامة الشبكة) للبائع
    - جرب وضع التصفح المتخفي/الخاص

لا يتلقى العميل إشعارات انتهاء الصلاحية

- **السبب:** لم يتم تكوين Mailjet أو نموذج البريد الإلكتروني مفقود
- **الإصلاح:**
    - تحقق من بيانات اعتماد Mailjet في *crm_config.yaml*
    - تحقق من وجود نموذج البريد الإلكتروني لبطاقة انتهاء الصلاحية
    - تأكد من أن عنوان البريد الإلكتروني للعميل صحيح
    - تحقق من سجلات Mailjet لفشل التسليم

أفضل الممارسات

للعملاء:

- أضف طريقة الدفع قبل تفعيل الخدمة لتجنب التأخيرات
- احتفظ بما لا يقل عن بطاقتين في السجل كاحتياطي
- تحديث البطاقات المنتهية قبل 30 يومًا على الأقل من انتهاء الصلاحية
- احذف البطاقات القديمة/المنتهية لتجنب الارتباك
- استخدم أسماء مستعارة وصفية ("فيزا شخصية"، "أميكس للعمل")
- تحقق من أن طريقة الدفع الافتراضية صحيحة للفوترة التلقائية

لموظفي الدعم:

- تحقق من هوية العميل قبل الوصول إلى طرق الدفع
- لا تطلب أبدًا من العملاء إرسال تفاصيل البطاقة عبر البريد الإلكتروني/الرسائل ال��صيرة/الدردشة
- قم بمعالجة إضافات البطاقة على الفور أثناء المكالمات (لا تؤجل)
- تأكد من تعيين البطاقة الجديدة كطريقة الدفع الافتراضية بعد الإضافة
- احذف البطاقات القديمة فقط بعد التأكد من أن البطاقة الجديدة تعمل
- اختبر الدفع بعد تحديث البطاقة المنتهية (قم بمعالجة تفويض بقيمة 0.01£)

للمديرين:

- راقب لوحة معلومات البطاقات المنتهية أسبوعيًا
- أرسل رسائل تذكير عبر البريد الإلكتروني قبل 60/30/7 يومًا من انتهاء الصلاحية
- احتفظ بمفاتيح بائع الدفع للاختبار/الإنتاج منفصلة
- تأكد من تكوين نماذج Mailjet لإشعارات انتهاء الصلاحية
- راجع تقارير الدفع الفاشلة لتحديد البطاقات المنتهية
- درب الموظفين على إجراءات التعامل الآمن مع البطاقات

أفضل الممارسات الأمنية:

- استخدم فقط مفاتيح البائع القابلة للنشر (لا تستخدم المفاتيح السرية في الواجهة الأمامية)
- تأكد من تحميل جميع صفحات الدفع عبر HTTPS
- راجع لوحة معلومات بائع الدفع بانتظام للبحث عن أنشطة مشبوهة
- قم بتمكين قواعد منع الاحتيال للبائع
- تطلب CVC لجميع المعاملات التي لا تتطلب وجود البطاقة
- سجل تغييرات طريقة الدفع في سجل النشاط

الوثائق ذات الصلة

- دليل نظام الدفع - إعداد وتكوين تكامل بائع الدفع
- `payments_process` - معالجة المدفوعات باستخدام طرق الدفع المخزنة
- `payments_invoices` - الدفع التلقائي للفواتير باستخدام البطاقة الافتراضية
- `features_topup_recharge` - نظام التعبئة باستخدام طرق الدفع
- `basics_payment` - مفاهيم الدفع والفوترة العامة

- `customer_care` - بوابة الخدمة الذاتية للعملاء لإدارة بطاقاتهم الخاصة

معالجة المدفوعات

ستتم معالجة غالبية المدفوعات تلقائيًا بواسطة النظام، ولكن هناك أوقات قد تحتاج فيها إلى معالجة الدفع يدويًا.

لدفع فاتورة، حدد الفاتورة غير المدفوعة، وانقر على زر "دفع".

سيفتح هذا نموذج الدفع، حيث يمكنك إدخال طريقة الدفع، والنقر على "إرسال" لمعالجة الدفع.

سيتلقى العميل تلقائيًا إيصالًا للدفع، وسيتم وضع علامة على الفاتورة كمدفوعة.

بالنسبة للتحويلات البنكية، يمكنك إدخال مرجع الدفع وتاريخ إجراء الدفع (إذا كان مختلفًا عن التاريخ الحالي).

نظرة عامة على علامة الفوترة

توفر علامة الفوترة عرضًا موحدًا لجميع المعلومات المالية للعميل، حيث تجمع بين طرق الدفع، والمعاملات، والفواتير في واجهة واحدة لإدارة الفوترة بكفاءة.

المستندات ذات الصلة: طرق الدفع، المعاملات، الفواتير، معالجة الدفع، المحفظة مقابل دفتر معاملات العملاء.

الوصول إلى علامة الفوترة

عرض لكل عميل:

1. انتقل إلى **العملاء** ← **[اختر عميلًا]**
2. انقر على علامة **الفوترة**
3. عرض جميع الأقسام الثلاثة: طرق الدفع، المعاملات، والفواتير

عرض على مستوى النظام:

يمكن الوصول إلى بيانات الفوترة على مستوى النظام بشكل منفصل:

- **الفوترة** ← **المعاملات** - ��ميع المعاملات عبر جميع العملاء
- **الفوترة** ← **الفواتير** - جميع الفواتير عبر جميع العملاء

بوابة الخدمة الذاتية:

يرى العملاء الذين يصلون إلى بوابة الخدمة الذاتية نفس هيكل علامة الفوترة:

- عرض وإدارة طرق الدفع الخاصة بهم
- عرض تاريخ المعاملات
- عرض ودفع الفواتير عبر الإنترنت

هيكل علامة الفوترة

تنظم علامة الفوترة في ثلاثة أقسام رئيسية، تعرض كبطاقات:

القسم 1: طرق الدفع

الغرض: إدارة كيفية دفع العميل مقابل الخدمات

الميزات الرئيسية:

- عرض جميع بطاقات الائتمان المحفوظة
- تعيين طريقة الدفع الافتراضية
- إضافة طرق دفع جديدة (عبر Stripe)
- إزالة البطاقات المنتهية أو غير المستخدمة

المستندات: `basics_payment`

القسم 2: المعاملات

الغرض: تتبع جميع الرسوم والائتمانات للعميل

الميزات الرئيسية:

- عرض إحصائيات المعاملات (الإجمالي، غير المفوتر، هذا الشهر، الشهر الماضي)
- قائمة بجميع المعاملات مع تصفية حسب حالة الإلغاء/الفاتورة
- إضافة معاملات يدوية (رسوم أو ائتمانات)
- إلغاء المعاملات غير الصحيحة
- رؤية المعاملات التي تم فوترتها مقابل غير المفوترة

المستندات: `payments_transaction`

القسم 3: الفواتير

الغرض: تجميع المعاملات في فواتير لدفعها من قبل العميل

الميزات الرئيسية:

- عرض إحصائيات الفواتير (الإجمالي، غير المدفوعة، هذا الشهر، الشهر الماضي)
- قائمة بجميع الفواتير مع تصفية حسب حالة المدفوع/المُلغى
- إنشاء فواتير جديدة من المعاملات غير المفوترة

- تنزيل ملفات PDF للفواتير
- إرسال الفواتير عبر البريد الإلكتروني للعملاء
- دفع الفواتير عبر الإنترنت (أو طرق الدفع اليدوية Stripe)

- إلغاء أو استرداد الفواتير

المستندات: `payments_invoices`

تدفق البيانات بين الأقسام

فهم كيفية تدفق البيانات بين الأقسام الثلاثة أمر بالغ الأهمية لإدارة الفوترة بشكل فعال.

مخطط التدفق

علاقة المعاملة بالفاتورة

1. إنشاء المعاملة:

عندما يتم توفير خدمة أو إضافة رسوم يدوية:

- تم إنشاء المعاملة في قسم **المعاملات**
- حالة المعاملة: **غير مفوتر**

- حقل `invoice_id` للمعاملة هو `null`

مثال:

2. إنشاء الفاتورة:

عندما يقوم الموظف بإنشاء فاتورة:

- تم إنشاء الفاتورة في قسم **الفواتير**
- جميع المعاملات غير المفوترة ضمن نطاق التاريخ مجمعة في الفاتورة
- تم ملء حقل `invoice_id` للمعاملة
- تتغير حالة المعاملة إلى: **مفوتر**

مثال:

3. تحديث إحصائيات المعاملات:

- ينخفض إجمالي **المعاملات غير المفوترة**
- تزيد إحصائية **إجمالي الفواتير**
- يزيد إجمالي **الفواتير غير المدفوعة**

علاقة الفاتورة بالدفع

1. دفع الفاتورة:

عندما يدفع العميل الفاتورة:

- تتم معالجة الدفع باستخدام **طريقة الدفع** المحفوظة (بطاقة Stripe)
- أو يتم اختيار طريقة الدفع اليدوية (نقدًا، نقطة بيع، تحويل بنكي)
- تتغير حالة الفاتورة إلى: **مدفوعة**

2. إنشاء معاملة الدفع:

للدفعات اليدوية (غير Stripe):

- يتم إنشاء معاملة سلبية تلقائيًا
- عنوان المعاملة: "دفع للفاتورة #1234"
- مبلغ المعاملة: -45.00$ (سلبية، تعطي ائتمانًا للعميل)

- حقل `invoice_id` للمعاملة: يرتبط بالفاتورة المدفوعة

مثال:

3. تحديث الإحصائيات:

- ينخفض إجمالي **الفواتير غير المدفوعة**
- تبقى **إجمالي الفواتير لهذا الشهر** دون تغيير (الفاتورة كانت موجودة بالفعل)

علاقة طريقة الدفع بالفاتورة

تدفق الدفع عبر Stripe:

1. يضيف العميل بطاقة ائتمان في **طرق الدفع**
2. يتم تحويل البطاقة عبر Stripe، وتخزينها بشكل آمن
3. عند دفع الفاتورة، يختار العميل البطاقة المحفوظة
4. يقوم Stripe بخصم البطاقة
5. يتم وضع علامة على الفاتورة كمدفوعة
6. يتم ملء حقل `payment_reference` بمعرف نية الدفع من Stripe

تدفق الدفع اليدوي:

1. يدفع العميل نقدًا/نقطة بيع/تحويل بنكي (لا حاجة لطريقة دفع)
2. يختار الموظف طريقة الدفع في نافذة دفع الفاتورة
3. يدخل الموظف رقم المرجع (اختياري)
4. يتم إنشاء معاملة سلبية لمبلغ الدفع
5. يتم وضع علامة على الفاتورة كمدفوعة

سير العمل الكامل للفوترة

توضح هذه العمليات كيفية عمل الأقسام الثلاثة معًا لإنجاز المهام الشائعة.

سير العمل 1: إعداد عميل جديد والفاتورة الأولى

الهدف: إعداد الفوترة لعميل جديد وجمع أول دفعة

1. **إضافة طريقة الدفع:**

- انتقل إلى ��لعميل ← علامة **الفوترة**
- قسم **طرق الدفع** ← انقر على **"إضافة طريقة دفع"**
- يضيف العميل بطاقة ائتمان عبر Stripe
- يتم حفظ البطاقة كطريقة الدفع الافتراضية

2. **تحقق من المعاملات:**
   - يظهر قسم **المعاملات** المعاملات غير المفوترة:
     - رسوم إعداد الخدمة: 50.00$
     - خدمة الشهر الأول: 45.00$
     - إجمالي غير مفوتر: 95.00$
3. **إنشاء فاتورة:**
   - قسم **الفواتير** ← انقر على **"إنشاء فاتورة أولية"**
   - تعيين نطاق التاريخ ليشمل إعداد الخدمة والشهر الأول
   - انقر على **"إنشاء فاتورة"**
   - تم إنشاء الفاتورة #INV-2025-001234 بمبلغ 95.00$
4. **تحديث المعاملات:**
   - تظهر كلا المعاملتين الآن: فاتورة #INV-2025-001234
   - إجمالي **المعاملات غير المفوترة** الآن 0.00$
5. **إرسال الفاتورة عبر البريد الإلكتروني:**
   - انقر على أيقونة البريد الإلكتروني بجوار الفاتورة
   - يتلقى العميل بريدًا إلكترونيًا بالفاتورة مع رابط الدفع PDF
6. **يدفع العميل عبر الإنترنت:**
   - ينقر العميل على رابط الدفع في البريد الإلكتروني
   - يتم إعادة توجيهه إلى بوابة الخدمة الذاتية
   - انقر على زر **"دفع الفاتورة"**
   - اختر طريقة الدفع الافتراضية
   - انقر على **"دفع الفاتورة"**
   - يقوم Stripe بخصم البطاقة
7. **تحديث الفاتورة:**
   - تتغير حالة الفاتورة إلى **"مدفوعة"**
   - ينخفض إجمالي **الفواتير غير المدفوعة** بمقدار 95.00$

النتيجة: تم إعداد العميل بالكامل مع طريقة الدفع، وتم دفع الفاتورة الأولى.

سير العمل 2: الفوترة الشهرية المتكررة

الهدف: فوتره جميع العملاء مقابل الخدمة الشهرية في نهاية الشهر

1. **رسوم الخدمة التلقائية:**
   - يصل نهاية الشهر (31 يناير)
   - يقوم نظام الفوترة تلقائيًا بإنشاء معاملات لجميع الخدمات المتكررة
   - يظهر قسم **المعاملات** معاملات ج��يدة غير مفوترة
2. **مراجعة المعاملات غير المفوترة:**
   - انتقل إلى قسم **المعاملات**
   - تصفية: **حالة الفاتورة: غير مفوترة**
   - مراجعة قائمة جميع المعاملات الجاهزة للفوترة
   - تحقق من صحة المبالغ والوصف
3. **إنشاء الفواتير:**
   - انتقل إلى **الفوترة ← الفواتير** (على مستوى النظام)
   - لكل عميل (أو استخدم عملية الدفعة):
     - انقر على **"إنشاء فاتورة أولية"**
     - اختر العميل
     - تاريخ البدء: 01-01-2025
     - تاريخ الانتهاء: 31-01-2025
     - تاريخ الاستحقاق: 15-02-2025
     - انقر على **"إنشاء فاتورة"**
4. **تحديث المعاملات:**
   - جميع المعاملات مرتبطة الآن بالفواتير
   - تم إعادة تعيين إجمالي **المعاملات غير المفوترة** إلى 0.00$
5. **إرسال جميع الفواتير عبر البريد الإلكتروني:**
   - لكل ��اتورة، انقر على أيقونة البريد الإلكتروني
   - يتلقى جميع العملاء الفواتير الشهرية
6. **يدفع العملاء:**
   - يدفع العملاء الذين لديهم طرق دفع محفوظة عبر الإنترنت عبر الخدمة الذاتية
   - يقوم الموظف بمعالجة المدفوعات النقدية/نقطة البيع للعملاء الذين يدفعون شخصيًا
   - ينخفض إجمالي **الفواتير غير المدفوعة** مع استلام المدفوعات

النتيجة: تم فوتره جميع العملاء لشهر يناير، وتم إرسال الفواتير، ومعالجة المدفوعات.

سير العمل 3: معالجة ائتمان مشكلة الخدمة

الهدف: منح ائتمان للعميل بسبب انقطاع الخدمة، وتطبيقه على فاتورة غير مدفوعة

1. **تقرير العميل عن المشكلة:**
    - كانت الخدمة متوقفة لمدة يومين
    - يستحق العميل ائتمانًا قدره $10
2. **إضافة معاملة ائتمان:**
    - انتقل إلى العميل ← علامة **الفوترة** ← قسم **المعاملات**
    - انقر على "**إضافة معاملة +**"
    - نوع المعاملة: **ائتمان**
    - نوع الائتمان: **دفع نقدي** (أو النوع المناسب)
    - العنوان: "ائتمان انقطاع الخدمة"
    - الوصف: "تعويض عن انقطاع لمدة يومين 8-9 يناير"
    - التكلفة التجزئة: 10.00
    - انقر على "**إضافة معاملة**"
3. **تم إنشاء المعاملة:**
    - تظهر المعاملة في القائمة بمبلغ: **10.00$-**
    - حالة المعاملة: **غير مفوترة**
    - الآن يتضمن إجمالي **المعاملات غير المفوترة** -10.00$
4. **تطبيق على الفاتورة:**
    - إذا كان لدى العميل فاتورة غير مدفوعة بالفعل:
        - تبقى الفاتورة غير مدفوعة بالمبلغ الأصلي
        - سيتم تطبيق الائتمان على إنشاء الفاتورة التالية
    - إذا تم إنشاء فاتورة جديدة:
        - قسم **الفواتير** ← انقر على "**إنشاء فاتورة أولية**"
        - تضمين نطاق التاريخ مع معاملة الائتمان
        - تم إنشاء فاتورة مع تطبيق الائتمان:
5. **يدفع العميل:**
    - يدفع العميل المبلغ المخفض: 35.00$
    - يتم وضع علامة على الفاتورة كمدفوعة

النتيجة: تم منح العميل ائتمانًا عن الانقطاع، وتم تطبيق الائتمان على الفاتورة التالية، وتم جمع دفعة أقل.

سير العمل 4: انتهاء صلاحية طريقة الدفع - تحديث وإعادة المحاولة

الهدف: انتهت صلاحية بطاقة العميل، مما تسبب في فشل الدفع - تحديث البطاقة وإعادة المحاولة

1. **إشعار فشل الدفع:**
    - يحاول العميل دفع الفاتورة
    - تعيد Stripe رسالة خطأ: "انتهت صلاحية البطاقة"
    - يفشل الدفع، وتبقى الفاتورة غير مدفوعة
2. **تحديث طريقة الدفع:**
    - ينتقل العميل إلى علامة **الفوترة**
    - قسم **طرق الدفع** ← انقر على **"إضافة طريقة دفع"**
    - أدخل تفاصيل البطاقة الجديدة (تاريخ انتهاء الصلاحية المحدث)
    - تم حفظ البطاقة الجديدة
3. **تعيين كافتراضي:**
    - ينقر العميل على **"تعيين كافتراضي"** على البطاقة الجديدة
    - تتم إزالة البطاقة ��لقديمة تلقائيًا (إذا رغبت في ذلك)
4. **إعادة محاولة الدفع:**
    - انتقل إلى قسم **الفواتير**
    - حدد الفاتورة غير المدفوعة
    - انقر على أيقونة **"دفع"**
    - تفتح نافذة الدفع مع البطاقة الافتراضية الجديدة محددة مسبقًا
    - انقر على **"دفع الفاتورة"**
    - يقوم Stripe بخصم البطاقة الجديدة بنجاح
5. **تحديث الفاتورة:**
    - تتغير حالة الفاتورة إلى **"مدفوعة"**
    - يتم ملء حقل `payment_reference` من Stripe بمعرف نية الدفع الجديد

النتيجة: تم تحديث طريقة الدفع للعميل، وتم دفع الفاتورة بنجاح باستخدام البطاقة الجديدة.

سير العمل 5: إلغاء فاتورة غير صحيحة وإعادة الفوترة

الهدف: تم إنشاء فاتورة بواسطة الموظف مع معاملات خاطئة - إلغاء وإعادة إنشاء بشكل صحيح

1. **اكتشاف الخطأ:**

- مع نطاق تاريخ خاطئ #INV-2025-001234 تم إنشاء الفاتورة
- شملت معاملات من شهر خاطئ
- الفاتورة غير مدفوعة

2. **إلغاء الفاتورة:**
   - انتقل إلى علامة **الفوترة** → قسم **الفواتير**
   - حدد الفاتورة غير الصحيحة
   - انقر على أيقونة الحذف ()
   - تأكيد الإلغاء
   - تم إلغاء الفاتورة
3. **إطلاق المعاملات:**
   - انتقل إلى قسم **المعاملات**
   - جميع المعاملات من الفاتورة الملغاة تظهر الآن: **غير مفوترة**
   - يزيد إجمالي **المعاملات غير المفوترة**
   - المعاملات متاحة لفاتورة جديدة
4. **إنشاء فاتورة صحيحة:**
   - قسم **الفواتير** → انقر على **"إنشاء فاتورة أولية"**
   - تعيين نطاق التاريخ الصحيح
   - تطبيق الفلتر إذا لزم الأمر (على سبيل المثال، "محمول" لفاتورة مخصصة للمحمول)
   - انقر على **"إنشاء فاتورة"**
   - تم إنشاء فاتورة جديدة مع المعاملات الصحيحة
5. **التحقق والإرسال عبر البريد الإلكتروني:**
   - مراجعة تفاصيل الفاتورة الجديدة
   - التحقق من صحة المعاملات المضمنة
   - انقر على أيقونة البريد الإلكتروني لإرسالها إلى العميل

النتيجة: تم إلغاء الفاتورة غير الصحيحة، وتم إعادة فوتره المعاملات بشكل صحيح، وتلقى العميل الفاتورة المصححة.

سير العمل 6: معالجة الدفع النقدي لعدة فواتير

الهدف: يدفع العميل عدة فواتير غير مدفوعة بدفعة نقدية واحدة

1. **وصول العميل مع النقد:**
   - يحضر العميل 300$ نقدًا لدفع الفواتير المستحقة

- انتقل إلى العميل → علامة **الفوترة**

2. **مراجعة الفواتير غير المدفوعة:**
   - قسم **الفواتير** → تصفية: **مدفوعة: لم يتم دفعها بعد**
   - عرض الفواتير غير المدفوعة:
3. **دفع الفاتورة الأولى:**
   - انقر على أيقونة الدفع على الفاتورة #1234
   - تفتح نافذة الدفع
   - اختر طريقة الدفع **"نقدًا"**
   - أدخل المرجع: "نقد مدفوع 10-02-2025 - إيصال #001"
   - انقر على **"دفع الفاتورة"**
   - تم وضع علامة على الفاتورة #1234 كـ **"مدفوعة"**
4. **دفع الفواتير المتبقية:**
   - كرر العملية للفاتورة #1235:
     - المرجع: "نقد مدفوع 10-02-2025 - إيصال #001"
   - كرر للفواتيرة #1236:
     - المرجع: "نقد مدفوع 10-02-2025 - إيصال #001"
5. **التحقق من المعاملات:**
   - انتقل إلى قسم **المعاملات**
   - تم إنشاء ثلاث معاملات دفع جديدة:
   - جميعها مرتبطة بالفواتير المعنية
6. **تحديث الإحصائيات:**
   - قسم **الفواتير** → انخفض إجمالي **الفواتير غير المدفوعة** بمقدار 300.00$
   - تظهر جميع الفواتير الآن حالة **"مدفوعة"**

النتيجة: دفع العميل جميع الفواتير المستحقة نقدًا، وتم تسجيل معاملات الدفع مع مرجع الإيصال.

أفضل الممارسات

لمستخدمي الموظفين

إدارة المعاملات:

- أضف المعاملات اليدوية على الفور (لا تؤجل)

- استخدم عناوين وأوصاف وصفية للوضوح
- اربط المعاملات بالخدمات والمواقع عند الاقتضاء
- ألغِ المعاملات غير الصحيحة قبل فوترتها

إنشاء الفواتير:

- أنشئ الفواتير في فترات متسقة (على سبيل المثال، شهريًا في اليوم الأول من الشهر)
- استخدم نطاقات التاريخ بعناية لتجنب التداخل أو الفجوات
- استخدم الفلاتر لإنشاء فواتير خاصة بالخدمات عند الحاجة
- أرسل الفواتير عبر البريد الإلكتروني فور إنشائها
- راجع ملفات PDF للفواتير قبل إرسالها للعملاء

معالجة الدفع:

- تحقق من صحة طريقة الدفع قبل محاولة الخصم
- أدخل دائمًا أرقام المرجع للمدفوعات اليدوية (نقدًا، نقطة بيع، تحويل بنكي)
- ضع علامة على الفواتير كمدفوعة فور استلام الدفع
- استرد عبر Stripe (قم بإنشاء ائتمان يدوي لطرق الدفع الأخرى) فقط

نظافة البيانات:

- راجع بانتظام المعاملات غير المفوترة
- تحقق من المعاملات الملغاة لفهم أخطاء الفوترة
- راقب الفواتير غير المدفوعة وتابع مع العملاء
- حافظ على تحديث طرق الدفع (قم بإزالة البطاقات المنتهية)

للعملاء (بوابة الخدمة الذاتية)

طرق الدفع:

- احتفظ على الأقل بطريقة دفع واحدة صالحة في الملف
- قم بتحديث طرق الدفع قبل انتهاء صلاحية البطاقات
- عيّن بطاقتك المفضلة كافتراضية

مدفوعات الفواتير:

- ادفع الفواتير قبل تاريخ الاستحقاق لتجنب الرسوم المتأخرة

- راجع تفاصيل الفاتورة والمعاملات قبل الدفع
- قم بتنزيل ملفات PDF للفواتير لسجلاتك
- اتصل بالدعم على الفور إذا ظهرت الفاتورة بشكل غير صحيح

مراجعة المعاملات:

- راجع تاريخ المعاملات بانتظام
- أبلغ عن أي رسوم غير متوقعة على الفور
- افهم أي المعاملات تم فوترتها مقابل غير المفوترة

للمسؤولين

تكوين النظام:

- قم بتكوين قوالب البريد الإلكتروني Mailjet لتسليم الفواتير بشكل احترافي
- خصص قوالب PDF للفواتير لتتناسب مع العلامة التجارية
- قم بإعداد تكامل بائع الدفع (Stripe، PayPal) لمعالجة الدفع بشكل آمن
- قم بتكوين شروط الدفع وتواريخ الاستحقاق

المراقبة والتقارير:

- استخدم أدوات الإحصائيات لمراقبة صحة الفوترة
- تتبع إجمالي **المعاملات غير المفوترة** - يجب أن تنخفض بعد دورة الفوترة
- راقب إجمالي **الفواتير غير المدفوعة** - تابع المدفوعات المتأخرة
- راجع إحصائيات **هذا الشهر** مقابل **الشهر الماضي** للاتجاهات

الأتمتة:

- قم بأتمتة رسوم الخدمة المتكررة عبر تكوين المنتج
- قم بإعداد إنشاء الفواتير التلقائي للفوترة المتكررة (إذا كان متاحًا)
- قم بتكوين تذكيرات البريد الإلكتروني للفواتير المتأخرة

المشكلات الشائعة والحلول

المشكلة: لا يمكن للعميل دفع الفاتورة

الأعراض:

- ينقر العميل على زر الدفع ولكن لا يحدث شيء
- رسالة خطأ: "لم يتم العثور على طرق الدفع"

التشخيص:

1. انتقل إلى العميل ← علامة **الفوترة** ← قسم **طرق الدفع**
2. تحقق مما إذا كان لدى العميل أي طرق دفع محفوظة
3. تحقق مما إذا كانت البطاقات المحفوظة منتهية الصلاحية

الحل:

- يجب على العميل إضافة طريقة دفع صالحة قبل دفع الفواتير
- ارشد العميل إلى صفحة **طرق الدفع** لإضافة بطاقة ائتمان
- تحقق من قبول البطاقة (Visa، Mastercard، Amex، إلخ.)
- أعد محاولة الدفع بعد إضافة البطاقة

المشكلة: تم إنشاء فاتورة بمعاملات خاطئة

الأعراض:

- تتضمن الفاتورة معاملات من فترة خاطئة
- الفاتورة مفقودة معاملات متوقعة
- إجمالي الفاتورة غير صحيح

التشخيص:

1. افتح الفاتورة في قسم **الفواتير**
2. راجع المعاملات المضمنة في الفاتورة
3. تحقق من تواريخ المعاملات مقابل نطاق تاريخ الفاتورة
4. تحقق مما إذا تم تطبيق فلتر أثناء الإنشاء

الحل:

- **إذا كانت الفاتورة غير مدفوعة:** ألغِ الفاتورة، تحقق من المعاملات غير المفوترة�� وأعد إنشائها مع نطاق التاريخ الصحيح
- **إذا كانت الفاتورة مدفوعة:** لا يمكن إلغاؤها - أنشئ معاملة ائتمان للمبلغ غير الصحيح، وأنشئ فاتورة مصححة
- **الوقاية:** راجع دائمًا قسم **المعاملات** قبل إنشاء الفاتورة للتحقق من أن المعاملات الصحيحة ستتضمن

المشكلة: إجمالي المعاملات غير المفوترة لا ينخفض

الأعراض:

- يظهر عنصر واجهة المستخدم **المعاملات غير المفوترة** مبلغًا مرتفعًا
- تظهر قائمة المعاملات العديد من المعاملات غير المفوترة
- تم إنشاء الفواتير الشهرية بالفعل

التشخيص:

1. قم بتصفية المعاملات حسب **حالة الفاتورة: غير مفوترة**
2. راجع قائمة المعاملات غير المفوترة
3. تحقق من تواريخ المعاملات - قد تكون رسومًا حديثة بعد آخر إنشاء فاتورة
4. تحقق مما إذا كانت بعض المعاملات ملغاة (يجب ألا تُحتسب ضمن الإجمالي غير المفوتر)

الحل:

- **السلوك المتوقع:** تبقى المعاملات التي تم إنشاؤها بعد آخر إنشاء فاتورة غير مفوترة حتى دورة الفوترة التالية
- **إذا كانت المعاملات القديمة غير مفوترة:** أنشئ فاتورة لتلك المعاملات مع نطاق التاريخ المناسب
- **إذا كانت المعاملات الملغاة محسوبة:** يجب على النظام استبعاد المعاملات الملغاة تلقائيًا - أبلغ عن خطأ إذا لم يحدث ذلك

المشكلة: ترفض طريقة الدفع باستمرار

الأعراض:

- يفشل الدفع عبر Stripe مع "تم رفض البطاقة"
- يبلغ العميل أن البطاقة يجب أن تعمل
- تفشل محاولات إعادة المحاولة المتعددة

التشخيص:

1. تحقق من لوحة معلومات Stripe لمعرفة سبب الرفض
2. أسباب الرفض الشائعة:
    - عدم كفاية الأموال
    - انتهاء صلاحية البطاقة
    - تم الإبلاغ عن البطاقة كفقدت/مسروقة
    - منع الاحتيال من البنك
    - تم حظر البطاقة الدولية

الحل:

- اطلب من العميل الاتصال ببنكه لتفويض الدفع
- جرب طريقة دفع مختلفة (بطاقة مختلفة)
- للموظفين: قم بمعالجة الدفع كدفع يدوي (نقدًا، نقطة بيع) وضع علامة على الفاتورة كمدفوعة
- تحقق من أن عنوان الفوترة يتطابق مع عنوان فوترة البطاقة

اختصارات لوحة المفاتيح والنصائح

التنقل:

- **B** الوصول إلى علامة الفوترة بسرعة: صفحة العميل → مفتاح (إذا كانت اختصارات لوحة المفاتيح مفعلة)

التصفية:

- **المعاملات:** انقر على فلتر **"غير مفوتر"** قبل إنشاء الفواتير لرؤية ما سيتم تضمينه بالضبط
- **الفواتير:** انقر على فلتر **"لم يتم دفعها بعد"** لرؤية جميع الفواتير المستحقة التي تتطلب متابعة

العمليات الجماعية:

- حدد عدة فواتير (مربع الاختيار) لحذف/إلغاء عدة دفعة واحدة (للموظفين فقط)
- استخدم شريط البحث للعثور بسرعة على معاملة أو فاتورة محددة حسب المعرف

الإجراءات السريعة:

- انقر على عنوان المعاملة/الفاتورة لع��ض التفاصيل الكاملة في نافذة
- قائمة إجراءات النقر الأيمن (⋮) للوصول السريع إلى خيارات التنزيل/الإرسال عبر البريد الإلكتروني/الدفع

الوثائق ذات الصلة

- `basics_payment` - طرق الدفع وتكامل الدفع متعدد البائعين
- `payment_system_guide` - مرجع API للدفع وتكوين البائع
- `payments_transaction` - المعاملات بالتفصيل
- `payments_invoices` - الفواتير بالتفصيل
- `integrations_mailjet` - تسليم الفواتير عبر البريد الإلكتروني
- `csa_activity_log` - عرض تاريخ نشاط الفوترة
- مقاييس نظام فوترة - المراقبة والقياسات OCS/CGRateS والمراقبة

حركات المحفظة والمعاملات العملاء (محاذاة السجل)

نظرة عامة

لنظام إدارة علاقات GL / هي السجل الخاص بالفوترة `customer_transaction` صفوف تاريخ المحفظة التشغيلي. بالنسبة لـ **إعادة شحن محفظة** `wallet_ledger` العملاء. تظل الآن بإنشاء API **البطاقة** و **نقد الإدارة / الائتمان اليدوي للمحفظة**، يقوم مطابق مع **تكلفة التجزئة السلبية** (الأموال المستلمة)، مع تعيين `customer_transaction` إلى `payment_type`:

- `wallet_topup_card`: إعادة الشحن عبر طريقة الدفع المشحونة (`POST /crm/wallet/topup`)
- `wallet_topup_cash`: الائتمان اليدوي (`POST /crm/wallet/credit`، مثل النقد / البنك)
- `wallet_adjustment_debit`: خصم محفظة الإدارة (`POST /crm/wallet/debit`) (رسوم التعديل) **إيجابيًا** ينشر سطرًا

كل سطر يخزن `wallet_ledger_id` الذي يشير إلى صف `wallet_ledger` المقابل للتسوية.

GAAP قاعدة الإشارة مقابل

في هذا السجل الفرعي، تعني **تكلفة التجزئة السلبية** سطرًا **يقلل** من الرسوم الصافية من منظور العميل (مثل إعادة شحن المحفظة / الائتمانات على نمط الدفع)؛ بينما تعني **الإيجابية** سطرًا **مثل الرسوم** (مثل الفاتورة، خصم تعديل المحفظة). هذه هي **قاعدة فوترة CRM**، وليست بديلاً عن العرض الرسمي وفقًا لـ GAAP.

لا تفرض **GAAP الإدخال** هذه الإشارة المحددة للجداول الداخلية. تعتمد الامتثال على **الأمريكية** و **الإيرادات غير مكتسبة، إلخ**)، AR، نقد) **المزدوج في السجل العام**، و **تخطيط الحسابات** `customer_transaction.retail_cost` **إعداد البيانات المالية**، وليس على ما إذا كانت تستخدم الإيجابيات/السلبيات. بالنسبة للمشغلين، يُفضل واجهة المستخدم التي تعرض **المبلغ**

المطلق + التسمية وخيارًا **لرصيد جاري**، بدل** من الاعتماد على الإشارة الخام فقط. **Cr / Dr** مع **مبلغ بارز** و **مرتفع** `retail_cost` تنسق جدول **جميع المعاملات** في الفوترة بالإضافة إلى اللون الأخضر/الأحمر تحت عمود (رسوم ائتمان / خصم؛ توضح التلميحات ذلك) هنا **رسوم على Dr المستحقات / الائتمان**، لذا لا يتم نقل النوع من خلال اللون فقط. تعني **العميل** (المبلغ المستحق)، وليس "تم خصم بطاقتك".

دفع الفاتورة من المحفظة

يتضمن **دفع الفاتورة المحفظة فقط** عندما يكون لدى العميل رصيد إيجابي في المحفظة؛ يستدعي `POST /crm/payments/invoice` مع `routing_mode: wallet_only` (يجب لا تزال **البطاقة المحفوظة** تستخدم التوجيه **الهجين**: يتم تطبيق .(تغطية المبلغ الكامل المحفظة أولاً، ثم البطاقة لأي رصيد متبقي.

الخصم التلقائي من المحفظة عند إنشاء فاتورة؟

لم يتم تنفيذها بشكل افتراضي. **المدفوع مسبقًا مقابل المدفوع لاحقًا** هو قاعدة تجارية: يمكن أن يؤدي الصفر الصامت للمحفظة عند إصدار فاتورة إلى مفاجأة العملاء ويعقد المدفوعات الجزئية، والائتمانات، والنزاعات. نمط أكثر أمانًا: **إصدار الفاتورة**، ثم **الدفع بشكل صريح** (محفظة فقط، بطاقة، هجين، أو أنواع يدوية للموظفين). إذا احتاج المنتج لاحقًا إلى تطبيق تلقائي، أضف إعدادًا واضحًا (مثل لكل عميل أو لكل فاتورة) ووثق تأثير GL.

المدفوع لاحقًا

تقلل ائتمانات المحفظة من `customer.outstanding_balance` (حتى المبلغ الممنوح لـ `customer_payment_type = postpaid`).

الفوترة

تعتبر الصفوف غير المفوترة المتزامنة مع المحفظة **مدفوعات مسبقة من العملاء**. المعاينة **إنشاء فاتورة** في قائمة الفواتير **تستبعد** الصفوف التي يبدأ فيها `payment_type` بـ `wallet_` حتى لا يتم تقليل الفواتير الروتينية عن طريق ائتمانات المدفوعات المسبقة عن طريق الخطأ.

صريحة `transaction_ids` لتطبيق دفعة مسبقة على فاتورة معينة، استخدم سير ع��ل يختار (إذا كانت أدواتك تدعم ذلك)، أو تعامل مع التعديلات خارج هذه المعاينة الافتراضية.

الضرائب: تخزن الفواتير الجديدة من المعاملات المستحقة `customer_invoice.retail_cost` **= sum(line** `retail_cost`**) + sum(line** `tax_amount`**)** (المبلغ المستحق). تستجيب واجهات برمجة التطبيقات للقائمة / التفاصيل أيضًا **لإعادة حساب** هذا المجموع من الصفوف المرتبطة بحيث تظهر الفواتير القديمة بدون الضرائب المجموع الفرعي / الضريبة / PDF وتفرض المبلغ الصحيح. تستخدم `retail_cost` المدمجة في التذييل **عناصر الصف**، وليس رأس الفاتورة الخام وحده.

الأذونات

- `POST /crm/wallet/topup`: المستخدمون المعتمدون الذين يمكنهم استخدام التطبيق (يستخدم العملاء إعادة شحن البطاقة).
- `POST /crm/wallet/credit` و `POST /crm/wallet/debit`: تتطلب `manage_customer_wallet` (ممنوحة لـ `admin` و `customer_service_agent_2` في بذور الأدوار الافتراضية). أعد تشغيل `populate_roles_permissions` (أو ما يعادلها) بعد النشر حتى توجد الإذن وتكون مخصصة.

ائتمان" يدوي في إضافة معاملة"

لم تعد واجهة المستخدم تقدم إدخال معاملة **ائتمان** يدوي، لأنها لم تقم بائتمان المحفظة. استخدم **الفوترة → المحفظة → نقد / يدوي** (موظف) أو مدفوعات الفواتير بدلاً من ذلك.

لوحات (إدارة العمل)

اللوحات نظام إدارة عمل مرن على طراز كانبان لتتبع المهام، والتركيبات، OmniCRM تقدم وطلبات الميزات، وأي عمل آخر يحتاج فريقك إلى تنسيقه. تقع اللوحات بجانب نظام تذاكر الدعم. التذاكر مدفوعة بمشكلات العملاء. اللوحات مدفوعة بالعمل والمشاريع. تمنح اللوحات الفرق طريقة بصرية لتنظيم العناصر في مجموعات، وتحريكها عبر الحالات، وتعيين المالكين، والتعاون.

انظر أيضًا: تذاكر الدعم لتتبع مشكلات العملاء، المستخدمون والأدوار للأذونات، العملاء، جهات الاتصال والمواقع للسجلات التي يمكنك ربط العناصر بها.

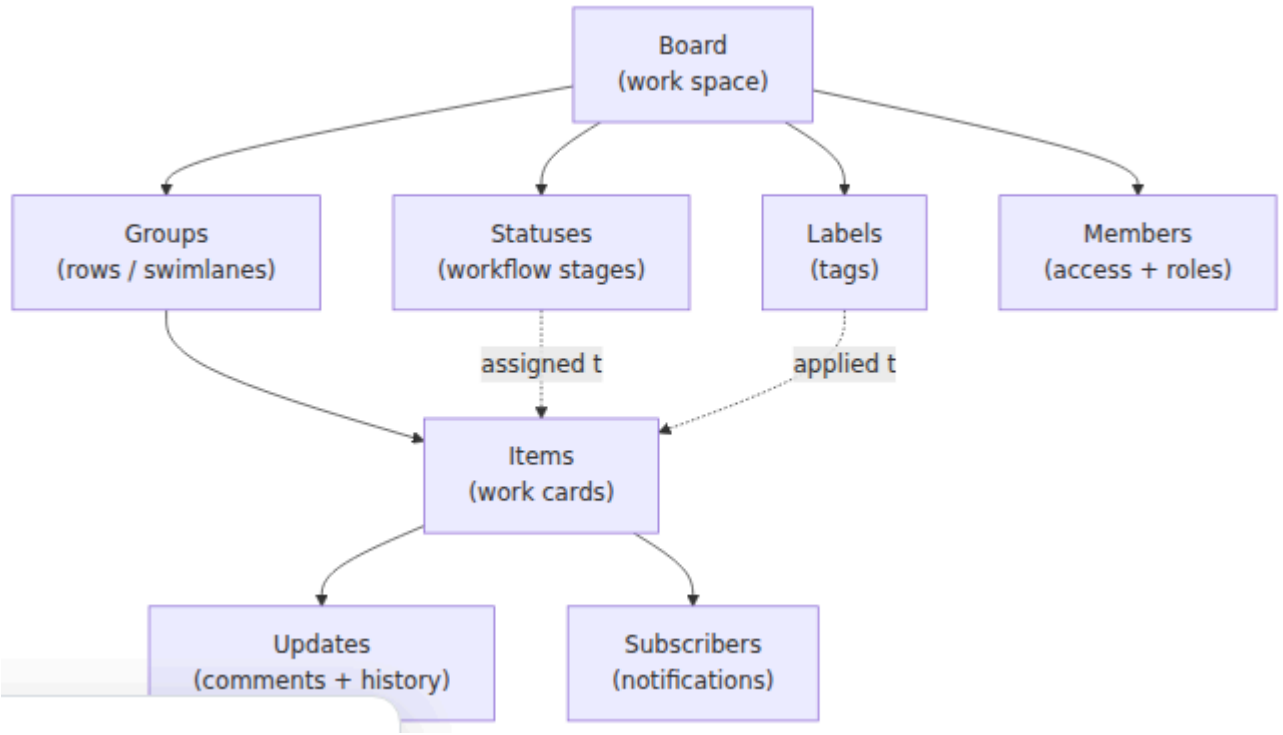
نظرة عامة

تمكن اللوحات:

1. **تتبع العمل ب��ريًا**: تنظيم العمل في مجموعات (صفوف) وحالات (مراحل سير العمل)
2. **القوالب**: البدء بسرعة مع تخطيطات مسبقة البناء لقوائم المهام، والتركيبات، وطلبات الميزات
3. **الملكية**: تعيين العناصر للموظفين، وتسجيل من طلبها، وتتبع المواعيد النهائية
4. **الربط**: ربط العناصر بالعملاء، والخدمات، وتذاكر الدعم
5. **التعاون**: التعليقات، تاريخ التغييرات، @الإشارات، والاشتراكات
6. **التحكم في الوصول**: عضوية لكل لوحة مع أذونات قائمة على الأدوار
7. **المشاركة العامة**: إمكانية كشف لوحة للقراءة فقط عبر رابط آمن للتضمين

المفاهيم الأساسية

تُبنى اللوحة من مجموعة صغيرة من العناصر الأساسية. فهم كيفية تناسقها يجعل بقية هذا الدليل بسيطًا.

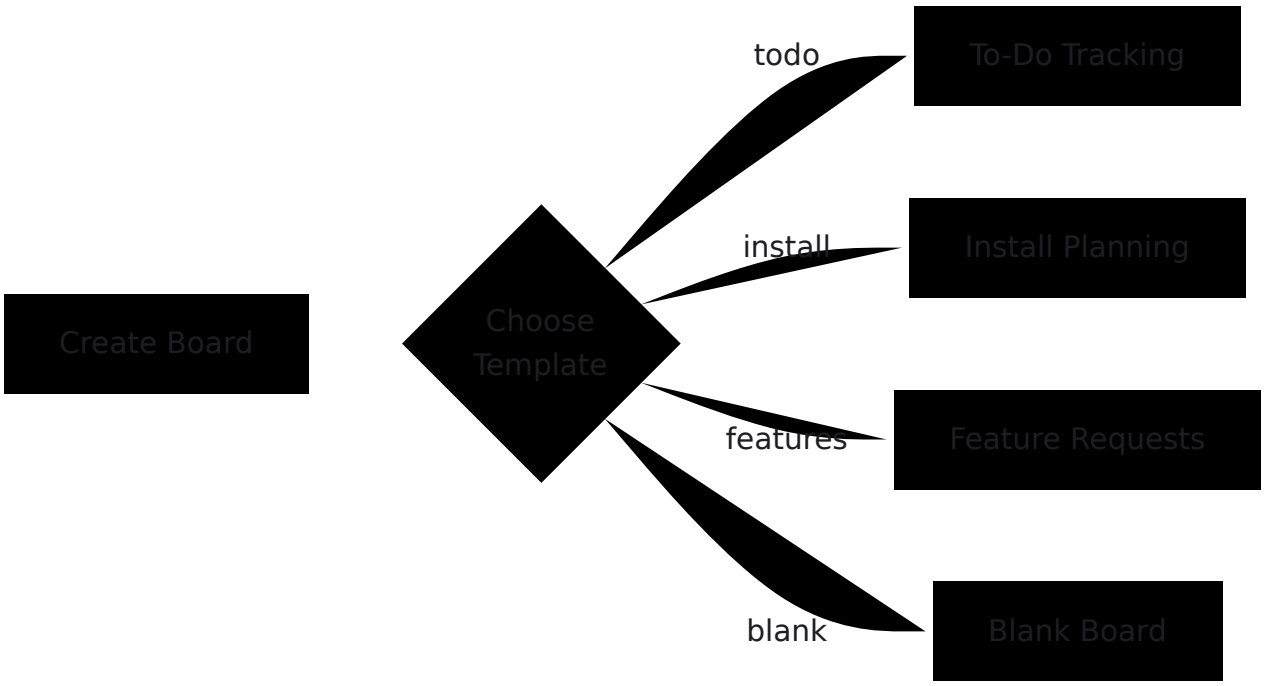


المفهوم	ما هو
اللوحة	مساحة العمل على المستوى الأعلى. تحتوي على مجموعاتها، وحالاتها، وعلاماتها، وأعضائها، وعناصرها الخاصة.
المجموعة	صف أفقي (مسار سباحة) ينظم العناصر في أقسام، على سبيل المثال *هذا الأسبوع*، *الأسبوع المقبل*، *لاحقًا*. يمكن طي المجموعات.
الحالة	مرحلة سير العمل لعنصر، على سبيل المثال *قيد التنفيذ*، *في التقدم*، *منجز*. حالة "منجزة" واحدة هي الافتراضية للعناصر الجديدة. يمكن تمييز أي حالة كحالة.
العنصر	وحدة عمل واحدة (بطاقة). لها عنوان، وصف، أولوية، مُعين، موعد نهائي، علامات، ورقم عنصر فريد.
العلامة	علامة ملونة تُطبق على العناصر للتصنيف (على سبيل المثال *خطأ*، *ميزة*). يمكن أن يحمل العنصر علامات متعددة.
العضو	مستخدم تم منحه الوصول إلى اللوحة، مع دور يتحكم فيما يمكنه القيام به.

كيفية عمل التخطيط: المجموعات هي الصفوف في اللوحة. داخل كل مجموعة، يتم سرد العناصر وتظهر حالتها الحالية، والمُعين، وموعدها النهائي، وأولويتها. تتحرك العناصر *بين المجموعات* لإعادة تنظيمها، و*بين الحالات* لتقدمها عبر سير العمل.

قوالب اللوحة

عند إنشاء لوحة، تختار قالبًا. يقوم القالب بملء حالات ومجموعات اللوحة مسبقًا حتى تتمكن من البدء في العمل على الفور. يمكنك إضافة، إعادة تسمية، تغيير لون، أو إزالة الحالات والمجموعات بعد ذلك.



القالب	الغرض	الحالات (الأولى افتراضيًا)	المجموعات
تتبع المهام (`todo`)	قوائم المهام العامة	قيد التنفيذ · في التقدم · **منجز**	الرئيسية
تخطيط التركيب (`install`)	جدولة التركيبات الميدانية / الشبكية	مجدولة · في التقدم · **مكتملة** · معلقة	هذا الأسبوع · الأسبوع المقبل · لاحقًا
طلبات الميزات (`features`)	خط أنابيب المنتج / الميزات	جديدة · قيد المراجعة · مخطط لها · قيد التطوير · **تم إصدارها**	طلبات الميزات · في التقدم · مكتملة
لوحة فارغة (`blank`)	نقطة انطلاق بسيطة	قيد التنفيذ · **منجز**	العناصر

الحالات المعروضة بالخط **العريض** مُعلمة كحالات "منجزة". العناصر في تلك الحالات تُعتبر مكتملة. إذا لم يتم اختيار قالب، يتم استخدام تخطيط **اللوحة الفارغة**.

ترقيم العناصر

يتم تعيين رقم فريد، قابل للقراءة البشرية لكل ��نصر عند إنشائه، بحيث يمكن الإشارة إليه في المحادثات، والتذاكر، والبريد الإلكتروني.

التنسيق هو:

```
{PREFIX}-{BOARD}-{SEQUENCE}
```

على سبيل المثال: **`BRD-001-00001`**

الجزء	المعنى
`PREFIX`	بادئة عنصر اللوحة. الافتراضي هو `BRD` ويمكن تكوينه لكل لوحة (حتى 10 أحرف، على سبيل المثال `INST` للوحة التركيب).
`BOARD`	معرف اللوحة، مُعبأ بالأصفار إلى ثلاثة أرقام.
`SEQUENCE`	تسلسل العنصر داخل اللوحة، مُعبأ بالأصفار إلى خمسة أرقام.

يمكنك فتح أي عنصر مباشرة برقم العنصر الخاص به من عرض اللوحة أو عبر API.

إنشاء وإدارة اللوحات

إنشاء لوحة

1. انتقل إلى **إدارة العمل** ← **اللوحات**
2. انقر على **إنشاء لوحة**
3. أكمل تفاصيل اللوحة:
   - **اسم اللوحة** (مطلوب)
   - **الوصف** (اختياري)

- **القالب**: *تتبع المهام، تخطيط التركيب، طلبات الميزات، أو لو��ة فارغة*
- **بادئة العنصر**: تُستخدم في أرقام العناصر (الافتراضي `BRD`، حتى 10 أحرف)
- **الرؤية**: *موظفين فقط* أو *عامة* *(تضمين API)*

4. انقر على **إنشاء**

يصبح المستخدم الذي ينشئ اللوحة **مالكًا** لها.

إعدادات اللوحة

يمكن لمالكي اللوحات والمديرين تعديل لوحة لتغيير اسمها، ووصفها، ورمزها، ولونها، والعميل المرتبط بها، وبادئة العنصر، ورؤيتها. يمكن أيضًا **أرشفة** اللوحات لإخفائها من القائمة النشطة دون حذف محتوياتها، و**حذفها** (حذف ناعم يزيل اللوحة من العرض مع الاحتفاظ بالسجلات الأساسية).

تخصيص الحالات، والمجموعات، والعلامات

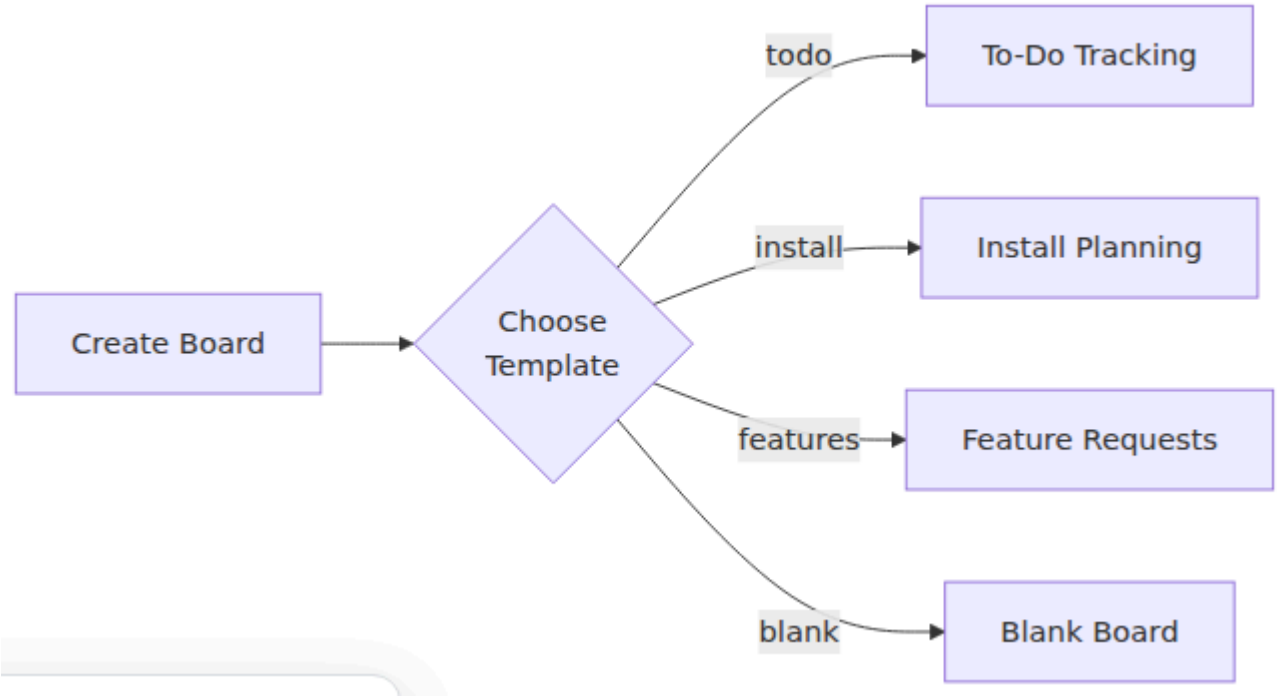
من داخل لوحة يمكنك:

- **الحالات**: إضافة، إعادة تسمية، تغيير لون، إعادة ترتيب، تعيين الحالة الافتراضية للعناصر الجديدة، وتمييز الحالات كـ "منجزة". إذا قمت بحذف حالة، تترك عناصرها بدون حالة. لا يتم حذفها.
- **المجموعات**: إضافة�� إعادة تسمية، تغيير لون، إعادة ترتيب، وطي. إذا قمت بحذف مجموعة، تترك عناصرها غير مجمعة. لا يتم حذفها.
- **العلامات**: إضافة، إعادة تسمية، وتغيير لون. إذا قمت بحذف علامة، يزيل النظام العلامة من جميع العناصر.

العمل مع العناصر

عرض اللوحة

يعرض عرض اللوحة كل مجموعة كقسم مع رأس ملون يعرض اسم المجموعة، وعدد العناصر، وتقدم الإنجاز (عدد العناصر في حالة "منجزة"). داخل كل مجموعة، يتم سرد العناصر مع عنوانها ورقمها، والمُعين، والحالة، وموعدها النهائي، وأولويتها. تتيح لك صف الإضافة السريعة إنشاء عناصر داخل السطر.



أسماء الحالات أعلاه توضيحية. كل لوحة تعرف حالات خاصة بها.

إنشاء عنصر

يمكن إنشاء العناصر من صف الإضافة ال��ريعة في أي مجموعة، أو عبر زر **إنشاء عنصر**.
يحتوي العنصر على:

الحقل	الوصف
العنوان	مطلوب. ملخص قصير للعمل.
الوصف	تفاصيل اختيارية.
المجموعة	الصف الذي ينتمي إليه العنصر. الافتراضي هو أول مجموعة إذا لم يتم اختياره.
الحالة	مرحلة سير العمل. الافتراضي هو الحالة الافتراضية للوحة.
الأولوية	(الافتراضي هو *متوسطة*) *منخفضة*، *متوسطة*، *عالية*، أو *عاجلة*.
موعد نهائي	موعد نهائي اختياري.
المعين	الموظف المسؤول.
الطالب	من طلب العمل: مستخدم موظف أو جهة اتصال عميل.
العميل / الخدمة / التذكرة	روابط اختيارية للسجلات ذات الصلة.
العلامات	علامة أو أكثر من علامات الفئة.

نقل العناصر

يتم نقل العناصر لإعادة تنظيمها أو لتقدمها:

- **تغيير المجموعة**: نقل العنصر إلى صف / مسار سباحة مختلف
- **تغيير الحالة**: تقدم العنصر عبر سير العمل
- **إعادة الترتيب**: تغيير موضع العنصر داخل مجموعته

يتم تسجيل كل حركة في تاريخ العنصر.

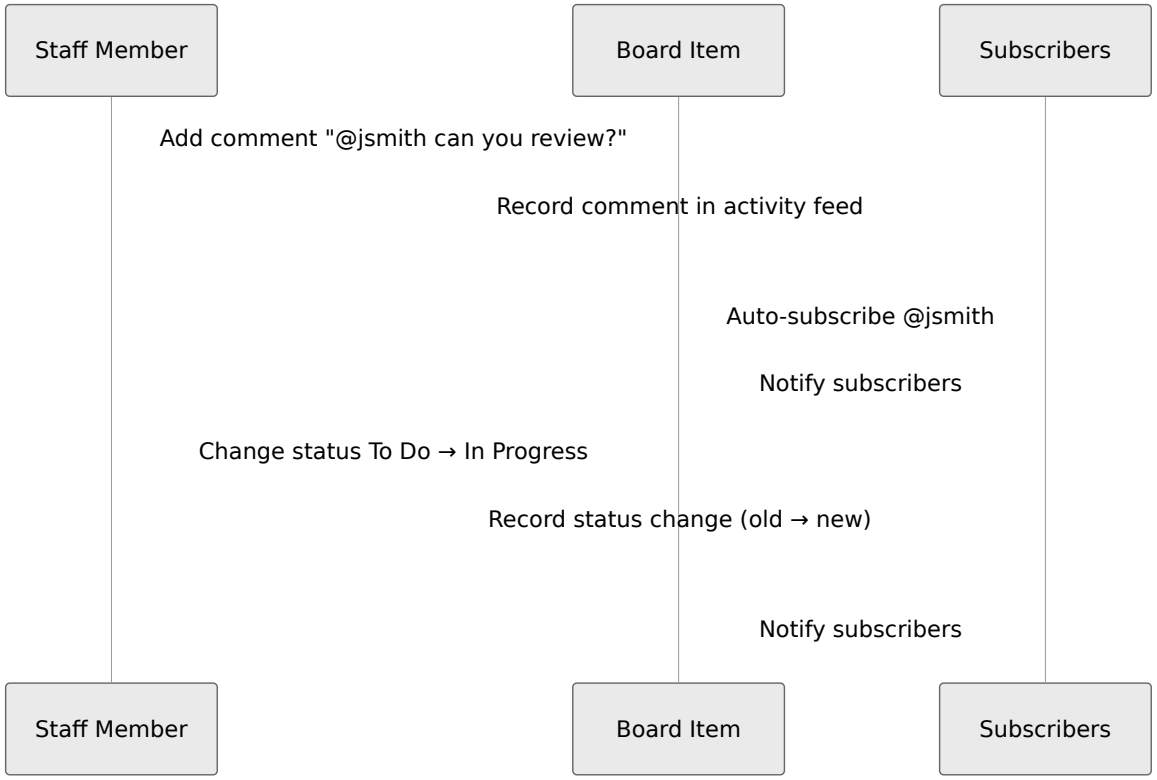
مستويات الأولوية

الأولوية	الاستخدام النموذجي
عاجل	عمل حرج يتطلب اهتمامًا فوريًا
عالي	عمل مهم، يجب معالجته قريبًا
متوسط	عمل قياسي (افتراضي)
منخفض	عمل ثانوي أو مرغوب فيه

التعاون

التعليقات وتاريخ النشاط

يمتلك كل عنصر تغذية نشاط تجمع بين **التعليقات** اليدوية و**تاريخ التغييرات** التلقائي. يسجل النظام التغييرات في الحالة، والمُعين، والعنوان، والأولوية، وموعد الاستحقاق، والمجموعة. يلتقط القيم السابقة والجديدة، بحيث تكون القصة الكاملة للعنصر مرئية دائمًا.



المشتركين والإشعارات

يتلقى المشتركين إشعارات عند تغيير عنصر. يتم إدارة الاشتراكات تلقائيًا ويدويًا:

- **تلقائي**: يشترك النظام في منشئ العنصر والمُعين تلقائيًا. عندما تعيد تعيين عنصر، يشترك النظام في المُعين الجديد.
- **الإشارات**: اذكر موظفًا باستخدام@ `@username` في تعليق للاشتراك بهم تلقائيًا.
- **يدوي**: اشترك في موظفين أو جهات اتصال عملاء في عنصر مباشرة، أو ألغِ اشتراكهم.

يمكن للمشتركين الذين تمت إضافتهم تلقائيًا إلغاء اشتراكهم بأنفسهم.

التحكم في الوصول والأدوار

تكون اللوحات خاصة بأعضائها. يتم منح كل عضو **دور** يتحكم فيما يمكنه القيام به على تلك اللوحة.

الدور	العرض	إنشاء / تعديل العناصر	إدارة الإعدادات (الحالات، العلامات)	إدارة الأعضاء	حذف اللوحة
المالك	✓	✓	✓	✓	✓
المدير	✓	✓	✓	✗	✗
العضو	✓	✓	✗	✗	✗
المشاهد	✓	✗	✗	✗	✗

يمكن لمشرفي النظام الوصول إلى جميع اللوحات وإدارتها بغض النظر عن العضوية. لا يرى المستخدمون غير المشرفين إلا اللوحات التي هم أعضاء فيها.

> **ملاحظة:** تخضع إجراءات اللوحة أيضًا للأذونات على مستوى النظام (انظر المستخدمون والأدوار). يحتاج المستخدم إلى كل من إذن النظام ذي الصلة *ودور اللوحة المناسب*. يجب أن تحتفظ اللوحة دائمًا بمالك واحد على الأقل.

الأذونات المتعلقة باللوحة

الإذن	يمنح القدرة على
`VIEW_BOARD`	قائمة وعرض اللوحات
`CREATE_BOARD`	إنشاء لوحات جديدة
`UPDATE_BOARD`	تعديل تفاصيل اللوحة وإدارة المج��وعات
`DELETE_BOARD`	حذف لوحة
`MANAGE_BOARD_SETTINGS`	إنشاء، تعديل، وحذف الحالات والعلامات
`MANAGE_BOARD_MEMBERS`	إضافة، تحديث، وإزالة أعضاء اللوحة
`GENERATE_PUBLIC_TOKEN`	إنشاء رابط مشاركة عامة
`VIEW_BOARD_ITEM`	عرض العناصر، والتعليقات، والمشتركين
`CREATE_BOARD_ITEM`	إنشاء عناصر
`UPDATE_BOARD_ITEM`	تعديل العناصر، نقل العناصر، وإضافة تعليقات
`DELETE_BOARD_ITEM`	حذف العناصر

الرؤية والمشاركة العامة

تحتوي كل لوحة على إعداد رؤية:

الرؤية	السلوك
موظفين فقط	الافتراضي. يمكن فقط لأعضاء اللوحة (والمشرفين على النظام) الوصول إلى اللوحة.
عامة	بالإضافة إلى الوصول للأعضاء، يمكن قراءة اللوحة عبر رابط عام آمن دون تسجيل الدخول.

توليد **رمز وصول عام** ينتج رابطًا فريدًا، غير قابل للتخمين ويحول اللوحة إلى رؤية *عامة*. الرابط العام هو **للقراءة فقط**. استخدمه لتضمين عرض لوحة، على سبيل المثال على صفحة حالة داخلية أو لوحة عرض. يؤدي إعادة توليد الرمز إلى إبطال الرابط السابق. لإيقاف الوصول العام، قم بتحويل رؤية اللوحة مرة أخرى إلى *موظفين فقط*.



العثور على العناصر

التصفية والبحث

يدعم عرض اللوحة وقوائم العناصر التصفية والبحث للتركيز على ما يهم:

التصفية	الوصف
بحث	يتطابق مع عنوان العنصر، الوصف، ورقم العنصر
الحالة	حالة واحدة أو أكثر
المعين	موظف محدد
الأولوية	مستوى أو أكثر من مستويات الأولوية
المجموعة	مجموعة محددة
العميل	عناصر مرتبطة بعميل محدد
العلامات	علامة أو أكثر
متأخر	عناصر تجاوز موعدها النهائي

الفرز

يمكن فرز العناصر حسب الموضع (ترتيب اللوحة اليدوي)، تاريخ الإنشاء، موعد الاستحقاق، الأولوية، العنوان، أو تاريخ آخر تعديل، بترتيب تصاعدي أو تنازلي. يتم تقسيم اللوحات الطويلة إلى صفحات.

أفضل الممارسات

إعداد اللوحة

1. **اختر القالب الأقرب**: ابدأ من *تخطيط التركيب* أو *طلبات الميزات* بدلاً من *فارغ* عندما تناسب. يمكنك تخصيصها لاحقًا.
2. **استخدم بادئة عنصر ذات معنى**: على سبيل المثال `INST` للتركيبات، بحيث تكون أرقام العناصر ذات وصف ذاتي.
3. **اجعل الحالات بسيطة**: مجموعة صغيرة وواضحة من مراحل سير العمل أسهل في الإدارة من العديد من المراحل المتداخلة.
4. **تمييز حالات "منجزة"**: حتى يظهر تقدم الإنجاز بشكل صحيح في رؤوس المجموعات.

الاستخدام اليومي

1. **عين كل عنصر**: الملكية الواضحة تمنع العمل من التوقف.
2. **حدد مواعيد نهائية**: حت�ً يكون فلتر *متأخر* ذا معنى.
3. **استخدم @الإشارات**: لجذب الأشخاص المناسبين إلى عنصر دون إدارة الاشتراكات يدويًا.
4. **اربط السجلات ذات الصلة**: اربط العناصر بالعميل، الخدمة، أو التذكرة ذات الصلة للسياق.
5. **نقل، لا تعيد إنشاء**: تقدم العناصر عبر الحالات بحيث يروي تاريخ التغيير القصة الكاملة.

استكشاف الأخطاء

المشكلة: لا يمكن لمستخدم رؤية لوحة

- تأكد من إضافة المستخدم كعضو في اللوحة.
- تأكد من أن المستخدم لديه إذن `VIEW_BOARD` لدوره.
- تذكر أن غير المشرفين لا يرون إلا اللوحات التي ينتمون إليها.

المشكلة: لا يمكن لمستخدم تغيير الحالات أو العلامات

- إدارة الحالات والعلامات تتطلب دور **مالك** أو **مدير** للوحة بالإضافة إلى إذن `MANAGE_BOARD_SETTINGS`.

المشكلة: اختفت العناصر بعد حذف حالة أو مجموعة

- العناصر لم تُحذف. حذف حالة يترك عناصرها بدون حالة؛ حذف مجموعة يت�ً ك عناصرها غير مجمعة. عيّن حالة أو مجموعة جديدة لإعادتها إلى العرض.

المشكلة: الرابط العام لا يظهر اللوحة

- تأكد من تعيين رؤية اللوحة إلى *عامة*.
- تأكد من أن الرابط حالي. إعادة توليد الرمز تبطل الرابط السابق.

المشكلة: لا يمكن إزالة عضو

- يجب أن تحتوي اللوحة دائمًا على مالك واحد على الأقل. عيّن مالكًا آخر قبل إزالة الأخير.

سلوكيات CGRateS والإجراءات

تشرح هذه الدليل كيفية عمل إجراءات CGRateS داخل OmniCRM، مع التركيز بشكل خاص على إدارة الرصيد، وسلوكيات التعبئة، وكيف تؤثر أنواع الإجراءات المختلفة على المنتجات الإضافية.

نظرة عامة

الإجراءات هي الآلية لإضافة OmniCRM، الخاص بـ (CGRateS) في نظام الشحن عبر الإنترنت أو تعديل أو إزالة الأرصدة على حسابات العملاء.

يقوم بالتلاعب CGRateS عند توفير منتج إضافي أو تعبئة، فإنك في الواقع تقوم بتنفيذ إجراء بأرصدة الحساب. (يمكنك اتخاذ نهج آخر أيضًا، مثل إضافة الأرصدة إلى الحسابات يدويًا من Playbooks. هذه مجرد نمط شائع نستخدمه للحفاظ على الأمور مرتبة.)

المفاهيم الأساسية

الإجراء - مجموعة من العمليات التي يتم تنفيذها على حساب (إضافة رصيد، خصم رصيد، تسجيل CDR، إلخ.)

الرصيد - كمية من مورد (نقدي، SMS، بيانات، صوت) مع وقت انتهاء ووزن

معرف الرصيد - معرف فريد لنوع الرصيد (على سبيل المثال، "حزمة البيانات"، "دقائق الصوت")

الوزن - أولوية استهلاك الرصيد (يتم استهلاك الوزن الأعلى أولاً)

وقت الانتهاء - عندما ينتهي الرصيد (تاريخ مطلق أو نسبي مثل "+5 أيام")

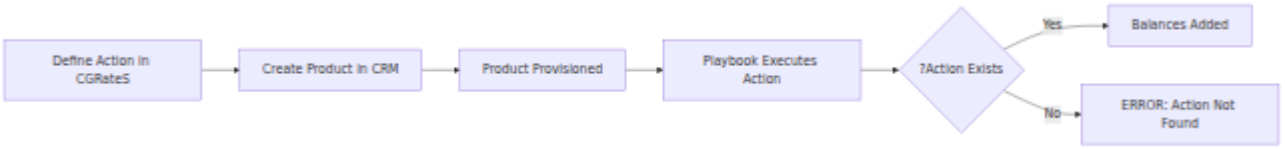
المعطل - علامة رصيد خاصة تمنع جميع الاستخدامات عندما يصل الرصيد إلى الصفر، حتى لو كانت هناك أرصدة أخرى موجودة (انظر معطلات الرصيد)

مهم: يجب تعريف الإجراءات أولاً

قبل أن تتمكن من تنفيذ إجراء في Playbook باستخدام `ExecuteAction`، يجب أن يكون ذلك الإجراء قد تم تعريفه بالفعل في CGRateS. هذا شرط أساسي حاسم غالبًا ما يتم تجاهله.

متى يتم تعريف الإجراءات

عادةً ما ي��م تعريف الإجراءات خلال **تهيئة النظام الأولية** أو **إعداد المنتج**، وليس أثناء التوفير. يتم إنشاؤها عادةً عبر سكربتات بايثون التي تقوم بتكوين كل من CRM و OCS في وقت واحد.



كيف ترتبط الإجراءات بالمنتجات

ترتبط الإجراءات بالمنتجات عبر اتفاقية تسمية:

- **إجراء CGRateS:** `ActionsId = "Action_50gb-data-pack"`
- **منتج CRM:** `product_slug = "50gb-data-pack"`
- **في Playbook:** `cgr_action_name = "Action_" + product_slug`

عند تشغيل Playbook، يقوم بإنشاء اسم الإجراء من product_slug ويستدعي `ExecuteAction`. إذا لم يكن ذلك الإجراء موجودًا في CGRateS، فإن التوفير يفشل.

أين يتم تعريف الإجراءات

يجب تعريف الإجراءات في سكربتات تكوين المنتج الخاصة بك، عادةً:

1. **أثناء الإعداد الأولي** - عند ت��وين النظام لأول مرة
2. **عند إنشاء منتجات جديدة** - تعريف الإجراء قبل إنشاء المنتج
3. **عبر سكربتات التكوين** - سكربتات بايثون التي تقوم بتكوين كل من CRM و OCS

مثال: تعريف إجراء قبل إنشاء المنتج

```
import cgrateshttpapi

OCS_Obj = cgrateshttpapi.CGRateS("ocs.example.com", "2080")
tenant = "your_tenant"

# أولاً CGRateS الخطوة 1: تعريف الإجراء في
Action_50GB_Data_Pack = {
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_50gb-data-pack",
        "Tenant": tenant,
        "Actions": [
            {
                "Identifier": "*topup",
                "BalanceType": "*data",
                "Units": 50 * 1024 * 1024 * 1024,
                "ExpiryTime": "+720h",
                "Weight": 10
            }
        ]
    }]
}

result = OCS_Obj.SendData(Action_50GB_Data_Pack)
assert result['error'] is None or result['error'] == "EXISTS"

# الخطوة 2: الآن قم بإنشاء المنتج في CRM
# (product_slug = "50gb-data-pack" سيرتبط بـ Action_50gb-data-pack)
```

ماذا يحدث إذا لم يكن الإجراء موجودًا:

```
# ف playbook
- name: Execute Action
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body:
      {
        "method": "APIerSv1.ExecuteAction",
        "params": [{
          "ActionsId": "Action_50gb-data-pack"
        }]
      }
  register: response

# النتيجة: إذا لم يتم تعريف الإجراء
# response.json.error = "SERVER_ERROR: Action not found"
# يفشل التوفير، العميل لا يحصل على رصيد
```

للحصول على تفاصيل كاملة حول تعريف الإجراءات وربطها بالمنتجات، انظر تعريف المنتجات.

استقلالية الرصيد

بشكل افتراضي، تقوم المنتجات الإضافية بإنشاء **أرصدة مستقلة** تعمل بشكل منفصل عن بعضها البعض. هذا يعني:

- يمكنك أن يكون لديك عدة إضافات نشطة في نفس الوقت
- يحتفظ كل إضافة برصيدها ووقت انتهاءها الخاص
- يتم استهلاك الأرصدة بناءً على الوزن وقواعد الانتهاء

مثال: إضافات مستقلة متعددة

```
# العميل لديه هذه الأرصدة النشطة:
Balance 1:
  ID: "Data_5GB_5days_uuid_abc123"
  Type: *data
  Value: 5368709120  # 5GB in bytes
  Expiry: 2024-12-29
  Weight: 10

Balance 2:
  ID: "Data_10GB_30days_uuid_def456"
  Type: *data
  Value: 10737418240  # 10GB in bytes
  Expiry: 2025-01-24
  Weight: 10

# كلا الرصيدين يتواجدان بشكل مستقل
# النظام يستهلك بناءً على الوزن؛ إذا كان الوزن متساويًا، فإن الترتيب غير مضمون
```

ترتيب الاستهلاك بناءً CGRateS عندما تكون الأرصدة بنفس الوزن وتطابق نفس الوجهة، لا يضمن على الانتهاء - يعتمد الترتيب على كيفية تخزين الأرصدة واسترجاعها. **استخدم أوزان مختلفة للتحكم في ترتيب الاستهلاك**.

أنواع الإجراءات: *topup مقابل *topup_reset

تحدد نوع الإجراء كيفية تفاعل الأرصدة الجديدة مع الأرصدة الحالية من نفس المعرف.

سلوك *topup - إضافي

الإجراء `*topup` **يضيف إلى الأرصدة الحالية** بنفس معرف الرصيد و**يمدد وقت الانتهاء**.

السلوك:

- يجد رصيدًا موجودًا مع معرف مطابق
- يضيف قيمة جديدة إلى ا��قيمة الحالية

- يحدث وقت الانتهاء إلى وقت انتهاء جديد
- يحافظ على الرصيد الحالي (الترحيل)

مثال:

```
# الحالة الأولية:
Balance:
  ID: "Data_Package__5368709120"
  Value: 1073741824  # 1GB remaining
  Expiry: 2024-12-24 (1 day remaining)

# مع تشغيل إجراء *topup:
Action:
  Identifier: "*topup"
  Balance:
    ID: "Data_Package__5368709120"  # نفس المعرف - يحفز الترحيل
    Value: 5368709120  # 5GB
    ExpiryTime: "+5d"

# النتيجة بعد *topup:
Balance:
  ID: "Data_Package__5368709120"
  Value: 6442450944  # 6GB (1GB + 5GB rolled over)
  OriginalValue: 5368709120  # 5 يزال يظهر 5GB الأصلية
  Value_hr: "6 GB"
  OriginalValue_hr: "5 GB"
  Remaining_hr: "6 GB of 5 GB (1 GB rolled over)"
  Expiry: 2024-12-29 (5 days from now)
```

حالات الاستخدام:

- مكافآت الولاء (إضافة بيانات إضافية إلى الحزمة الحالية)
- ائتمانات التعويض (إضافة إلى رصيد العميل)
- حزم البيانات القابلة للترحيل
- تمديد فترات السماح

سلوك إعادة التعيين - *topup_reset

الإجراء `*topup_reset` بنفس معرف الرصيد **يستبدل الأرصدة الحالية**.

السلوك:

- يجد رصيدًا موجودًا مع معرف مطابق
- يتجاهل القيمة القديمة (لا يوجد ترحيل)
- يحدد الرصيد إلى قيمة جديدة فقط
- يحدث وقت الانتهاء إلى وقت انتهاء جديد

مثال:

```
# الحالة الأولية:
Balance:
  ID: "Data_Package__5368709120"
  Value: 1073741824  # 1GB remaining
  Expiry: 2024-12-24 (1 day remaining)

# مع تشغيل إجراء *topup_reset:
Action:
  Identifier: "*topup_reset"
  Balance:
    ID: "Data_Package__5368709120"  # نفس المعرف - يحفز إعادة التعيين
    Value: 5368709120  # 5GB
    ExpiryTime: "+5d"

# النتيجة بعد *topup_reset:
Balance:
  ID: "Data_Package__5368709120"
  Value: 5368709120  # 5GB (1 تجاهل GBتم القديمة)
  OriginalValue: 5368709120
  Value_hr: "5 GB"
  Remaining_hr: "5 GB of 5 GB"
  PercentUsed: 0
  Expiry: 2024-12-29 (5 days from now)
```

حالات الاستخدام:

- حزم متكررة شهرية (إعادة تعيين إلى المبلغ الكامل كل شهر)
- تعبئات ثابتة الحجم (دائمًا استلام المبلغ الدقيق)
- تغييرات الخطط (استبدال رصيد الخطة القديمة بخطة جديدة)
- منع الإساءة (لا يمكن تكديس إضافات غير محدودة)

التحكم في سلوك الرصيد باستخدام معرفات الرصيد

معرف الرصيد هو أمر حاسم لتحديد ما إذا كانت الأرصدة مستقلة أم تتفاعل مع بعضها البعض.

اتفاقية تسمية معرف الرصيد وعرضه القابل للقراءة البشرية

تستخدم OmniCRM اتفاقية تسمية محددة لمعرفات الرصيد التي تشفر كل من الاسم الوصفي والحجم الأصلي. هذا يسمح لـ API بإنشاء حقول قابلة للقراءة البشرية تلقائيًا لواجهة الويب.

نمط معرف الرصيد:

```
{DescriptiveName}__{OriginalSizeInBaseUnits}
```

أمثلة على معرفات الرصيد:

```
# 100 :رصيد البياناتGB
"AU_Data_Domestic__107374182400"
# ينكسر إلى:
# - الجزء الوصفي: "AU_Data_Domestic" (ما هو - النوع/الوجهة)
# - الفاصل: "__" (شرطتان سفليتان)
# - الحجم الأصلي: "107374182400" (100GB in bytes)
# - واجهة المستخدم تظهر: "AU Data Domestic - 100 GB"

# رصيد الصوت: 3000 دقيقة
"AU_Voice_Domestic__1800000000000000"
# - الجزء الوصفي: "AU_Voice_Domestic" (ليس
"AU_Voice_Domestic_3000min")
# - الحجم الأصلي: "1800000000000000" (3000 دقيقة في النانوثانية)
# - واجهة المستخدم تظهر: "AU Voice Domestic - 3000 min"

# رصيد SMS: 3000 رسالة
"AU_SMS_Domestic__3000"
# - الجزء الوصفي: "AU_SMS_Domestic"
# - الحجم الأصلي: "3000" (عدد)
# - واجهة المستخدم تظهر: "AU SMS Domestic - 3000 msgs"
```

مهم: لا تشمل معلومات الحجم في الجزء الوصفي. إنها زائدة عن الحاجة حيث يتم تشفير الحجم بعد `__` و API تقوم بتحويله إلى تنسيق قابل للقراءة البشرية تلقائيًا.

كيف ينشئ API عرضًا قابلاً للقراءة البشرية:

عندما يسترجع API الخاص بـ OmniCRM بيانات الرصيد من CGRateS، فإنه يقوم تلقائيًا بتحليل معرف الرصيد وإنشاء حقول `_hr` (قابلة للقراءة البشرية):

```
{
  "BalanceMap": {
    "*data": [
      {
        "ID": "AU_Data_Domestic__107374182400",
        "Value": 53687091200,
        "ExpiryTime": "2025-01-25T23:59:59Z",
        "Weight": 1200,

        // حقول قابلة للقراءة البشرية التي تم إنشاؤها تلقائيًا:
        "ID_hr": "AU Data Domestic",
        "OriginalValue": 107374182400,
        "OriginalValue_hr": "100 GB",
        "Value_hr": "50 GB",
        "Remaining_hr": "50 GB of 100 GB",
        "PercentUsed": 50,
        "ExpiryTime_hr": "25 Jan 2025 (22 days)"
      }
    ]
  }
}
```

منطق معالجة API:

1. **تحليل معرف الرصيد:**

```
balance_id = "AU_Data_Domestic__107374182400"
parts = balance_id.split("__")

descriptive_name = parts[0]  # "AU_Data_Domestic"
original_size = int(parts[1]) if len(parts) > 1 else None
# 107374182400
```

2. **إنشاء اسم وصفي قابل للقراءة البشرية:**

```
# استبدال الشرطات السفلية بالمسافات
id_hr = descriptive_name.replace("_", " ")  # "AU Data
Domestic"
```

3. **تحويل الحجم الأصلي إلى وحدات بشرية:**

```
# (بايت) لأرصدة البيانات
if balance_type == "*data":
    original_value_hr = convert_bytes_to_gb(original_size)
# "100 GB"

# (نانوسكند) لأرصدة الصوت
elif balance_type == "*voice":
    original_value_hr =
convert_ns_to_minutes(original_size)  # "3000 min"

# لأرصدة SMS (عدد)
elif balance_type == "*sms":
    original_value_hr = f"{original_size} msgs"  # "3000
msgs"
```

4. **حساب نسبة الاستخدام:**

```
if original_size and original_size > 0:
    percent_used = ((original_size - current_value) /
original_size) * 100
```

5. **تنسيق العرض المتبقي:**

```
remaining_hr = f"{current_value_hr} of
{original_value_hr}"
# "50 GB of 100 GB"
```

عرض واجهة الويب:

لعرض معلومات الرصيد بطريقة سهلة الاستخدام `_hr` تستخدم الواجهة الأمامية هذه الحقول:

```
// بدلاً من عرض القيم الخام:
// ID: "AU_Data_Domestic__107374182400"
// Value: 53687091200

// عرض قابل للقراءة البشرية:
<BalanceCard>
  <Title>{balance.ID_hr}</Title>           {/* "AU Data Domestic" */}
  <Progress value={balance.PercentUsed}>   {/* 50% */}
    {balance.Remaining_hr}                 {/* "50 GB of 100 GB" */}
  </Progress>
  <Expiry>{balance.ExpiryTime_hr}</Expiry> {/* "25 Jan 2025 (22 days)" */}
</BalanceCard>
```

لماذا هذا مهم:

1. **تتبع الحجم الأصلي** - حتى عندما يتم استهلاك الأرصدة جزئ��ا، يمكن للواجهة عرض "50 GB من 100 GB" بدلاً من "50 GB المتبقية" فقط
2. **تصور التقدم** - تتيح حسابات النسبة الدقيقة عرض شريط التقدم بدقة
3. **تسمية متسقة** - تضمن الأسماء الوصفية المستخرجة من معرفات الرصيد التناسق بين الواجهة الخلفية والواجهة الأمامية
4. **عرض الترحيل** - عند استخدام `*topup` (الترحيل)، 70 GB إذا كان لدى العميل 100 GB المتبقية وقام بتعبئة:
   - يبقى معرف الرصيد: `"AU_Data_Domestic__107374182400"` (100 GB الأصلية)
   - تصبح القيمة الحالية: 170 GB
   - تظهر الواجهة: "170 GB (70 GB 100 + ترحيل GB جديدة)"

أفضل ممارسة - إنشاء معرفات الرصيد:

قم دائمًا بتضمين الحجم الأصلي بعد `__` لعرض واجهة المستخدم بشكل صحيح. لا تكرر معلومات الحجم في الاسم الوصفي:

```
# جيد - اسم وصفي + حجم في وحدات أساسية
Action_Data_100GB = {
    "Actions": [
        {
            "BalanceId": f"AU_Data_Domestic__{100 * 1024 * 1024 *
1024}",
            "Units": 100 * 1024 * 1024 * 1024
        }
    ]
}

# سيء - حجم زائد في الاسم الوصفي
Action_Data_100GB = {
    "Actions": [
        {
            "BalanceId": f"AU_Data_Domestic_100GB__{100 * 1024 *
1024 * 1024}",  # زائد!
            "Units": 100 * 1024 * 1024 * 1024
        }
    ]
}

# سيء - لا توجد معلومات حجم (لا يمكن للواجهة حساب النسبة)
Action_Data_100GB = {
    "Actions": [
        {
            "BalanceId": "AU_Data_Domestic",  # مفقود __size
            "Units": 100 * 1024 * 1024 * 1024
        }
    ]
}
```

حالة خاصة - الأرصدة النقدية:

لا تتضمن الأرصدة النقدية عادةً الحجم الأصلي لأنها يمكن أن يتم تعبئتها بأي مبلغ:

```
# رصيد نقدي بدون تشفير الحجم
{
    "BalanceId": "PAYG_Monetary_Balance",
    "BalanceType": "*monetary",
    "Units": 5000  # $50.00
}

# تعرض الواجهة ببساطة الرصيد الحالي بدون نسبة
# "50.00$ :الرصيد"
```

الاستراتيجية 1: معرفات فريدة (أرصدة مستقلة)

استخدم معرفات فريدة (مثل UUIDs) لإنشاء أرصدة مستقلة تمامًا لا تتفاعل أبدًا.

مثال على التنف��ذ:

```
- name: Generate unique balance identifier
  set_fact:
    uuid: "{{ 99999999 | random | to_uuid }}"
    balance_id: "Data_5days__5368709120_{{ uuid[0:8] }}"

- name: Add independent balance with *topup
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "{{ balance_id }}",
            "Value": 5368709120,
            "ExpiryTime": "+5d",
            "Weight": 10
          }
        }]
      }
```

النتيجة: كل إضافة تنشئ رصيدًا جديدًا ومنفصلًا حتى لو قام العميل بشراء نفس الإضافة عدة مرات.

```yaml
# العميل يشتري "إضافة بيانات 5 أيام" ثلاث مرات:
Balance 1:
  ID: "Data_5days__5368709120_a1b2c3d4"
  Value: 5368709120
  Value_hr: "5 GB"
  OriginalValue_hr: "5 GB"
  Remaining_hr: "5 GB of 5 GB"
  Expiry: 2024-12-29

Balance 2:
  ID: "Data_5days__5368709120_e5f6g7h8"
  Value: 5368709120
  Value_hr: "5 GB"
  Remaining_hr: "5 GB of 5 GB"
  Expiry: 2024-12-30

Balance 3:
  ID: "Data_5days__5368709120_i9j0k1l2"
  Value: 5368709120
  Value_hr: "5 GB"
  Remaining_hr: "5 GB of 5 GB"
  Expiry: 2024-12-31

# 15 GB الإجمالي المتاح: عبر ثلاثة أرصدة منفصلة
# 5 - أيام 5 "بيانات" كـ الواجهة في فردي بشكل منها يظهر GB كل 5 من GB"
```

الاستراتيجية 2: معرفات مشتركة مع *topup (الترحيل)

استخدم نفس معرف الرصيد مع إجراء `*topup` للسماح بترحيل الرصيد وتمديد الانتهاء.

مثال على التنفيذ:

```
- name: Set fixed balance ID
  set_fact:
    balance_id: "Data_5days__5368709120"

- name: Add balance with *topup (rollover)
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "Data_5days__5368709120",
            "Value": 5368709120,
            "ExpiryTime": "+5d",
            "Weight": 10
          }
        }]
      }
```

النتيجة: عمليات الشراء اللاحقة تضيف إلى الرصيد الحالي وتمدد الانتهاء.

```yaml
# اليوم الأول: العميل يشتري "إضافة بيانات 5 أيام"
Balance:
  ID: "Data_5days__5368709120"
  Value: 5368709120
  Value_hr: "5 GB"
  Remaining_hr: "5 GB of 5 GB"
  PercentUsed: 0
  Expiry: 2024-12-29

# اليوم الثالث: العميل استخدم 1GB، ثم يشتري نفس الإضافة مرة أخرى:
Balance:
  ID: "Data_5days__5368709120"
  Value: 9663676416  # 4GB 5 + جديدة GB المتبقية GB
  Value_hr: "9 GB"
  OriginalValue_hr: "5 GB"
  Remaining_hr: "9 GB (4 GB 5 + ترحيل GB جديدة)"
  PercentUsed: -80  # سالب يشير إلى الترحيل
  Expiry: 2024-12-27 (5 أيام جديدة من اليوم)
```

الاستراتيجية 3: معرفات مشتركة مع *topup_reset (مبلغ ثابت)

استخدم نفس معرف الرصيد مع إجراء `*topup_reset` لإعادة تعيينه دائمًا إلى مبلغ ثابت.

مثال على التنفيذ:

```
- name: Set fixed balance ID
  set_fact:
    balance_id: "Monthly_Plan__32212254720"

- name: Add balance with *topup_reset (no rollover)
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.SetActions",
        "params": [{
          "ActionsId": "Action_Reset_Monthly_Plan",
          "Actions": [{
            "Identifier": "*topup_reset",
            "BalanceType": "*data",
            "Units": 32212254720,  # 30GB
            "ExpiryTime": "*monthly",
            "DestinationIds": "*any",
            "BalanceId": "Monthly_Plan__32212254720",
            "Weight": 10
          }]
        }]
      }
```

النتيجة: في كل شهر، يتم إعادة تعيين الرصيد إلى 30 GB تمامًا بغض النظر عن مقدار الاستخدام.

```yaml
# الشهر الأول، اليوم الأول:
Balance:
  ID: "Monthly_Plan__32212254720"
  Value: 32212254720
  Value_hr: "30 GB"
  Remaining_hr: "30 GB of 30 GB"
  PercentUsed: 0
  Expiry: 2024-12-31

# الشهر الأول، اليوم 25: العميل استخدم 28 GB
Balance:
  ID: "Monthly_Plan__32212254720"
  Value: 2147483648
  Value_hr: "2 GB"
  Remaining_hr: "2 GB of 30 GB"
  PercentUsed: 93
  Expiry: 2024-12-31

# الشهر الثاني، اليوم الأول: ActionPlan يقوم بتشغيل *topup_reset
Balance:
  ID: "Monthly_Plan__32212254720"
  Value: 32212254720  # 2 GB غير مستخدمة ضاعت، إعادة تعيين إلى الكامل
  Value_hr: "30 GB"
  Remaining_hr: "30 GB of 30 GB"
  PercentUsed: 0
  Expiry: 2025-01-31
```

معطلات الرصيد

تسمح لك بـ **حظر أو تحديد الاستخدام** حتى CGRateS **معطلات الرصيد** هي ميزة قوية في عندما توجد أرصدة. يمنع رصيد المعطل الاستهلاك عندما يصل إلى الصفر، مما يمنع المزيد من الاستخدام بغض النظر عن الأرصدة المتاحة الأخرى.

كيف تعمل المعطلات

عندما يحتوي الرصيد على `Blocker: true`:

1. **بينما يحتوي المعطل على قيمة** → يُسمح بالاستخدام حتى مقدار المعطل

2. **عندما يصل المعطل إلى الصفر** → يتوقف كل الاستخدام، حتى لو كانت هناك أرصدة أخرى موجودة
3. **يتم إرجاع خطأ** → `INSUFFICIENT_CREDIT_BALANCE_BLOCKER` يمنع الجلسة

الخصائص الرئيسية:

- يتم **التحقق من أرصدة المعطل حتى عندما تكون القيمة صفرًا** (على عكس الأرصدة العادية التي يتم تخطيها)
- **يتوقف CGRateS عن** عند��ا يتم مواجهة معطل مع طلب استخدام متبقي، **يتوقف المعالجة** ويعيد خطأ
- تعمل المعطلات مع جميع أنواع الأرصدة: `*voice`, `*data`, `*sms`, `*monetary`

حالات الاستخدام للمعطلات

1. تعليق الحساب

حظر كل الاستخدام عندما يتم تعليق الحساب (على سبيل المثال، فشل الدفع):

```
Action_Suspend_Account = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_suspend-account",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            # إضافة معطل بقيمة صفرية لمنع كل الاستخدام
            {
                "Identifier": "*topup",
                "BalanceId": "Suspension_Blocker",
                "BalanceType": "*monetary",
                "DestinationIDs": "*any",
                "Units": 0,  # قيمة صفرية
                "BalanceWeight": 9999,  # أعلى أولوية - يتم التحقق منها أولاً
                "Blocker": True,  # حظر كل الاستخدام
                "Weight": 10
            }
        ]
    }]
}

result = OCS_Obj.SendData(Action_Suspend_Account)
```

النتيجة: جميع المكالمات / البيانات / الرسائل القصيرة محظورة بغض النظر عن الأرصدة الأخرى.

2. حدود الإنفاق

تحديد الحد الأقصى للإنفاق لمنع صدمة الفاتورة:

```
Action_Monthly_Plan_With_Cap = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_monthly-with-cap",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            # 10GB بيانات مشمولة
            {
                "Identifier": "*topup_reset",
                "BalanceId": f"Included_Data__{10 * 1024 * 1024 *
1024}",
                "BalanceType": "*data",
                "DestinationIDs": "Dest_PLMN_OnNet",
                "Units": 10 * 1024 * 1024 * 1024,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,  # تم استهلاكه أولاً
                "Weight": 95
            },
            # حد تجاوز 50 دولار (حاجز)
            {
                "Identifier": "*topup_reset",
                "BalanceId": "Overage_Cap",
                "BalanceType": "*monetary",
                "DestinationIDs": "*any",
                "Units": 5000,  # 50.00 دولار كحد أقصى
                "ExpiryTime": "*month",
                "BalanceWeight": 1000,  # تم استهلاكه بعد المشمول
                "Blocker": True,  # توقف عند إنفاق 50 دولار
                "Weight": 90
            }
        ]
    }]
}
```

التدفق:

1. العميل يستخدم 10GB مجانية → مشمولة (من Included_Data_10GB)
2. العميل يستخدم 5GB إضافية → يتم الخصم من Overage_Cap (أسعار PAYG)

3. عندما يصل Overage_Cap (تم الوصول إلى حد الإنفاق) → **جميع الاستخدامات محظورة** إلى 0 دولار

3. تجربة مجانية محدودة زمنياً

توفير استخدام مجاني محدود لحسابات التجربة:

```
Action_Trial_Account = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_trial-100-minutes",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            # 100 دقيقة مجانية (حاجز - يتوقف عند استنفادها)
            {
                "Identifier": "*topup",
                "BalanceId": f"Trial_Voice__{100 * 60 *
1000000000}",
                "BalanceType": "*voice",
                "DestinationIDs": "Dest_Domestic_All",
                "Units": 100 * 60 * 1000000000,  # 100 دقيقة
                "ExpiryTime": "+720h",  # 30 يوماً
                "BalanceWeight": 1200,
                "Blocker": True,  # لا استخدام بعد 100 دقيقة
                "Weight": 10
            }
        ]
    }]
}
```

النتيجة: يحصل العميل على 100 دقيقة مجانية بالضبط. بعد ذلك، يتم حظر جميع المكالمات (لا رسوم تلقائية).

4. حظر وجهات معينة

حظر وجهات معينة مع السماح لأخرى:

```
Action_Block_Premium_Numbers = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_block-premium",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            # الاستخدام العادي مسموح
            {
                "Identifier": "*topup_reset",
                "BalanceId": "Regular_Usage",
                "BalanceType": "*monetary",
                "DestinationIDs": "Dest_Domestic_All",
                "Units": 10000,  # 100 دولار
                "ExpiryTime": "*month",
                "BalanceWeight": 1000,
                "Weight": 20
            },
            # حظر الأرقام المميزة (0900، إلخ.)
            {
                "Identifier": "*topup",
                "BalanceId": "Premium_Blocker",
                "BalanceType": "*monetary",
                "DestinationIDs": "Dest_Domestic_Premium",
                "Units": 0,  # قيمة صفرية
                "BalanceWeight": 2000,  # وزن أعلى - يتم التحقق منه أولاً للأرقام المميزة
                "Blocker": True,
                "Weight": 10
            }
        ]
    }]
}
```

التدفق:

- المكالمات المحلية → تستخدم رصيد الاستخدام العادي
- مكالمات الأرقام المميزة → تطابق Premium_Blocker (الوزن 2000 > 1000) → **محظور**

الحاجز مقابل الأرصدة المعطلة

لا تخلط بين `Blocker` و `Disabled`:

الميزة	الحاجز	المعطل
الغرض	إيقاف الاستخدام عند نفاد الرصيد	إيقاف رصيد مؤقتاً
عندما تكون القيمة > 0	الرصيد قابل للاستخدام بشكل طبيعي	يتم تخطي الرصيد / تجاهله
عندما تكون القيمة = 0	**يمنع جميع الاستخدامات التالية**	يتم تخطي الرصيد (يتم تجربة الرصيد التالي)
حالة الاستخدام	حدود الإنفاق، الحدود، حدود التجربة	تعليق رصيد معين مؤقتاً

```
# 10 يسمح بـ :الحاجزGB، ثم يحظر كل شيء
{
    "BalanceId": f"Data_Cap__{10 * 1024 * 1024 * 1024}",
    "Units": 10 * 1024 * 1024 * 1024,
    "Blocker": True  # 10 عند الاستخدام الايقافGB
}

# (معلق مؤقتاً) المعطل: يتجاهل هذا الرصيد تماماً
{
    "BalanceId": f"Bonus_Data__{5 * 1024 * 1024 * 1024}",
    "Units": 5 * 1024 * 1024 * 1024,
    "Disabled": True  # لن يتم استخدام هذا الرصيد على الإطلاق
}
```

مثال عملي: خطة هجينة مع حد أمان

دمج الأرصدة الوحدوية، والتجاوز النقدي، وحد الحاجز:

```
Action_Safe_Hybrid_Plan = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_safe-hybrid-plan",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            {
                "Identifier": "*reset_account",
                "Weight": 700
            },
            # 500 دقيقة محلية مشمولة
            {
                "Identifier": "*topup_reset",
                "BalanceId": f"Domestic_Voice__{500 * 60 *
1000000000}",
                "BalanceType": "*voice",
                "DestinationIDs": "Dest_Domestic_All",
                "Units": 500 * 60 * 1000000000,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 95
            },
            # 20 دولار حد تجاوز
            {
                "Identifier": "*topup_reset",
                "BalanceId": "Overage_Allowance",
                "BalanceType": "*monetary",
                "DestinationIDs": "*any",
                "Units": 2000,  # 20.00 دولار
                "ExpiryTime": "*month",
                "BalanceWeight": 1000,
                "Weight": 90
            },
            # 50 (حاجز) دولار حد صارم
            {
                "Identifier": "*topup_reset",
                "BalanceId": "Hard_Spending_Cap",
                "BalanceType": "*monetary",
                "DestinationIDs": "*any",
                "Units": 5000,  # 50.00 دولار كحد أقصى مطلق
                "ExpiryTime": "*month",
```

```
                "BalanceWeight": 500,  # أقل من التجاوز - يستخدم أخيراً
                "Blocker": True,  # توقف عند الوصول إلى الحد
                "Weight": 85
            }
        ]
    }]
}
```

رحلة العميل:

1. **0-500 دقيقة:** يستخدم Domestic_Voice_500min (مجاني)
2. **500-700 دقيقة:** يستخدم Overage_Allowance 20 = بسعر 0.10 دولار للدقيقة دولار (200 دقيقة)
3. **700-1200 دقيقة:** يستخدم Hard_Spending_Cap = بسعر 0.10 دولار للدقيقة 50 دولار (500 دقيقة)
4. **عند 1200 دقيقة:** تم استنفاد Hard_Spending_Cap → **جميع الاستخدامات محظورة**

يحصل العميل على 1200 دقيقة إجمالية، مع حد إنفاق قدره 50 دولار كحد أقصى للتجاوز.

أفضل الممارسات مع الحواجز

1. **استخدم وزنًا عاليًا للحواجز**

   ```
   "BalanceWeight": 9999  # تأكد من فحص الحاجز أولاً
   ```

2. **حواجز بقيمة صفرية للحظر الفوري**

   ```
   "Units": 0,  # حظر فوري
   "Blocker": True
   ```

3. **إخطار العملاء قبل استنفاد الحاجز**
   - استخدم ActionTriggers لإرسال إشعارات عند 80%، 90%، 100% من استخدام الحاجز
   - أعط العملاء خيار زيادة الحد قبل حدوث الحظر

4. **إزالة الحواجز عند إلغاء التعليق**

```
# استخدم *remove_balance لحذف الحاجز
{
    "Identifier": "*remove_balance",
    "BalanceId": "Suspension_Blocker"
}
```

5. **اختبار سلوك الحاجز**

   - تحقق من أن الحاجز يعيد خطأ `INSUFFICIENT_CREDIT_BALANCE_BLOCKER`
   - تأكد من أن CDRs عند الحظر تظهر التكلفة = 1.0-
   - اختبر أن الأرصدة الأخرى لا تُستخدم بعد استنفاد الحاجز

استكشاف أخطاء الحواجز

المشكلة: الاستخدام غير محظور على الرغم من أن الحاجز عند الصفر

الأسباب المحتملة:

1. وزن الحاجز منخفض جداً (تم التحقق من الأرصدة الأخرى أولاً)
2. DestinationIDs لا تتطابق مع وجهة الاستخدام
3. لم يتم تعيين حقل الحاجز إلى `True`

الحل:

```
# تحقق من تكوين الحاجز
OCS_Obj.SendData({
    'method': 'ApierV2.GetAccount',
    'params': [{"Tenant": tenant, "Account": "service_uuid"}]
})

# تحقق من:
# - الحاجز: true
# - BalanceWeight (مثل 9999) هو الأعلى
# - DestinationIDs تشمل وجهة الاستخدام
# - القيمة = 0
```

المشكلة: الاستخدام محظور بشكل غير متوقع

السبب المحتمل: تم إنشاء رصيد الحاجز عن غير قصد بقيمة منخفضة

الحل: تحقق من جميع الأرصدة بحثاً عن `Blocker: true` وتحقق من أن قيمها مناسبة لحالة الاستخدام الخاصة بك.

قواعد استهلاك الرصيد

القاعدة 1: أولوية دقة الوجهة

يتم استهلاك الأرصدة ذات **دقة الوجهة الأعلى** (تطابق وجهة أكثر تحديداً) أولاً. يتم تحديد ذلك من خلال طول تطابق البادئة.

```
# العميل يتصل بـ +44-20-1234-5678 (لندن، المملكة المتحدة)

1 الرصيد:
  DestinationIDs: "Dest_UK_London"  # بادئة: "4420" (الدقة: 4)
  القيمة: 100 دقيقة
  الوزن: 10

2 الرصيد:
  DestinationIDs: "Dest_UK_All"     # بادئة: "44" (الدقة: 2)
  القيمة: 200 دقيقة
  الوزن: 10

# يتم استهلاك الرصيد 1 أولاً (الدقة 4 > الدقة 2)
# تطابق الوجهة الأكثر تحديداً يفوز
```

حالة الاستخدام: الأرصدة المحددة حسب المدينة أو المنطقة تأخذ الأولوية على الأرصدة التي تشمل البلاد.

القاعدة 2: أولوية الوزن (نفس الدقة)

عندما تكون دقة الوجهة متساوية، يتم استهلاك الأرصدة ذات **الوزن الأعلى** أولاً.

```
1 الرصيد:
  ID: "Premium_Data"
  5 :القيمةGB
  20 :الوزن

2 الرصيد:
  ID: "Standard_Data"
  10 :القيمةGB
  10 :الوزن

# (10 الوزن < 20 الوزن) أولاً 1 يتم استهلاك الرصيد
# على الرغم من أن الرصيد 2 يحتوي على المزيد من البيانات
```

حالة الاستخدام: الأرصدة ذات الأولوية (تستهلك البيانات الإضافية قبل البيانات العادية).

القاعدة 3: الأقدم أولاً

عندما يتطابق الوزن، يتم استخدام الرصيد الأ��دم أولاً.

```
1 الرصيد:
  ID: "Data_Package_A"
  5 :القيمةGB
  25-12-2024 :تاريخ انتهاء الصلاحية
  10 :الوزن

2 الرصيد:
  ID: "Data_Package_B"
  10 :القيمةGB
  15-01-2025 :تاريخ انتهاء الصلاحية
  10 :الوزن
```

للتحكم في ترتيب الاستهلاك، استخدم أوزان مختلفة:

```
# الطريقة الصحيحة: استخدم الوزن لإعطاء الأولوية للأرصدة التي ستنتهي قريباً
1 الرصيد:
  ID: "Data_Package_A"
  5 :القيمةGB
  25-12-2024 :تاريخ انتهاء الصلاحية
  الوزن: 11  # وزن أعلى = يتم استهلاكه أولاً

2 الرصيد:
  ID: "Data_Package_B"
  10 :القيمةGB
  15-01-2025 :تاريخ انتهاء الصلاحية
  الوزن: 10  # وزن أقل = يتم استهلاكه ثانياً

# يتم استهلاك الرصيد 1 أولاً (الوزن 11 > الوزن 10)
```

أفضل ممارسة: إذا كنت ترغب في ضمان استهلاك الأرصدة التي ستنتهي قريباً أولاً، قم بتعيين أوزان أعلى لها عند إنشائها.

أمثلة عملية

المثال 1: إضافة بيانات بسيطة (مستقلة)

السيناريو: يمكن للعميل شراء إضافة "5GB لمدة 5 أيام" عدة مرات، كل منها ينشئ رصيداً منفصلاً.

التنفيذ:

```
- name: توليد UUID فريد لرصيد
  set_fact:
    uuid: "{{ 99999999 | random | to_uuid }}"

- name: إضافة رصيد 5 GB مستقل
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Categories": "*any",
          "Balance": {
            "ID": "Data_5GB_{{ uuid[0:8] }}",
            "Value": 5368709120,
            "ExpiryTime": "+120h",  # 5 أيام
            "Weight": 10
          }
        }]
      }
```

تجربة العميل:

- يشتري الإضافة في 24 ديسمبر GB 5 ← يحصل على رصيد تنتهي في 29 ديسمبر
- يشتري الإضافة في 25 ديسمبر GB 5 ← يحصل على رصيد تنتهي في 30 ديسمبر
- كلا الرصيدين يتواجدان؛ يعتمد ترتيب الاستهلاك على أوزان الأرصدة (كلاهما له وزن 10، لذا لا يوجد ضمان لترتيب معين)

المثال 2: حزمة بيانات قابلة للتدوير

السيناريو: خطة "50 GB شهرية" حيث يتم تدوير البيانات غير المستخدمة عندما يقوم العميل بشحن رصيده مبكراً.

التنفيذ:

```
- name: إضافة رصيد بيانات قابل للتدوير
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "Rollover_Monthly_50GB",
            "Value": 53687091200,  # 50GB
            "ExpiryTime": "+720h",  # 30 يوماً
            "Weight": 10
          }
        }]
      }
```

الافتراضي (تمكين التدوير) `*topup` **نوع الإجراء:** يستخدم سلوك

تجربة العميل:

- تنتهي في 30 يوماً GBاليوم 1: يحصل على 50
- متبقية GBلديه 20، GBاليوم 20: استخدم 30
- إجمالية GBاليوم 20: يقوم بشحن رصيده مرة أخرى ← يحصل على 70 (20GB + 50GB)، ��20 تم تمديد انتهاء الصلاحية إلى +30 يوماً من اليوم

المثال 3: خطة شهرية ثابتة (بدون تدوير)

السيناريو: خطة "100GB شهرية غير محدودة" تعيد تعيين الرصيد إلى 100 GB بالضبط كل شهر، بدون تدوير.

التنفيذ:

```
- name: إنشاء إجراء إعادة تعيين شهرية
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.SetActions",
        "params": [{
          "ActionsId": "Action_Monthly_100GB_Reset",
          "Overwrite": true,
          "Actions": [{
            "Identifier": "*topup_reset",
            "BalanceType": "*data",
            "Units": 107374182400,  # 100GB
            "ExpiryTime": "*monthly",
            "BalanceId": "Monthly_Plan__107374182400",
            "Weight": 10
          }]
        }]
      }

- name: إنشاء خطة الإجراء الشهرية
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.SetActionPlan",
        "params": [{
          "Id": "ActionPlan_Monthly_100GB",
          "ActionPlan": [{
            "ActionsId": "Action_Monthly_100GB_Reset",
            "Time": "*monthly",
            "Weight": 10
          }],
          "Overwrite": true,
          "ReloadScheduler": true
        }]
      }

- name: تعيين خطة الإجراء للحساب
```

```
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV2.SetAccount",
        "params": [{
          "Account": "{{ service_uuid }}",
          "ActionPlanIds": ["ActionPlan_Monthly_100GB"],
          "ReloadScheduler": true
        }]
      }
```

تجربة العميل:

- يحصل على 100 GB في الشهر 1: 95 GB مستخدم، 5 GB متبقية لديه
- يتم إعادة تعيين الرصيد إلى 100 GB في الشهر 2 (5 GB غير مستخدمة فقدان)
- يستخدم 20 GB في الشهر 2: 80 GB متبقية لديه
- يتم إعادة تعيين الرصيد إلى 100 GB في الشهر 3 (80 GB غير مستخدمة فقدان)

المثال 4: أرصدة متعددة المستويات مع أولوية الوزن

السيناريو: لدى العميل "بيانات إضافية" (أولوية عالية) و"بيانات عادية" (أولوية منخفضة). يتم استهلاك البي��نات الإضافية أولاً.

التنفيذ:

```
# إضافة بيانات إضافية بوزن عالٍ
- name: إضافة رصيد بيانات إضافية
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "Bonus_Data",
            "Value": 5368709120,  # 5GB
            "ExpiryTime": "+240h",  # 10 أيام
            "Weight": 20  # أولوية أعلى
          }
        }]
      }

# إضافة بيانات عادية بوزن عادي
- name: إضافة رصيد بيانات عادية
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "Regular_Data",
            "Value": 53687091200,  # 50GB
            "ExpiryTime": "+720h",  # 30 يوماً
            "Weight": 10  # أولوية عادية
          }
        }]
      }
```

تجربة العميل:

- 5 لديهGB 50 + (وزن 20) GBبيانات إضافية (وزن 10) GB بيانات عادية
- يستهلك جميع بيانات الإضافات أولاً
- 50 ثم يستهلك من مجموعة بيانات العاديةGB
- يتم الحفاظ على البيانات العادية لفترة أطول

معرفات الإجراءات الشائعة

يدعم CGRateS عدة معرفات للإجراءات لعمليات مختلفة:

معالجة الرصيد

***topup** - إضافة إلى الرصيد الحالي (التدوير) ***topup_reset** - إعادة تعيين الرصيد إلى قيمة جديدة (بدون تدوير) ***debit** - خصم من الرصيد ***debit_reset** - تعيين الرصيد إلى قيمة سلبية ***reset_account** - إزالة جميع الأرصدة

تصميم منتجات الإضافات

عند تصميم منتجات الإضافات في OmniCRM، ضع في اعتبارك هذه الأسئلة:

السؤال 1: هل يجب أن تتراكم الأرصدة؟

لا (استبدال) → استخدم معرف (UUID مع) **نعم (مستقلة)** → استخدم معرفات رصيد فريدة
`*topup` **نعم (تدوير)** → استخدم معرف رصيد ثابت مع `*topup_reset` رصيد ثابت مع

السؤال 2: ماذا يحدث للرصيد غير المستخدم؟

مجموعات `*topup_reset` **فقدان** → استخدم إجراء `*topup` **تدوير** → استخدم إجراء
منفصلة → استخدم معرفات رصيد فريدة

السؤال 3: كيف يجب استهلاك الأرصدة؟

الأقدم أولاً → استخدم أوزان مختلفة (قم بتعيين وزن أعلى للأرصدة الأقدم / التي ستنتهي قريباً)
الأولوية أولاً → أوزان مختلفة (وزن أعلى = أولوية أعلى) **ترتيب محدد** → استخدم الأوزان: 30

(أولوية)، 20 (إضافات)، 10 (عادية) **عشوائي / لا تفضيل** ← استخدم نفس الوزن (ترتيب الاستهلاك غير مضمون)

السؤال 4: ما هي استراتيجية انتهاء الصلاحية؟

نهاية الشهر ← استخدم (لمدة 30 يوماً `+720h`) **مدة ثابتة** ← استخدم انتهاء الصلاحية النسبي أو مدة طويلة جداً `*unlimited` في خطة الإجراء **لا تنتهي أبداً** ← استخدم `*monthly`

هيكل إجراء CGRateS في Playbooks

إليك الهيكل الكامل لإنشاء إجراء:

```
- name: إجراء إنشاء CGRateS
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.SetActions",
        "params": [{
          "ActionsId": "Action_Name_Here",
          "Overwrite": true,  # استبدال إذا كان موجوداً
          "Actions": [
            {
              "Identifier": "*topup",  # أو *topup_reset
              "BalanceType": "*data",  # *data، *voice، *sms، *monetary
              "Units": 5368709120,  # المبلغ المضاف
              "ExpiryTime": "+120h",  # +Xh، *unlimited، *monthly
              "DestinationIds": "*any",  # عادة *any
              "BalanceId": "Balance_Name",  # فريد أو مشترك
              "Weight": 10,  # الأولوية (أعلى = يتم استهلاكه أولاً)
              "Blocker": false,  # إذا كانت true، تمنع الرصيد السلبي
              "Disabled": false,  # إذا كانت true، الرصيد معطل
              "SharedGroups": ""  # مجموعة الرصيد المشتركة (اختياري)
            },
            {
              "Identifier": "*cdrlog",  # سجل هذا الإجراء كـ CDR
              "BalanceType": "*generic",
              "ExtraParameters": "{\"Category\":\"^activation\",\"Destination\":\"Addon Name\"}"
            }
          ]
        }]
      }
```

أوصاف الحقول

ActionsId - معرف فريد لمجموعة الإجراءات هذه

Identifier - نوع العملية (`*topup`، `*topup_reset`، `*cdrlog`، إلخ.)

BalanceType - نوع الرصيد:

- `*data` - أرصدة البيانات (بايت)
- `*voice` - أرصدة الصوت (ثواني)
- `*sms` - أرصدة الرسائل القصيرة (عدد)
- `*monetary` - أرصدة نقدية (وحدات العملة)
- `*generic` - أرصدة عامة

Units - المبلغ المضاف / المخصوم (بالوحدات الأساسية: بايت للبيانات، ثواني للصوت)

ExpiryTime - متى ينتهي الرصيد:

- `+Xh` - نسبي (`720+` مثل `h` = 30 يوماً)
- `*unlimited` - لا تنتهي أبداً
- `*monthly` - نهاية الشهر
- `2024-12-31T23:59:59Z` - طابع زمني مطلق

BalanceId - معرف لهذا الرصيد (معرف مشترك = يتفاعل، معرف فريد = مستقل)

Weight - الأولوية (رقم أعلى = أولوية أعلى، يتم استهلاكه أولاً)

Blocker - إذا كانت true، تمنع الحساب من أن يصبح سلبياً

Disabled - إذا كانت true، الرصيد موجود لكن لا يمكن استخدامه

تعريف الإجراءات عبر Python (الإعداد الأولي)

عادةً ما يتم تعريف الإجراءات أثناء تكوين النظام الأولي باستخدام نصوص Python مع مكتبة `cgrateshttpapi`. تعرض هذه الأمثلة كيفية تعريف الإجراءات باستخدام `OCS_Obj.SendData()`.

المتطلبات الأساسية

```
import cgrateshttpapi
import time

OCS_Obj = cgrateshttpapi.CGRateS("ocs.example.com", "2080")
tenant = "your_tenant_name"
tpid = str(tenant) + "_" + str(int(time.time()))
```

تعريف الوجهات

قبل إنشاء الإجراءات، يجب عليك تعريف الوجهات التي تحدد أين يمكن استخدام الأرصدة.

الوجهات تأتي في نوعين:

- **وجهات جغرافية** - بادئات الأرقام للمكالمات الصوتية / الرسائل النصية القصيرة إلى الأماكن (مثل، `Dest_International_UK`)
- **وجهات PLMN** - رموز الشبكة للبيانات من الأماكن (مثل، `Dest_PLMN_OnNet`, `Dest_PLMN_US_Verizon`)

قاعدة حرجة:

- **أرصدة الصوت / الرسائل النصية القصيرة** → استخدم الوجهات الجغرافية (الرقم الذي يتم الاتصال به)
- **أرصدة البيانات** → استخدم وجهات PLMN (الشبكة التي يتصل بها العميل)

لإعداد الوجهات بالكامل بما في ذلك:

- الوجهات الجغرافية (محلية، دولية، مجانية، مميزة)
- وجهات PLMN (شبكات محلية، شبكات تجوال، مناطق)
- قواعد تنسيق PLMN وأفضل الممارسات
- استكشاف مشكلات الوجهات

انظر: **CGRateS** **تكوين وجهات**

حسابات الوحدات

فهم تحويلات الوحدات أمر حاسم لتعريف الأرصدة بشكل صحيح:

```
# أرصدة البيانات (بالبايت)
1_GB = 1 * 1024 * 1024 * 1024  # 1073741824 بايت
100_GB = 100 * 1024 * 1024 * 1024

# أرصدة الصوت (بالتنبيهات)
1_minute = 60 * 1000000000  # 60 مليار تنبيه
3000_minutes = 3000 * 60 * 1000000000

# أرصدة الرسائل النصية القصيرة (بالعدد)
3000_sms = 3000
```

المثال 1: خطة شهرية متعددة الأرصدة (بايثون)

خطة شاملة شهرية تحتوي على أرصدة بيانات وصوت ورسائل نصية قصيرة وتجوال يتم إعادة تعيينها كل شهر.

```
Action_AU_Premium_Plan_1 = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_au-premium-plan-1",
        "Overwrite": True,
        "Tenant": str(tenant),
        "Actions": [
            # أولاً، إعادة تعيين الحساب لمسح الأرصدة القديمة
            {
                "Identifier": "*reset_account",
                "Weight": 700
            },
            # 100 إضافة رصيد بيانات GB
            # مهم: أرصدة البيانات تستخدم وجهات PLMN (الشبكة التي يتصل بها العميل)
            # الخاص بك PLMN وليس الوجهات الجغرافية. استخدم.
            {
                "Identifier": "*topup_reset",
                "BalanceId": "AU_Data_Domestic__" + str(100 * 1024 * 1024 * 1024),
                "BalanceType": "*data",
                "DestinationIDs": "Dest_PLMN_OnNet",  # PLMN الخاص بك (mcc505.mnc057)
                "Units": 100 * 1024 * 1024 * 1024,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 90
            },
            # إضافة رصيد صوت 3000 دقيقة
            {
                "Identifier": "*topup_reset",
                "BalanceId": "AU_Voice_Domestic__" + str(3000 * 60 * 1000000000),
                "BalanceType": "*voice",
                "DestinationIDs": "Dest_AU_Mobile;Dest_AU_Fixed;Dest_AU_TollFree;",
                "Units": 3000 * 60 * 1000000000,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 89
            },
            # إضافة رصيد 3000 رسالة نصية قصيرة
```

```
            {
                "Identifier": "*topup_reset",
                "BalanceId": "AU_SMS_Domestic__" + str(3000),
                "BalanceType": "*sms",
                "DestinationIDs": "Dest_AU_Mobile;",
                "Units": 3000,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 88
            },
            # 6 بيانات تجوال GBإضافة
            {
                "Identifier": "*topup_reset",
                "BalanceId": "AU_Roaming_Data__" + str(6 * 1024 *
1024 * 1024),
                "BalanceType": "*data",
                "DestinationIDs": "Dest_Roaming_All",
                "Units": 6 * 1024 * 1024 * 1024,
                "ExpiryTime": "*month",
                "BalanceWeight": 1100,
                "Weight": 87
            },
            # تسجيل هذا الإجراء كـ CDR
            {
                "Identifier": "*cdrlog",
                "BalanceId": "",
                "BalanceUuid": "",
                "BalanceType": "*generic",
                "Directions": "*out",
                "Units": 0,
                "ExpiryTime": "",
                "Filter": "",
                "TimingTags": "",
                "DestinationIds": "",
                "RatingSubject": "",
                "Categories": "",
                "SharedGroups": "",
                "BalanceWeight": 0,
                "ExtraParameters": "
{\"Category\":\"^activation\",\"Destination\":\"AU Premium Plan
1\"}",
                "BalanceBlocker": "false",
                "BalanceDisabled": "false",
                "Weight": 80
```

```
            }
        ]
    }]
}

# إرسال تعريف الإجراء إلى CGRateS
result = OCS_Obj.SendData(Action_AU_Premium_Plan_1)
assert result['error'] is None or result['error'] == "EXISTS"
print("تم إنشاء الإجراء: Action_au-premium-plan-1")
```

نقاط رئيسية:

- يستخدم `*reset_account` لمسح الأرصدة القديمة أولاً
- يستخدم `*topup_reset` للحدود الشهرية الثابتة (بدون ترحيل)
- تحدد BalanceWeight ترتيب الاستهلاك (محلي 1200 < تجوال 1100)
- تحدد Weight ترتيب التنفيذ داخل الإجراء
- يتضمن `*cdrlog` لتتبع التفعيلات

المثال 2: إضافة بيانات بسيطة (بايثون)

إضافة بيانات بسيطة بسعة 20GB مع تعطيل الترحيل.

```
Action_AU_Data_Addon_20GB = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_au-data-addon-20gb",
        "Overwrite": True,
        "Tenant": str(tenant),
        "Actions": [
            # إعادة تعيين الحساب أولاً
            {
                "Identifier": "*reset_account",
                "Weight": 700
            },
            # 20 بيانات إضافةGB
            # أرصدة البيانات تستخدم وجهات PLMN (أي شبكة يتصل بها العميل)
            {
                "Identifier": "*topup_reset",
                "BalanceId": "AU_Data_Domestic__" + str(20 * 1024
* 1024 * 1024),
                "BalanceType": "*data",
                "DestinationIDs": "Dest_PLMN_OnNet",  # PLMN الخاص
بك (mcc505.mnc057)
                "Units": 20 * 1024 * 1024 * 1024,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 90
            }
        ]
    }]
}

result = OCS_Obj.SendData(Action_AU_Data_Addon_20GB)
assert result['error'] is None or result['error'] == "EXISTS"
print("تم إنشاء الإجراء: Action_au-data-addon-20gb")
```

المثال 3: إضافة صوت دولي (بايثون)

إضافة لدقائق المكالمات الدولية.

```
Action_AU_International_Voice_100min = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_au-international-voice-100min",
        "Overwrite": True,
        "Tenant": str(tenant),
        "Actions": [
            {
                "Identifier": "*reset_account",
                "Weight": 700
            },
            # إضافة 100 دقيقة للمكالمات الدولية
            {
                "Identifier": "*topup_reset",
                "BalanceId": "AU_Voice_International__" + str(100
* 60 * 1000000000),
                "BalanceType": "*voice",
                "DestinationIDs": "Dest_International_All",
                "Units": 100 * 60 * 1000000000,
                "ExpiryTime": "*month",
                "BalanceWeight": 1000,
                "Weight": 90
            }
        ]
    }]
}

result = OCS_Obj.SendData(Action_AU_International_Voice_100min)
assert result['error'] is None or result['error'] == "EXISTS"
print("تم إنشاء الإجراء: Action_au-international-voice-100min")
```

ملاحظة: تستخدم المكالمات الصوتية / الرسائل النصية القصيرة الوجهات الجغرافية (الرقم الذي يتم الاتصال به)، بينما تستخدم البيانات وجهات PLMN (الشبكة التي يتصل بها العميل). انظر تحديد المنتجات لتكوين الوجهات.

مرجع حقول إجراء بايثون

حقول تعريف الإجراء:

- **ActionsId** (مطلوب) - معرف فريد لمجموعة الإجراءات هذه (يجب أن يتطابق مع product_slug في CRM اتفاقية)
- **Overwrite** - إذا كانت True، استبدل الإجراء الحالي بنفس المعرف
- **Tenant** - اسم مستأجر CGRateS
- **Actions** - مصفوفة من الإجراءات الفردية التي سيتم تنفيذها

حقول الإجراء الفردية:

- **Identifier** - نوع الإجراء (`*topup`، `*topup_reset`، `*reset_account`، `*cdrlog`، إلخ.)
- **BalanceId** - معرف فريد لهذا الرصيد (يجب أن يتطابق عبر عمليات الشحن ليعمل الترحيل)
- **BalanceType** - نوع الرصيد (`*data`، `*voice`، `*sms`، `*monetary`)
- **DestinationIDs** - يتحكم في أين يمكن استخدام الرصيد:
  - بالنسبة لـ **الصوت / الرسائل النصية القصيرة**: استخدم الوجهات الجغرافية (مثل، `"Dest_AU_Mobile"`، `"Dest_International_UK"`)
  - بالنسبة لـ **البيانات**: استخدم وجهات PLMN (مثل، `"Dest_PLMN_OnNet"`، `"Dest_PLMN_US_Verizon"`)
- **Units** - المبلغ المراد إضافته (بايت للبيانات، تنبيهات للصوت، عدد للرسائل النصية القصيرة)
- **ExpiryTime** - متى تنتهي صلاحية الرصيد (`*month`، `+720h`، `2024-12-31`، إلخ.)
- **BalanceWeight** - أولوية الاستهلاك (أعلى = يتم استهلاكه أولاً)
- **Weight** - ترتيب التنفيذ داخل مجموعة الإجراءات (أعلى = يتم التنفيذ أولاً)
- **Blocker** (اختياري) - إذا كانت `True`، علامة منطقية؛ تمنع جميع الاستخدامات عندما يصل هذا الرصيد إلى الصفر (انظر حواجز الرصيد)
- **Disabled** (اختياري) - إذا كانت `True`، علامة منطقية؛ هذا تخ��ي / يتم تجاهل الرصيد أثناء الاستهلاك

تنفيذ الإجراءات

بمجرد إنشاء إجراء، قم بتنفيذه على حساب:

```
- name: تنفيذ الإجراء على الحساب
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "APIerSv1.ExecuteAction",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "ActionsId": "Action_Name_Here"
        }]
      }
```

هذا ينفذ جميع العمليات المحددة في الإجراء على الحساب المحدد.

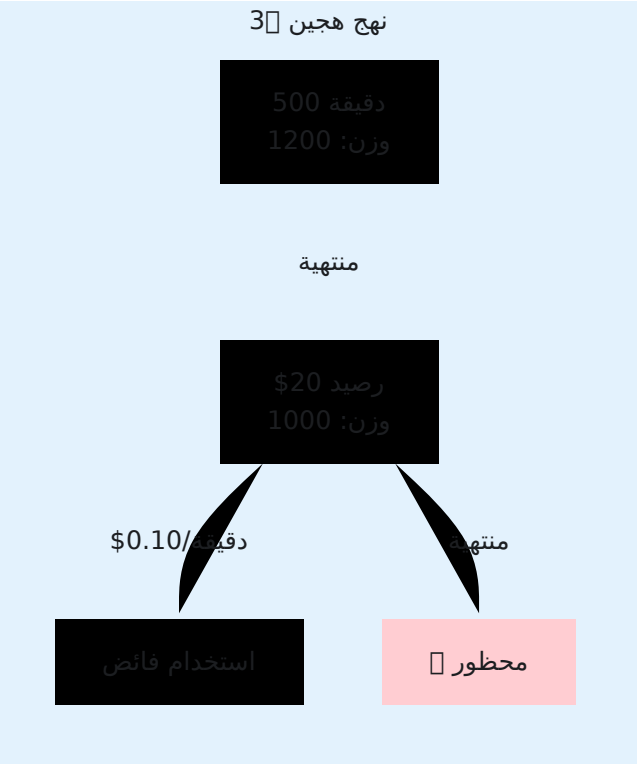
أساليب إدارة الأرصدة

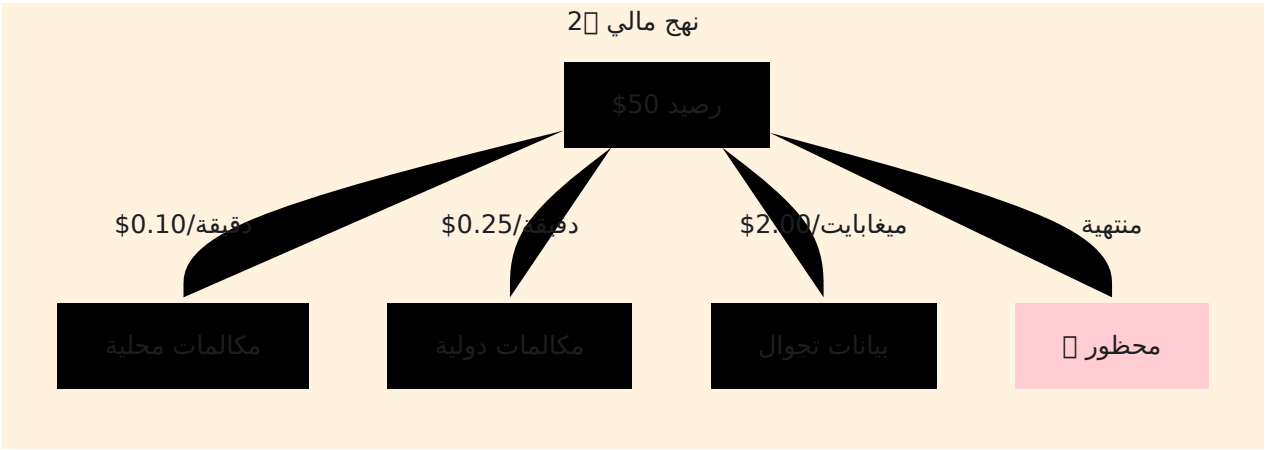
هناك ثلاث طرق رئيسية لإدارة الأرصدة في CGRateS، كل منها له مزايا وعيوب مختلفة لتجربة العملاء، وتوقعات الفوترة، وتعقيد العمليات.

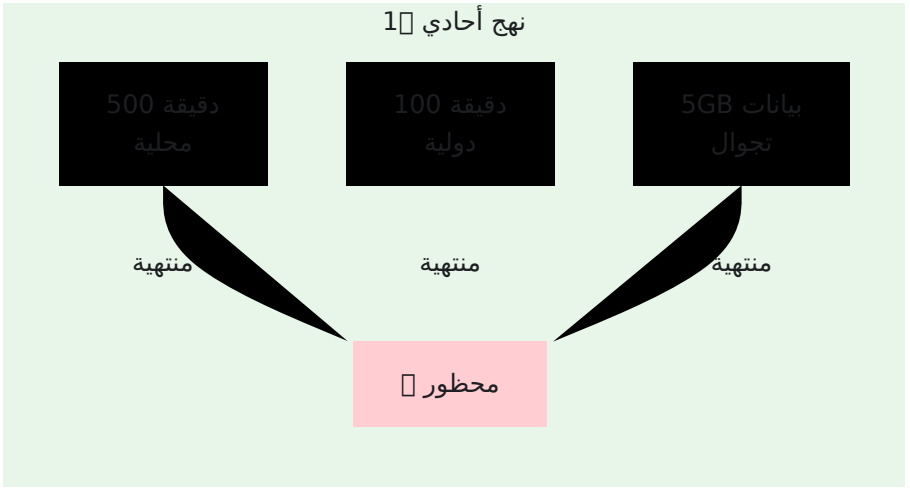
مقارنة: أحادي مقابل مالي مقابل هجين

تلخص هذه الجدول التبادلات بين الثلاثة أساليب للأرصدة:

الميزة	أحادي	مالي (PAYG)	هجين
توقعات العملاء	تكلفة شهرية ثابتة ✅	تكاليف متغيرة ❌	متوقعة إلى حد ⚠ كبير
مرونة الوجهة	محدودة بالوجهات ❌ المضمنة	اتصل / استخدم في أي ✅ مكان	مضمنة + في ✅ أي مكان
معالجة الفائض	قطع صعب ❌	استخدام تلقائي ✅	تدفق تلقائي ✅
خطر صدمة الفاتورة	منخفض (حدود ✅ صارمة)	مرتفع (فواتير غير ❌ محدودة)	معتدل (تدفق ⚠ محدود)
تعقيد التكوين	معتدل (عديد من ⚠ الأرصدة)	بسيط (رصيد واحد) ✅	معقد (كلاهما) ❌
تحسين الإيرادات	انخفاض ⚠ ARPU	ارتفاع ✅ من ARPU المستخدمين الكثيفين	متوازن ARPU ✅
رضا العملاء	مرتفع (بدون ✅ مفاجآت)	منخفض (صدمة ❌ الفاتورة)	مرتفع (أفضل ما ✅ في الاثنين)
الأفضل لـ	المستخدمون المتوقعون	المستخدمون العرضيون	معظم العملاء


3️⃣ نهج هجين
500 دقيقة
وزن: 1200
منتهية
$20 رصيد
وزن: 1000
$0.10/دقيقة
منتهية
استخدام فائض
محظور ⛔


2️⃣ نهج مالي
$50 رصيد
$0.10/دقيقة
$0.25/دقيقة
$2.00/ميغابايت
منتهية
مكالمات محلية
مكالمات دولية
بيانات تجوال
محظور ⛔


1️⃣ نهج أحادي
500 دقيقة
محلية
100 دقيقة
دولية
5GB بيانات
تجوال
منتهية
منتهية
منتهية
محظور ⛔

النهج 1: أرصدة أحادية

المفهوم: توفير كميات محددة لوجهات محددة. كل رصيد له مقدار ثابت (دقائق، جيجابايت، عدد الرسائل النصية القصيرة) مرتبط بوجهات محددة. عند نفادها، يتم حظر الاستخدام ما لم يكن هناك خيار مالي.

مثال: خطة محلية مع أرصدة متعددة

```
Action_Domestic_Plan = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_domestic-plan",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            {
                "Identifier": "*reset_account",
                "Weight": 700
            },
            # 500 دقيقة مكالمات محلية
            {
                "Identifier": "*topup_reset",
                "BalanceId": f"Domestic_Voice__{500 * 60 *
1000000000}",
                "BalanceType": "*voice",
                "DestinationIDs": "Dest_Domestic_All",  # محلية فقط
                "Units": 500 * 60 * 1000000000,  # 500 دقيقة بالتنبيهات
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 90
            },
            # 1000 رسالة نصية قصيرة محلية
            {
                "Identifier": "*topup_reset",
                "BalanceId": "Domestic_SMS__1000",
                "BalanceType": "*sms",
                "DestinationIDs": "Dest_Domestic_All",
                "Units": 1000,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 89
            },
            # 10GB بيانات محلية
            {
                "Identifier": "*topup_reset",
                "BalanceId": f"Domestic_Data__{10 * 1024 * 1024 *
1024}",
                "BalanceType": "*data",
                "DestinationIDs": "Dest_PLMN_OnNet",  # PLMN الخاص
بك
```

```
                "Units": 10 * 1024 * 1024 * 1024,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 88
            },
            {
                "Identifier": "*cdrlog",
                "BalanceType": "*generic",
                "ExtraParameters": "
{\"Category\":\"^activation\",\"Destination\":\"Domestic Plan\"}",
                "Weight": 80
            }
        ]
    }]
}

result = OCS_Obj.SendData(Action_Domestic_Plan)
assert result['error'] is None or result['error'] == "EXISTS"
```

كيف يعمل هذا:

- المكالمات المحلية (1-555-1234) → تستخدم رصيد 500 دقيقة
- المكالمات الدولية (20-44-xxx) → لا يوجد رصيد متاح، محظور أو يستخدم الرصيد المالي إذا كان متاحًا
- الرسائل النصية المحلية → تستخدم رصيد 1000 رسالة نصية قصيرة
- استخدام البيانات المنزلية → يستخدم رصيد 10 GB
- بيانات التجوال → لا يوجد رصيد متاح (ستحتاج إلى رصيد بيانات تجوال منفصل)

المزايا:

- تكاليف متوقعة للعملاء
- لا صدمة في الفاتورة
- حدود واضحة

العيوب:

- أقل مرونة - لا يمكن استخدام الخدمة خارج الوجهات المضمنة
- يتطلب أرصدة متعددة لحالات الاستخدام المختلفة
- قد يشعر العميل بالقيود

النهج 2: مالي (PAYG)

المفهوم: توفير رصيد مالي يتم خصمه بمعدلات محددة للوجهات. رصيد مالي واحد يستخدم لجميع أنواع الاستخدام. يبحث CGRateS عن المعدل لكل وجهة ويخصم التكلفة من الرصيد.

ملاحظة: يتطلب PAYG تعريف ملفات تعريف الأسعار لكل وجهة لتحديد المبلغ بالدولار لكل وحدة. انظر ملفات تعريف الأسعار للأرصدة المالية / PAYG للتكوين الكامل.

مثال: رصيد PAYG بقيمة 50 دولار

```
Action_PAYG_Credit = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_payg-50-credit",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            # رصيد مالي بقيمة 50 دولار
            {
                "Identifier": "*topup",
                "BalanceId": "PAYG_Monetary_Balance",
                "BalanceType": "*monetary",
                "DestinationIDs": "*any",  # يعمل مع أي وجهة
                "Units": 5000,  # 50.00 دولار (بالسنت)
                "ExpiryTime": "+2160h",  # 90 يومًا
                "BalanceWeight": 1000,  # أقل من الأرصدة الأحادية
                "Weight": 90
            },
            {
                "Identifier": "*cdrlog",
                "BalanceType": "*generic",
                "ExtraParameters": "
{\"Category\":\"^activation\",\"Destination\":\"$50 PAYG
Credit\"}",
                "Weight": 80
            }
        ]
    }]
}

result = OCS_Obj.SendData(Action_PAYG_Credit)
```

كيف يعمل PAYG:

السيناريو 1: مكالمة محلية، 10 دقائق

- المعدل: 0.10 دولار / دقيقة
- التكلفة: 10 × 0.10 = 1.00 دولار
- المتبقي: 49.00 دولار

السيناريو 2: مكالمة إلى المملكة المتحدة، 5 دقائق

- المعدل: 0.25 دولار / دقيقة + 0.05 دولار رسوم الاتصال
- التكلفة: (5 × 0.25) + 0.05 = 1.30 دولار
- المتبقي: 47.70 دولار

السيناريو 3: تجوال على Verizon، 100MB بيانات

- المعدل: 2.00 دولار / ميغابايت
- التكلفة: 100 × 2.00 = 200.00 دولار
- النتيجة: أموال غير كافية → تم حظر الجلسة عند حوالي 47 دولار (~23MB)

المزايا:

- رصيد واحد لكل شيء - مرونة كبيرة
- يمكن للعميل استخدام الخدمة في أي مكان يتم تحديد الأسعار فيه
- تكوين بسيط

العيوب:

- تكاليف غير متوقعة للعملاء
- خطر صدمة الفاتورة (خاصة أثناء التجوال)
- قد تكون مكلفة للمستخدمين الكثيفين

النهج 3: هجين (أفضل ما في الاثنين)

المفهوم: دمج الأرصدة الأحادية مع خيار مالي احتياطي. استخدم أرصدة أحادية ذات وزن عالٍ للاستخدام المضمن، مع رصيد مالي منخفض الوزن كفائض. أفضل تجربة للعملاء مع تكلفة أساسية متوقعة وفائض مرن.

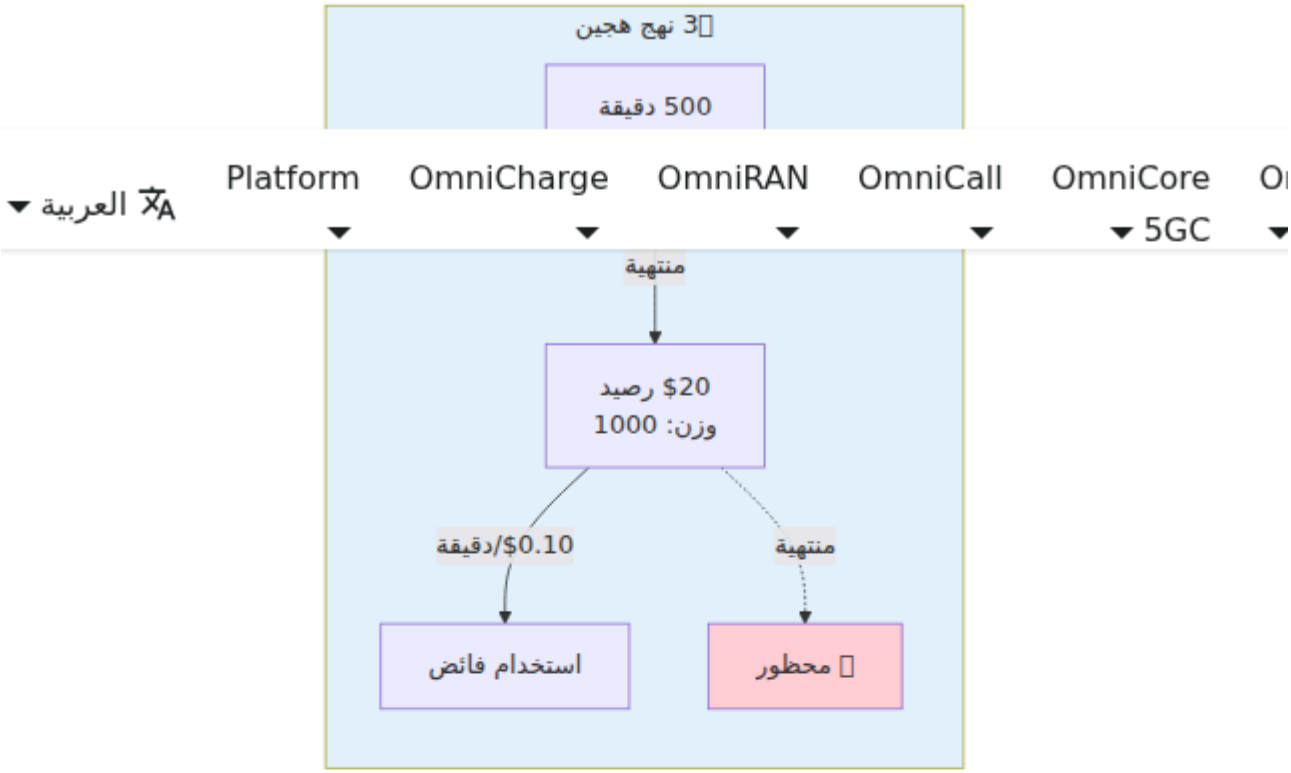
مثال: خطة مرنة هجينة

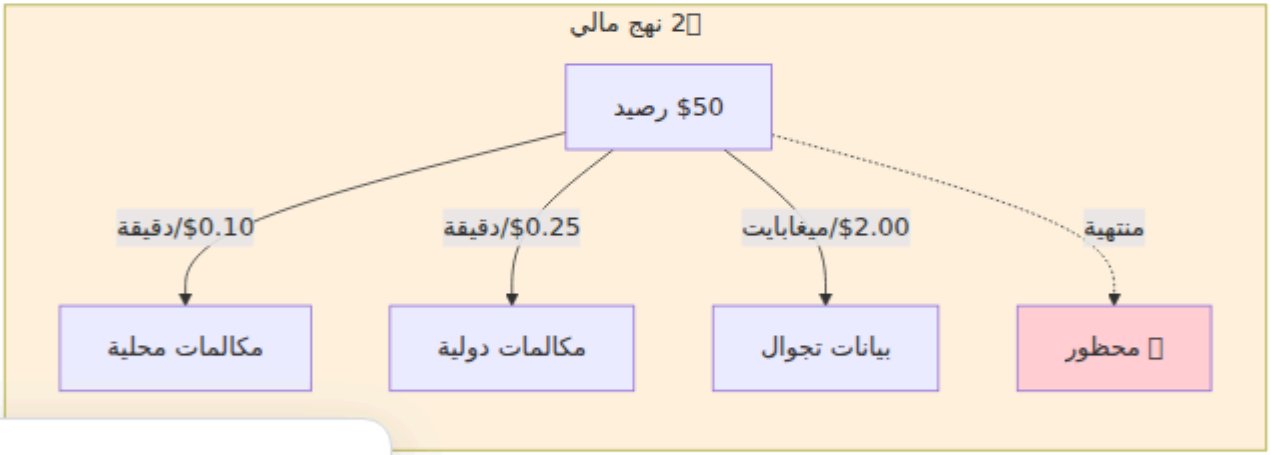
```
Action_Hybrid_Plan = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_hybrid-flex-plan",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            {
                "Identifier": "*reset_account",
                "Weight": 700
            },

            # ============= أرصدة أحادية (مضمنة) =============
            # 500 دقيقة مكالمات صوتية محلية
            {
                "Identifier": "*topup_reset",
                "BalanceId": f"Domestic_Voice__{500 * 60 *
1000000000}",
                "BalanceType": "*voice",
                "DestinationIDs": "Dest_Domestic_All",
                "Units": 500 * 60 * 1000000000,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,  # يتم استهلاكه أولاً للمكالمات
المحلية
                "Weight": 95
            },
            # 100 دقيقة مكالمات صوتية دولية
            {
                "Identifier": "*topup_reset",
                "BalanceId": f"International_Voice__{100 * 60 *
1000000000}",
                "BalanceType": "*voice",
                "DestinationIDs": "Dest_International_All",
                "Units": 100 * 60 * 1000000000,
                "ExpiryTime": "*month",
                "BalanceWeight": 1150,  # يتم استهلاكه أولاً دوليًا
                "Weight": 94
            },
            # 1000 رسالة نصية قصيرة محلية
            {
                "Identifier": "*topup_reset",
                "BalanceId": "Domestic_SMS__1000",
```

```
                "BalanceType": "*sms",
                "DestinationIDs": "Dest_Domestic_All",
                "Units": 1000,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 93
            },
            # 15GB بيانات محلية
            {
                "Identifier": "*topup_reset",
                "BalanceId": f"Domestic_Data__{15 * 1024 * 1024 *
1024}",
                "BalanceType": "*data",
                "DestinationIDs": "Dest_PLMN_OnNet",
                "Units": 15 * 1024 * 1024 * 1024,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 92
            },
            # 2GB (المنطقة 1) بيانات تجوال
            {
                "Identifier": "*topup_reset",
                "BalanceId": f"Roaming_Zone1_Data__{2 * 1024 *
1024 * 1024}",
                "BalanceType": "*data",
                "DestinationIDs": "Dest_PLMN_Zone_NorthAmerica",
                "Units": 2 * 1024 * 1024 * 1024,
                "ExpiryTime": "*month",
                "BalanceWeight": 1100,
                "Weight": 91
            },

            # ============= رصيد مالي (فائض / PAYG) =============
            # 20 دولار للفوائض
            {
                "Identifier": "*topup",
                "BalanceId": "PAYG_Overflow_Balance",
                "BalanceType": "*monetary",
                "DestinationIDs": "*any",
                "Units": 2000,  # 20.00 دولار
                "ExpiryTime": "*month",
                "BalanceWeight": 1000,  # يتم استهلاكه أخيرًا (احتياطي)
                "Weight": 90
            },
```

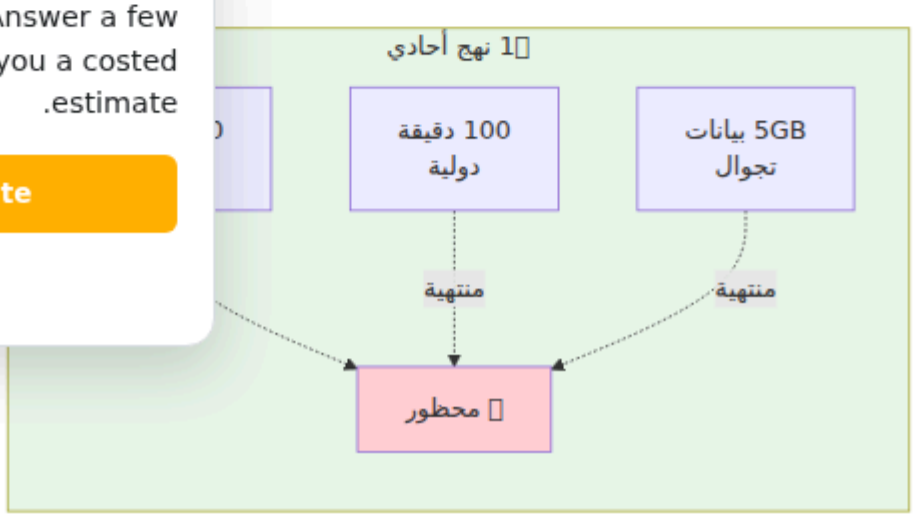
```
            {
                "Identifier": "*cdrlog",
                "BalanceType": "*generic",
                "ExtraParameters": "
{\"Category\":\"^activation\",\"Destination\":\"Hybrid Flex
Plan\"}",
                "Weight": 80
            }
        ]
    }]
}

result = OCS_Obj.SendData(Action_Hybrid_Plan)
```

تدفق استهلاك الرصيد:





have looked at 39 of our
k functions. Answer a few
nd we send you a costed
.estimate
et an estimate
No thanks



سيناريوهات المثال:

السيناريو 1: استخدام 600 دقيقة محلية

1. تستخدم → أول 500 دقيقة "Domestic_Voice_500min" (الوزن 1200) - مجانية
2. مستنفدة، يتم الانتقال إلى "Domestic_Voice_500min" → الـ 100 دقيقة التالية "PAYG_Overflow_Balance" (الوزن 1000)
3. تم الخصم: 100 دقيقة × 0.10 دولار = 10.00 دولار من الرصيد النقدي
4. المتبقي: 10.00 دولار نقدي

السيناريو 2: 150 دقيقة دولية (إلى المملكة المتحدة)

1. تستخدم → أول 100 دقيقة "International_Voice_100min" (الوزن 1150) - مجانية
2. يتم الانتقال إلى → الـ 50 دقيقة التالية "PAYG_Overflow_Balance"
3. تم الخصم: (50 × 0.25 دولار) + 0.05 دولار اتصال = 12.55 دولار
4. المتبقي: 7.45 دولار نقدي (إذا بدأ برصيد 20 دولار)

الإيجابيات:

- تكلفة أساسية متوقعة
- مرونة للتجاوزات العرضية
- أفضل تجربة للعميل
- لا يوجد قطع قاسي - الخدمة مستمرة

السلبيات:

- أكثر تعقيدًا في الإعداد
- أكثر تعقيدًا في الشرح للعملاء
- يتطلب إدارة دقيقة لوزن الرصيد

سيناريوهات الاستخدام في العالم الحقيقي

تظهر هذه السيناريوهات كيف تتجمع المفاهيم معًا في الممارسة العملية. كل منها يظهر التدفق الكامل من حدث الاستخدام إلى خصم الرصيد أو الحظر.

السيناريو 1: عميل في المنزل يتصل بالمملكة المتحدة

العميل لديه: خطة محلية (500 دقيقة محلية، 1000 رسالة نصية، 10 جيجابايت بيانات)

الإجراء: يتصل برقم المملكة المتحدة (0958-7946-20-44) لمدة 10 دقائق

التحقق من الرصيد:

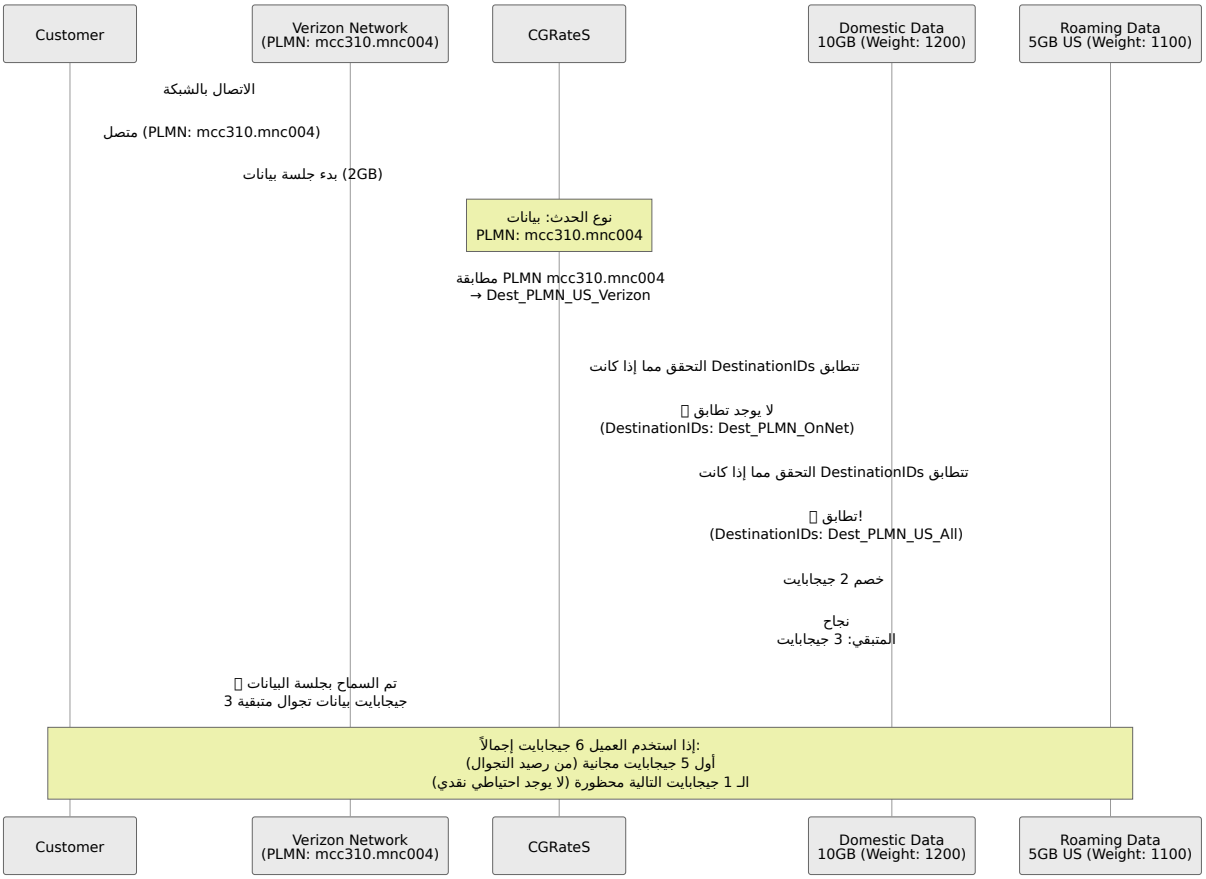
1. CGRateS يستقبل المكالمة إلى `44207946...`
2. يتطابق مع الوجهة: `Dest_International_UK`
3. يتحقق من الرصيد مع `DestinationIDs: "Dest_International_UK"`
4. لم يتم العثور على رصيد مطابق
5. يتحقق من الرصيد النقدي
6. لم يتم العثور على رصيد نقدي
7. **النتيجة:** المكالمة محظورة (رصيد غير كافٍ)

الحل: يحتاج العميل إلى خطة دولية أو رصيد PAYG لإجراء مكالمات إلى المملكة المتحدة

السيناريو 2: عميل يتجول في الولايات المتحدة تخدم البيانات ي��

العميل لديه: خطة محلية + إضافة تجوال 5 جيجابايت في الولايات المتحدة

الإجراء: يتجول على شبكة Verizon (PLMN mcc310.mnc004)، يستخدم 2 جيجابايت من البيانات



التحقق من الرصيد:

1. CGRateS يستقبل جلسة البيانات على PLMN `mcc310.mnc004`
2. يتطابق مع الوجهة: `Dest_PLMN_US_Verizon`
3. يتحقق من رصيد البيانات مع `DestinationIDs: "Dest_PLMN_US_All"` (مطابقة أوسع)
4. يجد: رصيد "Roaming_US_Data_5GB" (BalanceWeight: 1100)
5. يخصم 2 جيجابايت
6. **المتبقي:** 3 جيجابايت بيانات تجوال

ماذا لو استخدموا 6 جيجابايت؟

1. أول 5 جيجابايت → تستخدم رصيد التجوال (مستنفد)
2. الـ 1 جيجابايت التالية → يتحقق من الرصيد النقدي
3. لا يوجد رصيد نقدي → **تم حظر الجلسة** عند 5 جيجابايت

السيناريو 3: عميل بخطة هجينة يتجاوز دقائق محلية

العميل لديه: خطة هجينة (500 دقيقة محلية + 20 دولار احتياطي)

الإجراء: يجري 600 دقيقة من المكالمات المحلية

التحقق من الرصيد:

1. أول 500 دقيقة:
   - تستخدم "Domestic_Voice_500min" (BalanceWeight: 1200)
   - لا يوجد رسوم
2. "Domestic_Voice_500min" مستنفدة
3. الـ 100 دقيقة التالية:
   - يتم الانتقال إلى "PAYG_Overflow_Balance" (BalanceWeight: 1000)
   - السعر: 0.10 دولار/دقيقة
   - الرسوم: 100 × 0.10 دولار = 10.00 دولار
   - تم خصمها من رصيد 20 دولار النقدي
   - **المتبقي:** 10.00 دولار

الفوترة:

- يرى العميل: 500 دقيقة مشمولة + 100 دقيقة تجاوز ($10.00)

السيناريو 4: مستخدم يتجول بشكل غير متوقع PAYG

العميل لديه: رصيد PAYG بقيمة 50 دولار

الإجراء: يسافر إلى الولايات المتحدة، يتجول على Verizon، يستخدم 100 ميجابايت من البيانات (دون علمه)

حساب ا��رسوم:

1. Verizon ميجابايت على 100
2. السعر: 2.00 دولار/ميجابايت (من أسعار PAYG)
3. إجمالي الرسوم: 100 × 2.00 دولار = 200.00 دولار
4. المتاح: 50.00 دولار
5. **النتيجة:** العميل يستخدم ~25 ميجابايت قبل نفاد الرصيد، تم حظر الجلسة
6. **صدمة الفاتورة!** العميل استهلك بشكل غير متوقع كامل الـ 50 دولار

نهج أفضل: يوصى بإضافة تجوال لتجنب صدمة الفاتورة

أفضل الممارسات

1. استخدم معرفات رصيد وصفية

معرفات الرصيد الجيدة توثق نفسها:

```
# جيد
"Data_5GB_5days_{{ uuid }}"
"Voice_100min_Monthly"
"Bonus_Data_Loyalty"

# سيء
"balance1"
"data"
"temp"
```

2. وثق استراتيجية الوزن الخاصة بك

حدد نظام وزن متسق عبر جميع المنتجات:

```
# نظام الوزن:
# 30 = أرصدة مميزة/ترويجية
# 20 = أرصدة مكافآت/ولاء
# 10 = أرصدة عادية/مشتراة
# 5 = أرصدة احتياطية/احتياطية
```

3. تضمين تسجيل CDR

قم دائمًا بتسجيل إضافات الرصيد لأغراض التدقيق:

```
{
  "Identifier": "*cdrlog",
  "BalanceType": "*generic",
  "ExtraParameters": "
{\"Category\":\"^activation\",\"Destination\":\"{{ package_name
}}\"}"
}
```

4. استخدم ActionPlans للعمليات المتكررة

لإعادة تعيين شهرية، لا تقم بتشغيل الإجراء يدويًا. استخدم ActionPlans:

```
# إنشاء ActionPlan يعمل شهريًا
- name: Create monthly ActionPlan
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.SetActionPlan",
        "params": [{
          "Id": "ActionPlan_Monthly_Reset",
          "ActionPlan": [{
            "ActionsId": "Action_Monthly_Reset",
            "Time": "*monthly",
            "Weight": 10
          }],
          "ReloadScheduler": true
        }]
      }
```

5. إعادة تعيين ActionTriggers بعد التعبئة

بعد تنفيذ إجراء، قم بإعادة تعيين المحفزات لمنع الإشعارات المكررة:

```
- name: Reset ActionTriggers
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "APIerSv1.ResetAccountActionTriggers",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "Executed": false
        }]
      }
```

6. تعيين أوزان رصيد مناسبة

استخدم استراتيجية وزن متسقة عبر جميع المنتجات:

```
# أولوية الاستهلاك (من الأعلى إلى الأدنى)
BalanceWeight: 1200  # محلي مشمول (يستخدم أولاً في المنزل)
BalanceWeight: 1150  # دولي مشمول (يستخدم أولاً للمكالمات الدولية)
BalanceWeight: 1100  # تجوال مشمول (يستخدم أولاً عند التجوال)
BalanceWeight: 1000  # احتياطي نقدي (يستخدم أخيرًا)
```

هذا يضمن ترتيب استهلاك متوقع: الأرصدة المحلية أولاً، ثم الدولية، ثم التجوال، وأخيرًا النقدية كاحتياطي.

7. التحكم في استخدام الرصيد باستخدام DestinationIDs

استخدم دائمًا DestinationIDs محددة لمنع الاستخدام غير المقصود:

```
# جيد - تحكم وجهة صريح
{
    "BalanceType": "*voice",
    "DestinationIDs": "Dest_International_UK",  # فقط المملكة المتحدة
    "Units": 200 * 60 * 1000000000
}

# سيء - استخدام غير مقصود
{
    "BalanceType": "*voice",
    "DestinationIDs": "*any",  # يمكن استخدامه لأرقام مميزة!
    "Units": 200 * 60 * 1000000000
}
```

8. تجميع الوجهات منطقيًا

أنشئ مجموعات وجهات تتماشى مع عرض منتجك:

```
# جيد - تجميع منطقي
"Dest_International_Europe" = مجموعة) جميع دول الاتحاد الأوروبي
(مشتركة من 500 دقيقة

# سيء - تفصيل مفرط
"Dest_France", "Dest_Germany", "Dest_Italy" (مجموعات صغيرة منفصلة)
```

انظر وجهات CGRateS لأمثلة تجميع الوجهات.

9. تضمين احتياطي نقدي في الخطط

لأفضل تجربة للعميل، قم بتضمين رصيد نقدي صغير كاحتياطي:

```
# موصى به: دائمًا تضمين رصيد نقدي صغير
{
    "BalanceType": "*monetary",
    "Units": 1000,  # حماية من تجاوز 10 دولارات
    "BalanceWeight": 1000  # أقل من الأرصدة الوحدوية
}
```

هذا يمنع قطع الخدمة القاسي ويوفر مرونة للتجاوزات العرضية.

10. استخدم زيادات الأسعار بشكل استراتيجي

اختر زيادات الأسعار المناسبة لأنواع الخدمات المختلفة:

```
# محلي: صديقة للعميل لكل ثانية
"RateIncrement": "1s"

# دولي: كتل قياسية مدتها 6 ثوانٍ
"RateIncrement": "6s"

# مميز: كتل مدتها 30 ثانية لتشجيع الاستخدام العادل
"RateIncrement": "30s"

# بيانات: لكل كيلوبايت من أجل الدقة
"RateIncrement": "1024"
```

11. تعيين أوقات انتهاء معقولة

تطابق أوقات الانتهاء مع أنواع المنتجات:

```
# خطط الاشتراك الشهرية
"ExpiryTime": "*month"

# تعبئة لمرة واحدة/إضافات
"ExpiryTime": "+720h"  # 30 يومًا

# رصيد PAYG (صلاحية أطول)
"ExpiryTime": "+2160h"  # 90 يومًا

# إضافات التجوال (مبنية على الرحلة)
"ExpiryTime": "+360h"  # 15 يومًا
```

استكشاف الأخطاء وإصلاحها

المشكلة: الرصيد لم يضاف

الأعراض: يتم تنفيذ الإجراء بنجاح لكن الرصيد لا يظهر

الأسباب المحتملة:

- UUID الحساب خاطئ
- عدم تطابق نوع الرصيد
- انتهاء الصلاحية بالفعل

الحل: تحقق من وجود الحساب وتحقق من وقت انتهاء صلاحية الرصيد

المشكلة: استهلاك رصيد خاطئ

الأعراض: النظام يستهلك من رصيد غير متوقع

الأسباب المحتملة:

- تكوين الوزن غير صحيح
- أرصدة متعددة بنفس المعرف

الحل: مراجعة قيم الوزن والتأكد من أن معرفات الرصيد فريدة إذا كانت الاستقلالية مطلوبة

المشكلة: عدم عمل التمرير

الأعراض: استخدام `*topup` لكن الرصيد القديم لا يتجدد

الأسباب المحتملة:

- استخدام معرفات رصيد مختلفة (يحتاج إلى نفس المعرف)
- الإجراء يستخدم `*topup_reset` بدلاً من `*topup`

الحل: تحقق من تناسق معرف الرصيد ونوع الإجراء

المشكلة: انتهاء صلاحية الرصيد على الفور

الأعراض: تم إضافة الرصيد لكنه يظهر كمنتهي الصلاحية

الأسباب المحتملة:

- وقت انتهاء الصلاحية في الماضي
- استخدام طابع زمني مطلق بدلاً من النسبي

الحل: استخدم انتهاء الصلاحية النسبي (`+Xh`) بدلاً من الطوابع الزمن��ة المطلقة

المشكلة: عدم استهلاك الرصيد

الأعراض: العميل لديه رصيد لكنه لا يزال محظورًا أو يتم فرض رسوم PAYG

الأسباب المحتملة:

1. عدم تطابق DestinationIDs - وجهات الرصيد لا تتطابق مع وجهة الاستخدام
2. انتهاء صلاحية الرصيد
3. نوع الرصيد خاطئ للاستخدام

خطوات تصحيح الأخطاء:

```
# تحقق من أرصدة الحساب
OCS_Obj.SendData({
    'method': 'ApierV2.GetAccount',
    'params': [{"Tenant": tenant, "Account": "service_uuid"}]
})

# تحقق مما يتم إرجاعه:
# - هل الرصيد موجود؟
# - هل DestinationIDs صحيحة؟
# - هل وقت انتهاء الصلاحية في المستقبل؟
# - هل الوحدات > 0؟
```

الحل: تحقق من تطابق الوجهة وأن الرصيد نشط. انظر السيناريو 1: عميل في المنزل يتصل بالمملكة المتحدة للمثال.

المشكلة: تطبيق سعر خاطئ

الأعراض: تم فرض رسوم خاطئة على العميل لاستخدام PAYG

الأسباب المحتملة:

1. تداخل الوجهات (الأولوية الخاطئة في خطة التسعير)
2. أوزان ربط خطة التسعير غير صحيحة
3. تعريف السعر يحتوي على قيم خاطئة

الحل:

- تأكد من أن الوجهات المحددة لها وزن أعلى في روابط خطة التسعير
- تحقق من أن أطول تطابق بادئة يعمل بشكل صحيح
- تحقق من أن قيم الأسعار والزيادات صحيحة

انظر ملفات تعريف الأسعار لأرصدة PAYG/نقدية للتكوين الصحيح.

المشكلة: عدم استخدام رصيد التجوال

الأعراض: العميل يتجول، لديه رصيد تجوال، لكن تم فرض رسوم أو تم حظره PAYG

الأسباب المحتملة:

1. عدم تطابق وجهة PLMN - الرصيد لا يتضمن PLMN الذي تمت زيارته
2. العميل متصل بشبكة خاطئة/غير متوقعة
3. DestinationIDs لا تتطابق مع PLMN الفعلي

الحل:

```
# تحقق من أي PLMN يتواجد العميل عليه (من CDRs أو أحداث الاستخدام)
# تحقق من أن الرصيد يتضمن ذلك PLMN:
"DestinationIDs": "Dest_PLMN_US_All"  # يجب أن يتضمن mcc310.mnc004
```

انظر وجهات CGRateS لتكوين وجهات PLMN والسيناريو 2: عميل يتجول في الولايات المتحدة يستخدم البيانات للمثال العملي.

المشكلة: خطة هجينة لا تتراجع إلى النقدي

الأعراض: الرصيد الوحدوي مستنفد لكن الرصيد النقدي لا يتم استخدامه

الأسباب المحتملة:

1. رصيد النقدي لديه قيود على DestinationIDs (يجب أن يكون `*any`)
2. لا توجد أسعار PAYG محددة لتلك الوجهة
3. مشكلة وزن الرصيد - وزن الرصيد النقدي ليس أقل من الوحدوي

الحل:

- تأكد من أن الرصيد النقدي لديه `DestinationIDs: "*any"`
- تحقق من تعريف أسعار PAYG لجميع الوجهات التي قد يستخدمها العميل
- تحقق من أوزان الرصيد: يجب أن يكون النقدي هو الأدنى (مثل 1000)

انظر النهج 3: هجينة للتكوين الصحيح للهجينة.

المشكلة: رسوم دولية غير متوقعة

الأعراض: العميل اتصل برقم محلي، وتم فرض رسوم دولية

الأسباب المحتملة:

1. الرقم في الواقع دولي (على سبيل المثال، رقم كندي +1 يتم اعتباره دوليًا)
2. تداخل بادئة في تعريفات الوجهة
3. العميل قام بالاتصال بشكل خاطئ (أضاف أرقامًا إضافية)

الحل:

- تحقق من CDRs للرقم الذي تم الاتصال به فعليًا
- تحقق من أن تعريفات بادئة الوجهة لا تتداخل بشكل غير صحيح
- اعتبر وجهة كندا منفصلة إذا كانت دول NANP بحاجة إلى أسعار مختلفة

انظر وجهات CGRateS لتكوين الوجهات الجغرافية.

الوثائق ذات الصلة

المفاهيم الأساسية

- تعريف المنتجات - سير العمل الكامل لإنشاء المنتجات، وتحديد إجراءات CGRateS، وربطها معًا
- وجهات CGRateS - كيفية تعريف الوجهات الجغرافية وPLMN، لخدمات الصوت، والرسائل النصية، والبيانات

التنفيذ والعمليات

- الرسوم والمدفوعات من دفاتر التشغيل - تدفق الدفع ذو المرحلتين، حسابات النسبة المئوية، وفرض الرسوم على العملاء
- دليل دفاتر التشغيل Ansible - هيكل دفتر التشغيل، تدفقات التوفير، وإلغاء التوفير مع كتل الإنقاذ
- نظرة عامة على نظام التوفير OmniCRM - كيفية عمل التوفير في

CGRateS تكوين وجهات

تشرح هذه الوثيقة كيفية تكوين الوجهات في CGRateS لـ OmniCRM. تحدد الوجهات أين يمكن استخدام الأرصدة، سواء كان ذلك للاتصال بأرقام محددة (جغرافية) أو استخدام البيانات/الصوت على شبكات محددة (PLMN).

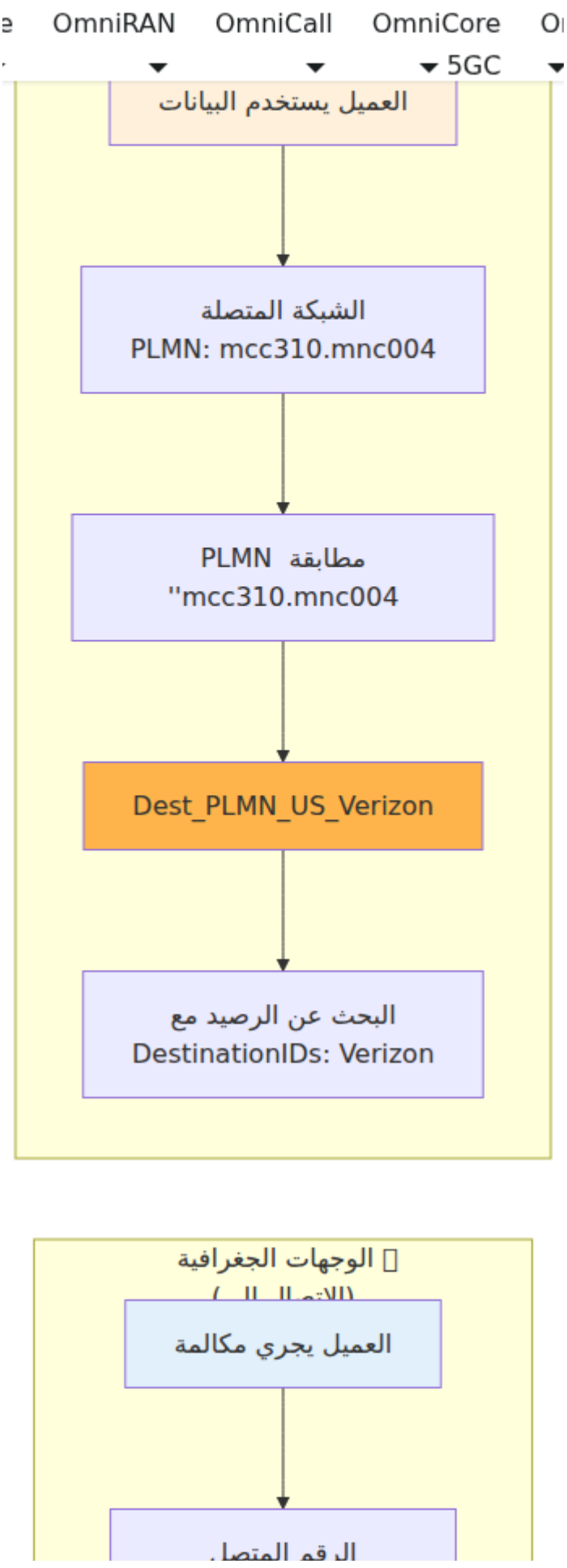
نظرة عامة

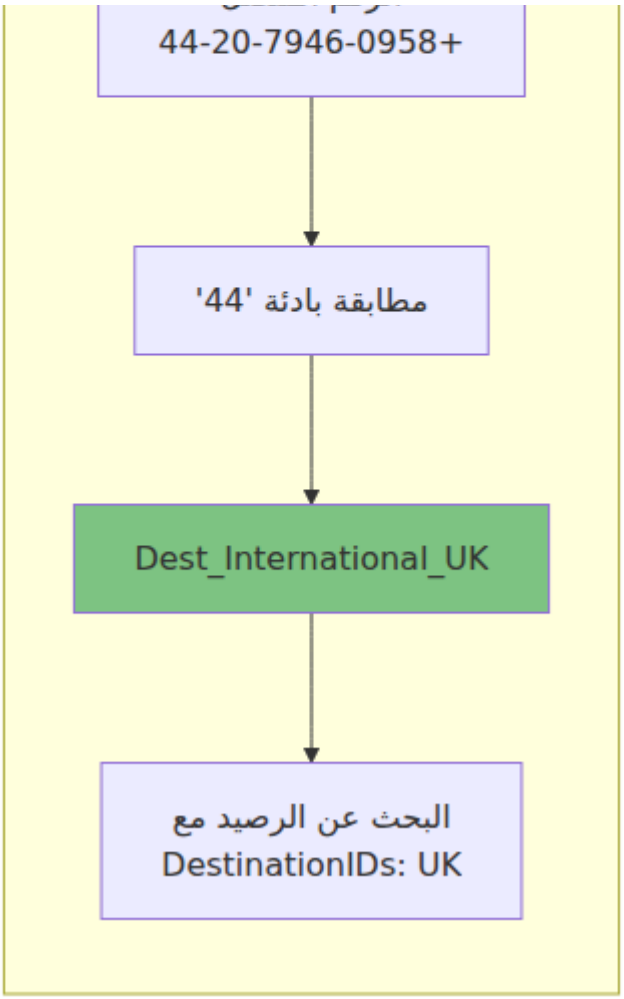
تستخدم CGRateS للتحكم في المكان الذي يمكن للعملاء استخدام أرصدتهم فيه **الوجهات**:

- **وجهات جغرافية**/SMS - بادئات الأرقام للمكالمات (مثل، أرقام إلى مواقع محددة المملكة المتحدة، أرقام الولايات المتحدة المجانية)
- **وجهات PLMN** - رموز الشبكة لاستخدام البيانات والتجوال من شبكات محددة (مثل، Verizon، Vodafone UK)

عندما يستخدم العملاء الخدمات، تطابق CGRateS:

- **الصوت/SMS**: بادئة الرقم المتصل ← وجهة جغرافية
- **البيانات**: رمز شبكة PLMN ← وجهة PLMN


e
OmniRAN
OmniCall
OmniCore
O
5GC
العميل يستخدم البيانات
الشبكة المتصلة
PLMN: mcc310.mnc004
PLMN مطابقة
''mcc310.mnc004
Dest_PLMN_US_Verizon
البحث عن الرصيد مع
DestinationIDs: Verizon
الوجهات الجغرافية
العميل يجري مكالمة



القاعدة الحرجة

- استخدم الوجهات الجغرافية (الرقم الذي يتم الاتصال به) → **أرصدة الصوت/SMS**
- استخدم وجهات (الشبكة التي يتصل بها العميل) PLMN → **أرصدة البيانات**

المتطلبات المسبقة

```
import cgrateshttpapi
import time

OCS_Obj = cgrateshttpapi.CGRateS("ocs.example.com", "2080")
tenant = "your_tenant_name"
tpid = str(tenant) + "_" + str(int(time.time()))
```

الجزء 1: الوجهات الجغرافية (الاتصال إلى الأماكن)

تحدد الوجهات الجغرافية بادئات الأرقام للمكالمات الصوتية و SMSإلى مواقع محددة.

الوجهات المحلية

```
#
# ============================================================================
# الوجهات المحلية (الولايات المتحدة/كندا - NANP)
#
# ============================================================================

# الهواتف المحمولة والثابتة في الولايات المتحدة/كندا (تشارك رموز المناطق)
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_Domestic_All",
        "Prefixes": ["1"]  # NANP (الولايات المتحدة/كندا)
    }]
})

# الأرقام المجانية في الولايات المتحدة
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_Domestic_TollFree",
        "Prefixes": ["1800", "1888", "1877", "1866", "1855",
"1844", "1833"]
    }]
})

# الأرقام ذات التكلفة العالية في الولايات المتحدة
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_Domestic_Premium",
        "Prefixes": ["1900"]
    }]
})
```

الوجهات الدولية

```
#
# ============================================================================
# الوجهات الدولية (حسب الدولة)
#
# ============================================================================

# المملكة المتحدة
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_International_UK",
        "Prefixes": ["44"]
    }]
})

# الصين
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_International_China",
        "Prefixes": ["86"]
    }]
})

# أستراليا - الهواتف المحمولة
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_AU_Mobile",
        "Prefixes": ["614"]  # الهواتف المحمولة الأسترالية (04xx يتم الاتصال بها كـ 614xx)
    }]
})

# أستراليا - الخط الثابت
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
```

```
        "TPid": tpid,
        "ID": "Dest_AU_Fixed",
        "Prefixes": [
            "612",  # NSW (سيدني)
            "613",  # VIC (ملبورن)
            "617",  # TAS (هوبارت)
            "618"   # SA, WA, NT (أديلايد، بيرث، داروين)
        ]
    }]
})

# أستراليا - الأرقام المجانية
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_AU_TollFree",
        "Prefixes": [
            "611800",  # مجانية
            "611300"   # سعر محلي
        ]
    }]
})

# أستراليا - الأرقام ذات التكلفة العالية
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_AU_Premium",
        "Prefixes": ["6119"]  # خدمات ذات تكلفة عالية
    }]
})

# أوروبا (مجمعة)
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_International_Europe",
        "Prefixes": [
            "33",  # فرنسا
            "49",  # ألمانيا
            "39",  # إيطاليا
```

```
            "34",  # إسبانيا
            "31",  # هولندا
            "32",  # بلجيكا
            "41"   # سويسرا
        ]
    }]
})

# دولية - جميع (شاملة، تستثني المحلية)
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_International_All",
        "Prefixes": ["2", "3", "4", "5", "6", "7", "8", "9"]  # جميع ما عدا "1" (NANP)
    }]
})
```

حالات الاستخدام:

- "**خطط محلية**" → `Dest_Domestic_All`: مكالمات غير محدودة إلى أرقام الولايات المتحدة/كندا
- "**إضافة مكالمات إلى المملكة المتحدة**" → `Dest_International_UK`: 100 دقيقة إلى المملكة المتحدة
- "**خطة أسترالية**" → `Dest_AU_Mobile`, `Dest_AU_Fixed`, `Dest_AU_TollFree`: 3000 دقيقة إلى الهواتف المحمولة والخطوط الثابتة الأسترالية
- "**حزمة دولية**" → `Dest_International_All`: 50 دقيقة إلى أي رقم دولي

الجزء 2: وجهات PLMN (الاستخدام من الأماكن - التجوال / استخدام البيانات)

رموز الشبكة لاستخدام البيانات وسيناريوهات التجوال. استخدم التنسيق PLMN تحدد وجهات `mccXXX.mncYYY` حيث:

- **MCC** = رمز الدولة المحمولة (3 أرقام)
- **MNC** = رمز الشبكة المحمولة (2-3 أرقام)

قواعد تنسيق PLMN

- **شبكة محددة**: `"mcc310.mnc004"` → فقط شبكة Verizon 004
- **جميع الشبكات في الدولة**: `"mcc310"` → أي شبكة أمريكية
- **أمثلة**: `mcc310.mnc004`, `mcc234.mnc015`, `mcc505.mnc057`

على الشبكة (شبكتك المنزلية)

مهم: حدد شبكتك المحددة حيث تم تزويد بطاقات SIM الخاصة بك.

```
# على الشبكة (شبكتك المنزلية)
# هذه هي PLMN لمشغل الشبكة الخاص بك - الشبكة التي تكون فيها بطاقات SIM الخاصة بعملائك نشطة
# يجب أن تستخدم جميع أرصدة البيانات المحلية على الشبكة هذه الوجهة

OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_OnNet",
        "Prefixes": ["mcc505.mnc057"]  # مثال: MCC 505، MNC 57 إذا كانت هذه هي شبكتك المنزلية
    }]
})
# استبدل mcc505.mnc057 برمز PLMN الفعلي لمشغل الشبكة الخاص بك
# يجب أن تستخدم جميع أرصدة البيانات المحلية على الشبكة البادئات المحددة هنا
# خلاف ذلك، سيتم اعتبارها تجوال
```

للتجوال في الولايات المتحدة PLMN وجهات

```
#
# ============================================================================
# وجهات PLMN للتجوال في الولايات المتحدة
#
# ============================================================================

# Verizon Wireless
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_US_Verizon",
        "Prefixes": [
            "mcc310.mnc004", "mcc310.mnc010", "mcc310.mnc012",
"mcc310.mnc013",
            "mcc311.mnc480", "mcc311.mnc481", "mcc311.mnc482",
"mcc311.mnc483"
        ]
    }]
})

# جميع PLMNs (شاملة) الأمريكية
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_US_All",
        "Prefixes": ["mcc310", "mcc311", "mcc312", "mcc313",
"mcc316"]
    }]
})
# ملاحظة: هذا يتطابق مع أي شبكة بهذه MCCs (310، 311، إلخ.)
# العميل على mcc310.mnc004 (Verizon)، mcc310.mnc410 (AT&T)،
mcc311.mnc580 (US Cellular)
# سيتطابق جميعهم مع هذه الوجهة
```

للتجوال في المملكة المتحدة PLMN وجهات

```
#
# ==========================================================================
# للتجوال في المملكة المتحدة PLMN وجهات
#
# ==========================================================================

# Vodafone UK
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_UK_Vodafone",
        "Prefixes": ["mcc234.mnc015"]
    }]
})

# EE (كل شيء في كل مكان)
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_UK_EE",
        "Prefixes": ["mcc234.mnc030", "mcc234.mnc033"]
    }]
})

# جميع PLMNs في المملكة المتحدة
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_UK_All",
        "Prefixes": ["mcc234"]
    }]
})
```

مناطق التجوال (مجموعات متعددة البلدان)

رائع لحزم التجوال الإقليمية مثل "التجوال في أوروبا" ��و "التجوال في منطقة آسيا والمحيط الهادئ".

```
#
======================================================================
# مناطق التجوال (مجموعات متعددة البلدان)
#
======================================================================

# المنطقة 1: أمريكا الشمالية
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_Zone_NorthAmerica",
        "Prefixes": [
            "mcc310", "mcc311", "mcc312", "mcc313", "mcc316",  #
الولايات المتحدة
            "mcc302", "mcc334"  # كندا
        ]
    }]
})
# فقط يوفر تغطية على أي شبكة في هذه البلدان MCC استخدام بادئات
# رائع لحزم "التجوال في أمريكا الشمالية" التي تعمل على جميع شركات الاتصالات

# المنطقة 2: أوروبا
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_Zone_Europe",
        "Prefixes": [
            "mcc234",  # المملكة المتحدة
            "mcc208",  # فرنسا
            "mcc262",  # ألمانيا
            "mcc222",  # إيطاليا
            "mcc214",  # إسبانيا
            "mcc228"   # سويسرا
        ]
    }]
})

# المنطقة 3: منطقة آسيا والمحيط الهادئ
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
```

```
        "TPid": tpid,
        "ID": "Dest_PLMN_Zone_AsiaPacific",
        "Prefixes": [
            "mcc505",  # أستراليا
            "mcc460",  # الصين
            "mcc454",  # هونغ كونغ
            "mcc440",  # اليابان
            "mcc520",  # تايلاند
            "mcc525",  # سنغافورة
            "mcc530"   # نيوزيلندا
        ]
    }]
})

# شاملة لأي تجوال (استخدم بحذر)
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_Roaming_All",
        "Prefixes": ["mcc"]  # يتطابق مع أي PLMN (واسع جداً)
    }]
})
```

حالات الاستخدام:

- 5" :**التجوال المحدد بالشبكة**GB على Verizon فقط" → `Dest_PLMN_US_Verizon`
- 10" :**التجوال على مستوى الدولة**GB (أي شبكة) في الولايات المتحدة" → `Dest_PLMN_US_All`
- 15" :**التجوال الإقليمي**GB تجوال في أوروبا" → `Dest_PLMN_Zone_Europe`
- 1" :**التجوال العالمي**GB في أي مكان" → `Dest_Roaming_All`

متى تستخدم كل نهج

أنواع الأرصدة	مثال	حالة الاستخدام	نوع الوجهة
صوت، SMS	"100 دقيقة إلى أرقام المملكة المتحدة"	الصوت/SMS إلى دول محددة	**جغرافية**
بيانات	"100GB بيانات على الشبكة"	البيانات المحلية على شبكتك	**PLMN على الشبكة**
بيانات (وصوت/SMS أثناء التجوال)	"5GB على Verizon فقط"	التجوال المتميز على الشبكات المفضلة	**PLMN محدد**
بيانات (وصوت/SMS أثناء التجوال)	"5GB في الولايات المتحدة (أي شبكة)"	التجوال في أي شبكة في الدولة	**PLMN على مستوى الدولة**
بيانات (وصوت/SMS أثناء التجوال)	"10GB تجوال في أوروبا"	حزم التجوال الإقليمية	**PLMN المنطقة**

تحديثات الوجهات الديناميكية والوصول إلى الخدمة

كيف تعمل تحديثات الوجهات

المفهوم الحرج: يتم **الإشارة إلى الوجهات في CGRateS بواسطة ID**، وليس نسخها. **وتحميل خطة التعريفة في وقت** PLMN عندما تقوم بتحديث بادئات الوجهة أو رموز **التشغيل، تتأثر جميع الأرصدة الحالية التي تشير إلى تلك الوجهة على الفور**.

هذا يعني:

- ✅ **لا حاجة لتحديث أرصدة العملاء الفردية** عند توسيع أو تقييد تغطية الخدمة
- ✅ **تغييرات على مستوى الشبكة** لجميع العملاء الذين يستخدمون تلك الوجهة بمجرد تحميل خطة التعريفة

- ⚠ يمكنك - **تدخل التغييرات حيز التنفيذ عندما تقوم بتحميل خطة التعريفة** تعديل الوجهات عدة مرات قبل التحميل، وتصبح التغييرات نشطة فقط بعد `LoadTariffPlanFromStorDb`

المثال 1: توسيع التغطية

لديك وجهة للتجوال الأوروبي:

```
# التكوين الأولي - 3 دول فقط
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_Zone_Europe",
        "Prefixes": [
            "mcc262",  # ألمانيا
            "mcc208",  # فرنسا
            "mcc222"   # إيطاليا
        ]
    }]
})
```

يمكنهم التجوال في `Dest_PLMN_Zone_Europe` **العملاء الذين لديهم أرصدة** تشير إلى ألمانيا وفرنسا وإيطاليا.

لاحقًا، تضيف إسبانيا والمملكة المتحدة:

```
# التكوين المحدث - الآن 5 دول
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_Zone_Europe",
        "Prefixes": [
            "mcc262",  # ألمانيا
            "mcc208",  # فرنسا
            "mcc222",  # إيطاليا
            "mcc214",  # (جديد) إسبانيا
            "mcc234"   # (جديد) المملكة المتحدة
        ]
    }]
})

# تحميل خطة التعريفة لجعل التغييرات نشطة
OCS_Obj.SendData({
    'method': 'ApierV1.LoadTariffPlanFromStorDb',
    'params': [{
        "TPid": tpid,
        "DryRun": False,
        "Validate": True
    }]
})

# إعادة تحميل ذاكرة التخزين المؤقت للوجهة
OCS_Obj.SendData({
    'method': 'CacheSv1.ReloadCache',
    'params': [{
        "DestinationIDs": ["Dest_PLMN_Zone_Europe"]
    }]
})
```

النتيجة: بعد تحميل خطة التعريفة، يحصل جميع العملاء الحاليين الذين لديهم أرصدة لـ `Dest_PLMN_Zone_Europe` على إمكانية التجوال في إسبانيا والمملكة المتحدة - دون الحاجة إلى تحديثات أرصدة فردية.

المثال 2: تقييد التغطية

إزالة دولة من وجهة تمنع الوصول بمجرد تحميل خطة التعريفة:

```
# إزالة ألمانيا من منطقة أوروبا
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_Zone_Europe",
        "Prefixes": [
            "mcc208",  # فرنسا
            "mcc222",  # إيطاليا
            "mcc214",  # إسبانيا
            "mcc234"   # المملكة المتحدة
            # تمت إزالتها (mcc262) ألمانيا
        ]
    }]
})

# تحميل خطة التعريفة لجعل التغييرات نشطة
OCS_Obj.SendData({
    'method': 'ApierV1.LoadTariffPlanFromStorDb',
    'params': [{
        "TPid": tpid,
        "DryRun": False,
        "Validate": True
    }]
})

# إعادة تحميل الذاكرة المؤقتة
OCS_Obj.SendData({
    'method': 'CacheSv1.ReloadCache',
    'params': [{
        "DestinationIDs": ["Dest_PLMN_Zone_Europe"]
    }]
})
```

النتيجة: بعد تحميل خطة التعريفة، سيتم **منع** العملاء الذين يحاولون التجوال في ألمانيا، حتى لو كانت لديهم أرصدة متبقية لـ `Dest_PLMN_Zone_Europe`. لا يزال الرصيد موجودًا ولكنه لم يعد يتطابق مع وجهة PLMN.

المثال 3: الوجهات الجغرافية للصوت/SMS

ينطبق نفس المبدأ على الوجهات الجغرافية للصوت و SMS:

```
# إضافة كندا إلى "المكالمات المحلية"
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_Domestic_All",
        "Prefixes": [
            "1",  # يشمل بالفعل الولايات المتحدة (وكندا سابقًا)
            # لا حاجة لتغيير - بادئة "1" تغطي كلاهما
        ]
    }]
})

# أو إنشاء وجهة منفصلة لأمريكا الشمالية
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_North_America",
        "Prefixes": [
            "1"  # الولايات المتحدة + كندا
        ]
    }]
})
```

تعمل الآن لكل من `DestinationIDs: "Dest_Domestic_All"` **التأثير:** أي أرصدة مع المكالمات إلى الولايات المتحدة وكندا (إذا كانت تستخدم بادئة "1").

قواعد الوصول إلى الخدمة

حرج: إذا حاول العميل استخدام خدمة (بيانات/SMS/صوت) سيتم **رفض** **ولم تتطابق أي وجهة**، **الخدمة**، حتى لو كانت لديهم أرصدة.

السيناريو 1: لا توجد وجهة مطابقة

```
# لدى العميل رصيد:
{
    "BalanceId": "Europe_Data_5GB",
    "BalanceType": "*data",
    "DestinationIDs": "Dest_PLMN_Zone_Europe",  # فقط فرنسا، ألمانيا، إيطاليا
    "Units": 5 * 1024 * 1024 * 1024
}

# يتجول العميل على شبكة في إسبانيا (mcc214.mnc001)
# إسبانيا ليست في Dest_PLMN_Zone_Europe
```

النتيجة:

- ❌ جلسة البيانات **محجوبة** (لا توجد وجهة مطابقة)
- يبقى الرصيد غير مستخدم
- يتلقى العميل "رصيد غير كافٍ" أو خطأ مشابه

الإصلاح: أضف إسبانيا إلى `Dest_PLMN_Zone_Europe` أو تأكد من أن العميل لديه رصيد نقدي كخيار احتياطي. مع `DestinationIDs: "*any"`.

السيناريو 2: لا يوجد خيار احتياطي للـ Wildcard

```
# لدى العميل رصيدين:
{
    "BalanceId": "UK_Voice_100min",
    "BalanceType": "*voice",
    "DestinationIDs": "Dest_International_UK",  # فقط بادئة "44"
    "Units": 100 * 60 * 1000000000
}
# لا يوجد رصيد نقدي مع وجهة "*any"

# يتصل العميل بأستراليا (+61)
```

النتيجة:

- ❌ المكالمة **محجوبة** (لا تتطابق الوجهة مع بادئة "61")
- رصيد الدقائق في المملكة المتحدة غير مستخدم (يتطابق فقط مع "44")
- لا يوجد خيار احتياطي نقدي لتغطية المكالمة

أو أضف وجهة أستراليا المحددة `DestinationIDs: "*any"` **الإصلاح:** أضف رصيدًا نقديًا مع إلى رصيدهم.

السيناريو 3: يعمل خيار الاحتياطي للـ Wildcard

```
# لدى العميل:
# الرصيد 1: دقائق المملكة المتحدة
{
    "BalanceId": "UK_Voice_100min",
    "BalanceType": "*voice",
    "DestinationIDs": "Dest_International_UK",
    "Units": 100 * 60 * 1000000000,
    "BalanceWeight": 1200
}

# الرصيد 2: نقدي مع wildcard
{
    "BalanceId": "PAYG_Credit",
    "BalanceType": "*monetary",
    "DestinationIDs": "*any",  # يتطابق مع أي شيء
    "Units": 5000,  # $50
    "BalanceWeight": 1000  # أولوية أقل
}

# يتصل العميل بأستراليا (+61)
```

النتيجة:

- ✅ **مسموح بها** المكالمة
- رصيد دقائق المملكة المتحدة لم يُستخدم (لا يتطابق مع "61")
- تم استخدام رصيد PAYG Credit (يتطابق مع "*any")
- يتم تحصيل الرسوم وفقًا لسعر PAYG لأستراليا

الاختبار والمحاكاة

يمكن اختبار تغييرات الوجهات ومحاكاتها في بيئات غير إنتاجية قبل تطبيقها على الإنتاج. يتيح لك ذلك:

- التحقق من صحة رموز PLMN الجديدة

- اختبار تأثير توسيع/تقييد التغطية
- التحقق من أن الأرصدة الحالية تتصرف كما هو متوقع مع الوجهات المحدثة
- التأكد من أن آليات الاحتياطي تعمل بشكل صحيح

التوصية: دائمًا اختبر تحديثات الوجهات في بيئة اختبار مع حسابات اختبار قبل تطبيقها على الإنتاج.

النقاط الرئيسية

1. **الوجهات ديناميكية** - تؤثر التغييرات على جميع الأرصدة التي تشير إليها بمجرد تحميل خطة التعريفة
2. **تحميل خطة التعريفة لتفعيل** - استخدم `LoadTariffPlanFromStorDb` لجعل تغييرات الوجهة نشطة في وقت التشغيل
3. **لا تطابق الوجهة = لا خدمة** - يمنع النظام العملاء إذا لم يتطابق استخدامهم مع أي وجهة
4. **Wildcard (`*any`) قوي** - استخدمه كاحتياطي نقدي لمنع الحجب غير المتوقع
5. **إعادة تحميل الذاكرة المؤقتة مطلوبة** - دائمًا أعد تحميل ذاكرة التخزين المؤقت للوجهة بعد تحميل خطة التعريفة
6. **اختبر قبل الإنتاج** - تؤثر تغييرات الوجهة على جميع العملاء بمجرد تحميلها

ملفات تعريف الأسعار للأرصدة النقدية/الدفع حسب الاستخدام

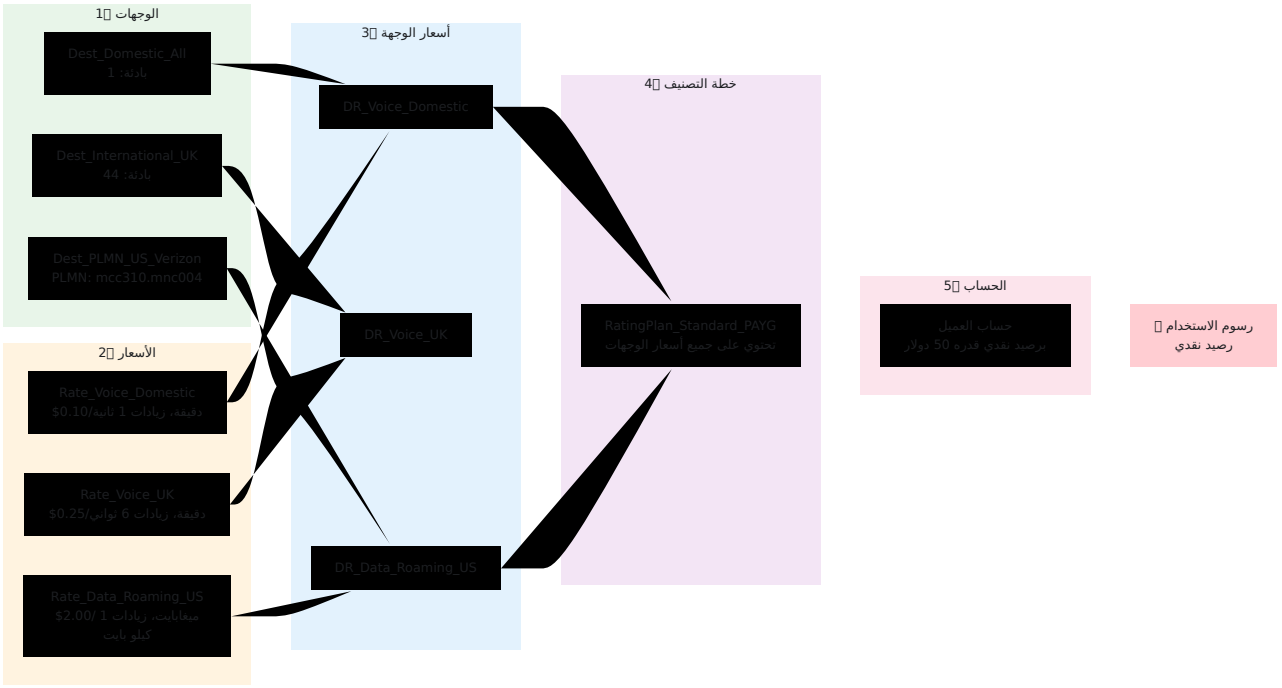
تحتاج إلى تحديد **ملفات تعريف الأسعار** التي تحدد مقدار الرسوم لكل وحدة لمختلف الوجهات، عند استخدام الأرصدة النقدية (PAYG - الدفع عند الاستخدام).

ملاح��ة مهمة حول مرونة CGRateS: CGRateS مرن للغاية ويمكنه تطبيق أسعار وخطط مختلفة بناءً على عوامل عديدة بما في ذلك:

- من أين يتصل العميل (PLMN - تم تغطيته هنا)
- ما الذي يتصلون به (وجهة جغرافية - تم تغطيته هنا)
- وقت اليوم (تصنيف ذروة/غير ذروة)
- مستوى العميل (VIP، إلخ. قياسي)
- مستوى جودة الخدمة
- والعديد من سيناريوهات التصنيف المعقدة الأخرى

والتصنيف القائم على الوجهة PLMN لتجوال **CRM** تركز هذه الوثيقة على **تكوين مستوى** الجغرافية، والتي تغطي أكثر حالات الاستخدام شيوعًا. السيناريوهات المتقدمة للتصنيف (وقت اليوم، مستويات العملاء، التوجيه القائم على الجودة) ممكنة في CGRateS ولكنها خارج نطاق هذه الوثيقة التي تركز على CRM.

كيف تعمل ملفات تعريف الأسعار



تحديد ملفات تعريف الأسعار

تحدد ملفات تعريف الأسعار تكلفة كل وحدة لأنوا�ع الاستخدام المختلفة. إليك أمثلة للأسعار للصوت و SMSوالبيانات:

```
#
# ============================================================================
# أسعار الصوت (لكل دقيقة)
#
# ============================================================================

# صوت محلي: $0.10/دقيقة
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPRate',
    'params': [{
        "TPid": tpid,
        "ID": "Rate_Voice_Domestic",
        "RateSlots": [{
            "ConnectFee": 0,
            "Rate": 0.10,
            "RateUnit": "60s",          # 1 دقيقة
            "RateIncrement": "1s",      # فواتير بالدقيقة
            "GroupIntervalStart": "0s"
        }]
    }]
})

# صوت المملكة المتحدة: $0.25/دقيقة
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPRate',
    'params': [{
        "TPid": tpid,
        "ID": "Rate_Voice_UK",
        "RateSlots": [{
            "ConnectFee": 0.05,         # $0.05 رسوم الاتصال
            "Rate": 0.25,
            "RateUnit": "60s",
            "RateIncrement": "6s",      # كتل 6 ثواني
            "GroupIntervalStart": "0s"
        }]
    }]
})

#
# ============================================================================
# أسعار SMS (لكل رسالة)
#
# ============================================================================
```

```
# SMS رسالة/0.05$ :محلي
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPRate',
    'params': [{
        "TPid": tpid,
        "ID": "Rate_SMS_Domestic",
        "RateSlots": [{
            "ConnectFee": 0,
            "Rate": 0.05,
            "RateUnit": "1",          # 1 SMS
            "RateIncrement": "1",     # لكل رسالة
            "GroupIntervalStart": "0s"
        }]
    }]
})

#
# أسعار البيانات (لكل ميغابايت) - للتجوال
#

# التجوال على الشبكات الأمريكية المتميزة (Verizon، AT&T): $2.00/ميغابايت
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPRate',
    'params': [{
        "TPid": tpid,
        "ID": "Rate_Data_Roaming_US_Premium",
        "RateSlots": [{
            "ConnectFee": 0,
            "Rate": 2.00,
            "RateUnit": "1048576",    # 1 ميغابايت بالبايت
            "RateIncrement": "1024",  # لكل كيلو بايت
            "GroupIntervalStart": "0s"
        }]
    }]
})
```

حقول سعر الوحدة:

- **ConnectFee** - رسوم لمرة واحدة لكل جلسة
- **Rate** - التكلفة لكل RateUnit

- **RateUnit** - حجم وحدة الفوترة (رسالة 1، 1MB، 60s)
- **RateIncrement** - الحد الأدنى للوحدة القابلة للفوترة
- **GroupIntervalStart** - عندما تبدأ هذه الفئة السعرية

ربط الوجهات بالأسعار

بعد تحديد الأسعار، اربطها بالوجهات باستخدام أسعار الوجهات:

```
# ربط الوجهة المحلية بسعر الصوت المحلي
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestinationRate',
    'params': [{
        "TPid": tpid,
        "ID": "DR_Voice_Domestic",
        "DestinationRates": [{
            "DestinationId": "Dest_Domestic_All",
            "RateId": "Rate_Voice_Domestic",
            "RoundingMethod": "*up",
            "RoundingDecimals": 4
        }]
    }]
})

# ربط الوجهة في المملكة المتحدة بسعر الصوت في المملكة المتحدة
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestinationRate',
    'params': [{
        "TPid": tpid,
        "ID": "DR_Voice_UK",
        "DestinationRates": [{
            "DestinationId": "Dest_International_UK",
            "RateId": "Rate_Voice_UK",
            "RoundingMethod": "*up",
            "RoundingDecimals": 4
        }]
    }]
})

# ربط PLMN Verizon بسعر بيانات التجوال
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestinationRate',
    'params': [{
        "TPid": tpid,
        "ID": "DR_Data_Roaming_US_Premium",
        "DestinationRates": [{
            "DestinationId": "Dest_PLMN_US_Verizon",
            "RateId": "Rate_Data_Roaming_US_Premium",
            "RoundingMethod": "*up",
            "RoundingDecimals": 4
        }]
```

```
    }]
})
```

إنشاء خطة التصنيف

اجمع جميع أسعار الوجهات في خطة تصنيف:

```
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPRatingPlan',
    'params': [{
        "TPid": tpid,
        "ID": "RatingPlan_Standard_PAYG",
        "RatingPlanBindings": [
            # أسعار الصوت (أعلى وزن = أكثر تحديدًا)
            {"DestinationRatesId": "DR_Voice_UK", "TimingId":
"*any", "Weight": 40},
            {"DestinationRatesId": "DR_Voice_Domestic",
"TimingId": "*any", "Weight": 20},

            # أسعار SMS
            {"DestinationRatesId": "DR_SMS_Domestic", "TimingId":
"*any", "Weight": 20},

            # أسعار البيانات أثن��ء التجوال
            {"DestinationRatesId": "DR_Data_Roaming_US_Premium",
"TimingId": "*any", "Weight": 50}
        ]
    }]
})
```

تحميل خطة التعريفة

أخيرًا، قم بتحميل خطة التعريفة إلى CGRateS:

```
# تحميل خطة التعريفة
OCS_Obj.SendData({
    'method': 'ApierV1.LoadTariffPlanFromStorDb',
    'params': [{
        "TPid": tpid,
        "DryRun": False,
        "Validate": True
    }]
})

# إعادة تحميل الذاكرة المؤقتة
OCS_Obj.SendData({
    'method': 'CacheSv1.ReloadCache',
    'params': [{
        "Tenant": tenant,
        "DestinationIDs": ["*all"]
    }]
})
```

انظر أيضًا:

- تحديد المنتجات - سير العمل الكامل لإنشاء المنتجات
- أفعال CGRateS وسلوكيات الشحن - كيفية تحديد الأفعال وطرق إدارة الأرصدة

أفضل الممارسات

1. كن محددًا مع الشبكة المحلية

```
# جيد - شبكة منزلية محددة
"DestinationIDs": "Dest_PLMN_OnNet",  # mcc505.mnc057

# سيئ - واسع جدًا، يتطابق مع المنافسين
"DestinationIDs": "Dest_PLMN_AU_All",  # mcc505 (جم🔴ع مشغلي أستراليا)
```

2. استخدم أسماء وجهات ذات معنى

```
# جيد - توثيق ذاتي
Dest_PLMN_US_Verizon
Dest_International_UK
Dest_PLMN_Zone_Europe

# سيئ - غير واضح
Dest_Data_1
Dest_Voice_Package
```

3. قم بتجميع الوجهات ذات الصلة منطقيًا

```
# جيد - تجميع منطقي
Dest_Domestic_All
Dest_Domestic_TollFree
Dest_Domestic_Premium

# سيئ - دقيق جدًا
Dest_Area_Code_212
Dest_Area_Code_213
Dest_Area_Code_214
```

4. وثق مصادر PLMN

أضف تعليقات تشير إلى مصدر رموز MCC/MNC، عند تحديد وجهات PLMN:

```
# Verizon Wireless
# المصدر: https://www.mcc-mnc.com/ (تم التحقق منه 12-2024)
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_US_Verizon",
        "Prefixes": [
            "mcc310.mnc004",  # Verizon - على مستوى البلاد
            "mcc310.mnc010",  # Verizon - الشرق
            ...
        ]
    }]
})
```

استكشاف الأخطاء وإصلاحها

المشكلة: الرصيد لا يتطابق مع الوجهة

الأعراض: لدى العميل رصيد ولكنه لا يمكنه ��ستخدامه لوجهة معينة

تحقق:

```
# 1. تحقق من وجود الوجهة
OCS_Obj.SendData({
    'method': 'ApierV2.GetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_US_Verizon"
    }]
})

# 2. تحقق من حقل DestinationIDs في الرصيد
# تأكد من أنه يتضمن الوجهة أو "*any"

# 3. تحقق مما إذا كان العميل فعلاً على تلك الشبكة PLMN: بالنسبة لـ
# تحقق من رسائل CCR Diameter للحصول على AVP Visited-PLMN-Id
```

المشكلة: استخدام البيانات يتم احتسابه كتجوال

الأعراض: استخدام البيانات المحلية يستهلك رصيد التجوال بدلاً من رصيد الشبكة المحلية

السبب: لم يتم تعريف PLMN العميل في `Dest_PLMN_OnNet`

الحل:

```
# تحقق من PLMN الذي يتواجد عليه العميل (من CCR Diameter)
# تحقق من أن الرصيد يتضمن ذلك PLMN:

OCS_Obj.SendData({
    'method': 'ApierV2.GetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_OnNet"
    }]
})
# تأكد من أن Prefixes تتضمن شبكة العميل الفعلية
# مثال: ["mcc505.mnc057"] جب أن تتطابق مع شبكة الخاصة بالعميل SIM
```

المشكلة: المكالمات الدولية لا تعمل

الأعراض: لدى العميل "دقائق دولية" ولكن المكالمات تفشل

تحقق:

1. **بادئة الوجهة**: هل يتطابق الرقم الذي تم الاتصال به مع بادئة الوجهة؟
   - "يحتاج إلى وجهة مع بادئة "44 → الاتصال بـ +44-20-xxxx
2. **الرصيد**: هل يتضمن الرصيد تلك الوجهة؟ **DestinationIDs**
   - تحقق من أن الرصيد يحتوي على `"DestinationIDs": "Dest_International_UK"` أو `"Dest_International_All"`
3. **الرصيد غير منتهي**: تحقق من `ExpiryTime`
4. **الرصيد لديه وحدات**: تحقق من `Units > 0`

الوثائق ذات الصلة

- وسلوكيات الشحن - كيفية استخدام الوجهات في الأفعال، طرق CGRateS أفعال الأرصدة، واستراتيجيات التصنيف
- تحديد المنتجات - سير العمل الكامل لإنشاء المنتجات

أدلة تفصيلية حول Playbooks في Ansible

تُقدم منتجات OmniCRM باستخدام **Ansible**، مما يسمح بإدارة الخدمات بشكل آلي بناءً على المتطلبات المحددة لكل منتج ومخزونه المرتبط.

للخدمات Ansible للحصول على مثال كامل عن التوفير المعتمد على SIM انظر أيضًا: توفير بطاقة SIM الفعلية و eSIMs، بما في ذلك بطاقات SIM المحمولة.

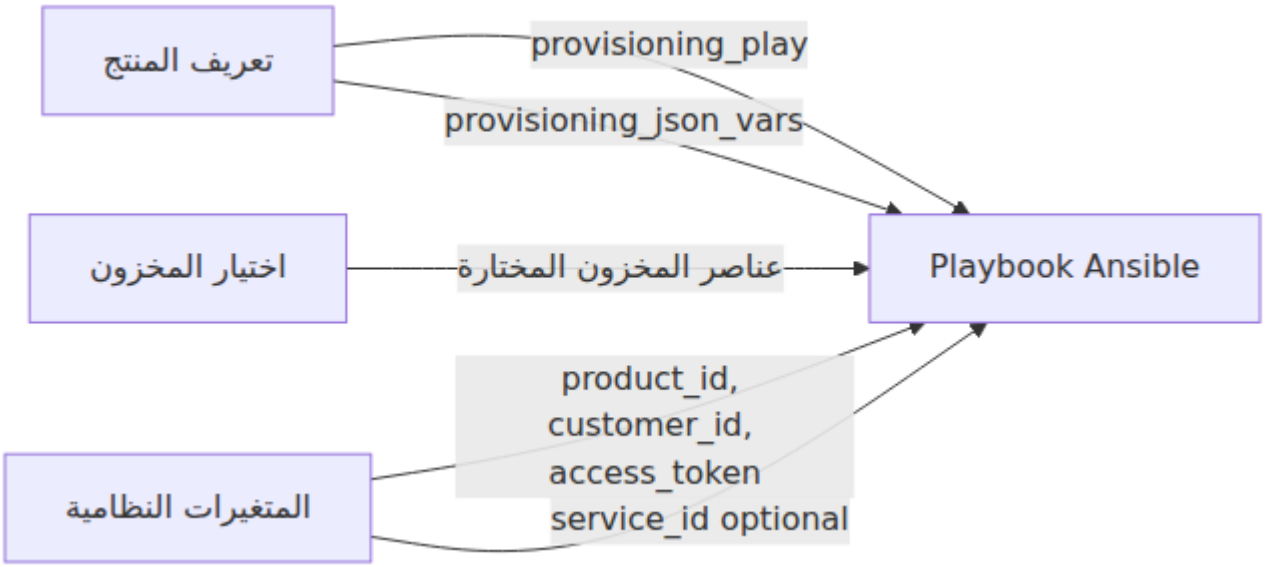
كيف تعمل Playbooks والمنتجات معًا

المفهوم الحاسم: Playbooks في OmniCRM هي ما ينشئ فعليًا الخدمات. عندما تعين لمنتج، فإنك تحدد **ما يحدث** عندما يتم توفير ذلك المنتج. يمكن أن يعني ذلك أشياء مختلفة لمنتجات مختلفة.

Playbooks تُحفز المنتجات

عندما يتم توفير منتج في OmniCRM:

1. يحدد تعريف المنتج أي Playbook يتم تشغيله (عبر حقل `provisioning_play`)
2. يمرر المنتج متغيرات إلى Playbook (عبر `provisioning_json_vars` واختيارات المخزون)
3. يتم تنفيذ Playbook ويقوم بما تم برمجته للقيام به
4. يحدد Playbook ما يتم إنشاؤه (إذا كان هناك شيء)



Playbooks ما يمكن أن تفعله

يمكن أن تقوم Playbook بـ توفير واحدة:

إنشاء خدمات متعددة
قد تقوم Playbook بإنشاء مجمعة منتج:

- سجل خدمة إنترنت رئيسية
- سجل خدمة IPTV إضافية
- سجل خدمة VoIP
- كل ذلك من خلال إجراء توفير منتج واحد

إنشاء خدمات بدون أي
بعض Playbooks لا تنشئ سجلات خدمة على الإطلاق:

- Playbook تقوم فقط بتكوين معدات CPE
- Playbook ترسل تكوينًا إلى معدات الشبكة
- Playbook تقوم بتحديث الأنظمة الخارجية

إنشاء خدمة واحدة
النمط الأكثر شيوعًا:

- إنشاء سجل خدمة واحد للعميل
- ربط المخزون بتلك الخدمة
- إعداد الفوترة لتلك الخدمة

تعديل الخدمات الموجودة
التعبئة والإضافات Playbooks:

- لا تنشئ خدمات جديدة
- تحديث سجلات الخدمة الموجودة (إضافة بيانات، تمديد انتهاء الصلاحية، إلخ)
- إضافة أرصدة إلى حسابات الفوترة الموجودة

تنفيذ إجراءات بدون سجلات خدمة
بعض Playbooks هي عمليات بحتة:

- إعادة تعيين أرصدة الحسابات
- تبديل عناصر المخزون بين العملاء
- توليد تقارير أو تكوينات

مثال: سلوكيات Playbook المختلفة

```
# المنتج 1: خدمة SIM المحمولة (تقوم بإنشاء خدمة واحدة)
# provisioning_play: play_simple_service
- إنشاء سجل خدمة في CRM
- إنشاء حساب فوترة في OCS
- تعيين بطاقة SIM ورقم الهاتف من المخزون
- إرسال بريد إلكتروني ترحيبي

# المنتج 2: حزمة الإنترنت (تقوم بإنشا 3 خدمات)
# provisioning_play: play_bundle_internet_tv_voice
- إنشاء سجل خدمة إنترنت
- إنشاء سجل خدمة IPTV
- إنشاء سجل خدمة VoIP
- ربط جميعها بنفس العميل
- حساب فوترة واحد للحزمة

# المنتج 3: تعبئة البيانات (تقوم بإنشاء 0 خدمات)
# provisioning_play: play_topup_no_charge
- العثور على الخدمة الحالية بواسطة service_id
- إضافة رصيد بيانات إلى حساب OCS الموجود
- تحديث تاريخ انتهاء الخدمة
- لا توجد خدمة جديدة تم إنشاؤها

# المنتج 4: تكوين CPE (تقوم بإنشاء 0 خدمات)
# provisioning_play: play_prov_cpe_mikrotik
- توليد تكوين جهاز التوجيه
- تحديث سجل المخزون بالتكوين
- إرسال البريد الإلكتروني بالتكوين إلى فريق الدعم
- لا توجد خدمة تم إنشاؤها (فقط إعداد المعدات)
```

تحدد النقطة الرئيسية: **Playbook محفز هو مجرد والمنتج، السلوك.**

مقابل المهام Plays

فهم الفرق بين Plays والمهام هو أمر أساسي للعمل مع Playbooks في OmniCRM.

Play (Playbook)
سير عمل توفير كامل ينظم مهام متعددة لتحقيق هدف تجاري. Plays هي Playbooks على

ويتم الإشارة إليها في `OmniCRM-API/Provisioners/plays/` المستوى الأعلى المخزنة في تعريفات المنتجات.

أمثلة:

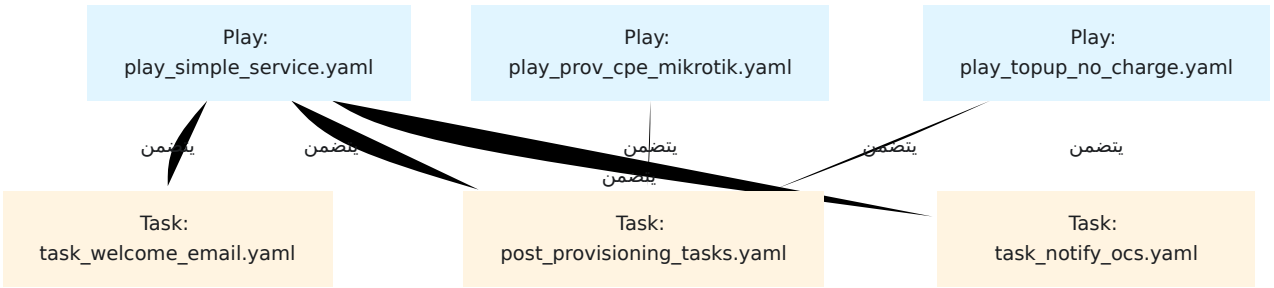
- `play_simple_service.yaml` - توفير خدمة أساسية
- `play_topup_no_charge.yaml` - تطبيق تعبئة مجانية على خدمة
- `play_prov_cpe_mikrotik.yaml` - تكوين معدات العملاء

Task (مكون قابل لإعادة الاستخدام)
مجموعة من العمليات المستقلة والقابلة لإعادة الاستخدام التي يمكن تضمينها بواسطة عدة Plays.
تُسبق المهام بـ `task_` وتعيش في نفس الدليل.

أمثلة:

- `task_welcome_email.yaml` - إرسال بريد إلكتروني ترحيبي إلى عميل
- `task_activate_olt.yaml` - تفعيل معدات OLT
- `task_notify_ocs.yaml` - إرسال إشعارات إلى نظام الفوترة

العلاقة بينها:



```yaml
# play_simple_service.yaml (A Play)
- name: Play توفير بسيطة
  hosts: localhost
  tasks:
    - name: الكتلة الرئيسية للتوفير
      block:
        - name: إنشاء الخدمة
          uri: ...

        - name: تكوين الفوترة
          uri: ...

        # تضمين مهمة قابلة لإعادة الاستخدام
        - include_tasks: task_welcome_email.yaml

        # تضمين المهام بعد التوفير
        - include_tasks: post_provisioning_tasks.yaml
```

هيكل Playbook وتشريحه

تتبع جميع Playbooks في OmniCRM هيكلًا متسقًا. فهم هذا الهيكل أمر ضروري لإنشاء وصيانة Playbooks.

الهيكل الأساسي

تبدأ كل Playbook بهذه العناوين القياسية:

```yaml
- name: الاسم الوصفي لـ Playbook
  hosts: localhost           # دائمًا localhost لـ OmniCRM
  gather_facts: no           # معطل لأداء أفضل
  become: False              # لا تصعيد الامتيازات

  tasks:
    - name: الكتلة الرئيسية
      block:
        # مهام التوفير تذهب هنا

      rescue:
        # مهام التراجع/التنظيف تذهب هنا
```

شرح العنوان

name
اسم وصفي يظهر في سجلات التوفير وواجهة المستخدم. يظهر هذا كـ `playbook_description` في سجل التوفير.

hosts: localhost
تعمل جميع Playbooks في OmniCRM على localhost لأنها تتفاعل مع الأنظمة البعيدة عبر APIs، وليس SSH.

gather_facts: no
تم تعطيل جمع الحقائق في Ansible لأن:

- لا نحتاج إلى معلومات النظام
- يضيف عبءًا غير ضروري
- يمكن أن يتسبب في تعطل المتصفحات إذا تم عرضه في مخرجات التصحيح

become: False
لا حاجة لتصعيد الامتيازات لأننا نقوم بإجراء مكالمات API، وليس تعديل ملفات النظام.

تحميل التكوين

يجب على كل Playbook تحميل ملف التكوين المركزي:

```yaml
tasks:
  - name: تضمين متغيرات crm_config
    ansible.builtin.include_vars:
      file: "../../crm_config.yaml"
      name: crm_config
```

هذا يجعل التكوين متاحًا كـ `crm_config.ocs.cgrates`، `crm_config.crm.base_url`، إلخ.

عادةً ما يحتوي `crm_config.yaml` على:

```yaml
ocs:
  cgrates: "10.0.1.100:2080"
  ocsTenant: "default_tenant"
crm:
  base_url: "https://crm.example.com"
```

أنماط الوصول إلى المتغيرات

يمكن أن تأتي المتغيرات من عدة مصادر:

من تعريف المنتج:

```yaml
- name: الوصول إلى product_id الممرر بواسطة OmniCRM
  debug:
    msg: "توفير المنتج {{ product_id }}"
```

من اختيار المخزون:

```yaml
- name: الحصول على معرف المخزون لبطاقة SIM
  set_fact:
    sim_card_id: "{{ hostvars[inventory_hostname]['SIM Card'] |
int }}"
  when: "'SIM Card' in hostvars[inventory_hostname]"
```

من استجابات API:

```
- name: الحصول على معلومات المنتج من API CRM
  uri:
    url: "http://localhost:5000/crm/product/product_id/{{
product_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    return_content: yes
  register: api_response_product

- name: استخدام اسم المنتج
  debug:
    msg: "اسم المنتج هو {{ api_response_product.json.product_name
}}"
```

أنماط Playbook الشائعة

نمط توفير الخدمة

هذا هو النمط الأكثر شيوعًا لإنشاء خدمات جديدة.

```yaml
- name: Playbook توفير الخدمة
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: الكتلة الرئيسية
      block:

      # 1. تحميل التكوين
      - name: تضمين متغيرات crm_config
        ansible.builtin.include_vars:
          file: "../../crm_config.yaml"
          name: crm_config

      # 2. الحصول على معلومات المنتج
      - name: الحصول على معلومات المنتج من API CRM
        uri:
          url: "http://localhost:5000/crm/product/product_id/{{
product_id }}"
          method: GET
          headers:
            Authorization: "Bearer {{ access_token }}"
          return_content: yes
          validate_certs: no
        register: api_response_product

      # 3. الحصول على معلومات العميل
      - name: الحصول على معلومات العميل من API CRM
        uri:
          url: "http://localhost:5000/crm/customer/customer_id/{{
customer_id }}"
          method: GET
          headers:
            Authorization: "Bearer {{ access_token }}"
          return_content: yes
        register: api_response_customer

      # 4. تعيين الحقائق من البيانات المسترجعة
      - name: تعيين حقائق الحزمة
        set_fact:
          package_name: "{{ api_response_product.json.product_name
}}"
```

```
          package_comment: "{{ api_response_product.json.comment
}}"
          setup_cost: "{{
api_response_product.json.retail_setup_cost }}"
          monthly_cost: "{{ api_response_product.json.retail_cost
}}"

      # 5. توليد معرفات فريدة
      - name: توليد UUID
        set_fact:
          uuid: "{{ 99999999 | random | to_uuid }}"

      - name: توليد UUID الخدمة
        set_fact:
          service_uuid: "Service_{{ uuid[0:8] }}"

      # 6. إنشاء حساب في نظام الفوترة
      - name: إنشاء حساب في OCS/CGRateS
        uri:
          url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
          method: POST
          body_format: json
          headers:
            Content-Type: "application/json"
          body:
            {
              "method": "ApierV2.SetAccount",
              "params": [{
                  "Tenant": "{{ crm_config.ocs.ocsTenant }}",
                  "Account": "{{ service_uuid }}",
                  "ActionPlanIds": [],
                  "ActionPlansOverwrite": true,
                  "ExtraOptions": {
                    "AllowNegative": false,
                    "Disabled": false
                  },
                  "ReloadScheduler": true
              }]
            }
          status_code: 200
        register: ocs_response

      - name: التحقق من إنشاء حساب OCS
        assert:
```

```
      that:
        - ocs_response.status == 200
        - ocs_response.json.result == "OK"

    # 7. إضافة رصيد أولي
    - name: إضافة رصيد مالي 0
      uri:
        url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
        method: POST
        body_format: json
        body:
          {
            "method": "ApierV1.AddBalance",
            "params": [{
                "Tenant": "{{ crm_config.ocs.ocsTenant }}",
                "Account": "{{ service_uuid }}",
                "BalanceType": "*monetary",
                "Categories": "*any",
                "Balance": {
                    "ID": "رصيد أولي",
                    "Value": 0,
                    "ExpiryTime": "+4320h",
                    "Weight": 1,
                    "Blocker": true
                }
            }]
          }
        status_code: 200
      register: balance_response

    # 8. إنشاء سجل الخدمة في CRM
    - name: الحصول على التاريخ والوقت الحالي بتنسيق ISO 8601
      command: date --utc +%Y-%m-%dT%H:%M:%S%z
      register: current_date_time

    - name: إضافة الخدمة عبر API
      uri:
        url: "http://localhost:5000/crm/service/"
        method: PUT
        body_format: json
        headers:
          Content-Type: "application/json"
          Authorization: "Bearer {{ access_token }}"
        body:
```

```
            {
              "customer_id": "{{ customer_id }}",
              "product_id": "{{ product_id }}",
              "service_name": "{{ package_name }} - {{
service_uuid }}",
              "service_type": "generic",
              "service_uuid": "{{ service_uuid }}",
              "service_billed": true,
              "service_taxable": true,
              "service_provisioned_date": "{{
current_date_time.stdout }}",
              "service_status": "Active",
              "wholesale_cost": "{{
api_response_product.json.wholesale_cost | float }}",
              "retail_cost": "{{ monthly_cost | float }}"
            }
          status_code: 200
        register: service_creation_response

      # 9. إضافة معاملة تكلفة الإعداد
      - name: إضافة معاملة تكلفة الإعداد عبر API
        uri:
          url: "http://localhost:5000/crm/transaction/"
          method: PUT
          headers:
            Content-Type: "application/json"
            Authorization: "Bearer {{ access_token }}"
          body_format: json
          body:
            {
              "customer_id": {{ customer_id | int }},
              "service_id": {{
service_creation_response.json.service_id | int }},
              "title": "{{ package_name }} - تكاليف الإعداد",
              "description": "تكاليف الإعداد لـ {{ package_comment
}}",
              "invoice_id": null,
              "retail_cost": "{{ setup_cost | float }}"
            }
          return_content: yes
        register: transaction_response

      # 10. تضمين المهام بعد التوفير
      - include_tasks: post_provisioning_tasks.yaml
```

```
    rescue:

    # قسم التراجع/التنظيف
    - name: طباعة جميع المتغيرات لأغراض التصحيح
      debug:
        var: hostvars[inventory_hostname]

    - name: إزالة الحساب في OCS
      uri:
        url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
        method: POST
        body_format: json
        body:
          {
            "method": "ApierV2.RemoveAccount",
            "params": [{
                "Tenant": "{{ crm_config.ocs.ocsTenant }}",
                "Account": "{{ service_uuid }}",
                "ReloadScheduler": true
            }]
          }
        status_code: 200
      ignore_errors: True
      when: service_uuid is defined

    - name: إذا تم إنشاؤها CRM حذف الخدمة من
      uri:
        url: "http://localhost:5000/crm/service/service_id/{{
service_creation_response.json.service_id }}"
        method: DELETE
        headers:
          Authorization: "Bearer {{ access_token }}"
        status_code: 200
      ignore_errors: True
      when: service_creation_response is defined

    - name: الفشل إذا لم يكن إلغاء التوفير مقصودًا
      assert:
        that:
          - action == "deprovision"
```

نمط التعبئة/إعادة الشحن

يستخدم لإضافة أرصدة أو بيانات أو وقت إلى الخدمات الموجودة.

```yaml
- name: Playbook تعبئة الخدمة
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: تضمين متغيرات crm_config
      ansible.builtin.include_vars:
        file: "../../crm_config.yaml"
        name: crm_config

    # 1. الحصول على معلومات الخدمة
    - name: الحصول على معلومات الخدمة من API CRM
      uri:
        url: "http://localhost:5000/crm/service/service_id/{{
service_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
      register: api_response_service

    # 2. الحصول على معلومات المنتج (ما يجب تعبئته)
    - name: الحصول على معلومات المنتج من API CRM
      uri:
        url: "http://localhost:5000/crm/product/product_id/{{
product_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
      register: api_response_product

    # 3. استخراج تفاصيل الخدمة
    - name: تعيين حقائق الخدمة
      set_fact:
        service_uuid: "{{ api_response_service.json.service_uuid
}}"
        customer_id: "{{ api_response_service.json.customer_id }}"
        package_name: "{{ api_response_product.json.product_name
}}"
        topup_value: "{{ api_response_product.json.retail_cost }}"
```

```
    # 4. تنفيذ الإجراء في نظ🔴م الفوترة (تعبئة مجانية)
    - name: تنفيذ الإجراء لإضافة أرصدة
      uri:
        url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
        method: POST
        body_format: json
        body:
          {
            "method": "APIerSv1.ExecuteAction",
            "params": [{
                "Tenant": "{{ crm_config.ocs.ocsTenant }}",
                "Account": "{{ service_uuid }}",
                "ActionsId": "Action_Topup_Standard"
            }]
          }
        status_code: 200
      register: action_response

    - name: التحقق من تنفيذ الإجراء بنجاح
      assert:
        that:
          - action_response.status == 200
          - action_response.json.result == "OK"

    # 5. إعادة تعيين أي حدود تم تشغيلها
    - name: إعادة تعيين ActionTriggers
      uri:
        url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
        method: POST
        body_format: json
        body:
          {
            "method": "APIerSv1.ResetAccountActionTriggers",
            "params": [{
                "Tenant": "{{ crm_config.ocs.ocsTenant }}",
                "Account": "{{ service_uuid }}",
                "Executed": false
            }]
          }
        status_code: 200

    # 6. تحديث تواريخ الخدمة
    - name: حساب تاريخ انتهاء جديد
      command: "date --utc +%Y-%m-%dT%H:%M:%S%z -d '+30 days'"
```

```
      register: new_expiry_date

    - name: تحديث الخدمة مع انتهاء جديد
      uri:
        url: "http://localhost:5000/crm/service/{{ service_id }}"
        method: PATCH
        headers:
          Authorization: "Bearer {{ access_token }}"
          Content-Type: "application/json"
        body_format: json
        body:
          {
            "service_deactivate_date": "{{ new_expiry_date.stdout
}}",
            "service_status": "Active"
          }

    # 7. اختياري: إرسال إشعار
    - name: إرسال إشعار SMS
      uri:
        url: "http://sms-gateway/api/send"
        method: POST
        body_format: json
        body:
          {
            "source": "اسم الشركة",
            "destination": "{{ customer_phone }}",
            "message": "تم تعبئة خدمتك. انتهاء جديد: {{
new_expiry_date.stdout }}"
          }
        status_code: 201
      ignore_errors: True
```

نمط توفير CPE

يستخدم لتكوين معدات العملاء (أجهزة التوجيه، المودمات، ONTs).

```yaml
- name: Playbook توفير CPE
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: تضمين متغيرات crm_config
      ansible.builtin.include_vars:
        file: "../../crm_config.yaml"
        name: crm_config

    # 1. الحصول على عنصر المخزون لـ CPE
    - name: تعيين معرف مخزون CPE من hostvars
      set_fact:
        cpe_inventory_id: "{{ hostvars[inventory_hostname]['WiFi
Router CPE'] | int }}"
      when: "'WiFi Router CPE' in hostvars[inventory_hostname]"

    # 2. الحصول على تفاصيل CPE من المخزون
    - name: الحصول على بيانات المخزون لـ CPE
      uri:
        url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ cpe_inventory_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
      register: api_response_cpe

    # 3. الحصول على معلومات موقع العميل
    - name: الحصول على معلومات الموقع من API
      uri:
        url: "{{ crm_config.crm.base_url
}}/crm/site/customer_id/{{ customer_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
      register: api_response_site

    # 4. تحديث مخزون CPE بالموقع
    - name: Patch عنصر مخزون CPE بالموقع
      uri:
```

```
        url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ cpe_inventory_id }}"
        method: PATCH
        body_format: json
        headers:
          Authorization: "Bearer {{ access_token }}"
        body:
          {
            "address_line_1": "{{
api_response_site.json.0.address_line_1 }}",
            "city": "{{ api_response_site.json.0.city }}",
            "state": "{{ api_response_site.json.0.state }}",
            "latitude": "{{ api_response_site.json.0.latitude }}",
            "longitude": "{{ api_response_site.json.0.longitude
}}"
          }
        status_code: 200

    # 5. توليد بيانات الاعتماد
    - name: تعيين اسم مضيف CPE
      set_fact:
        cpe_hostname: "CPE_{{ cpe_inventory_id }}"
        cpe_username: "admin_{{ cpe_inventory_id }}"

    - name: توليد كلمة مرور عشوائية
      set_fact:
        cpe_password: "{{ lookup('pipe', 'cat /dev/urandom | tr -
dc a-zA-Z0-9 | head -c 16') }}"

    # 6. توليد بيانات اعتماد WiFi
    - name: تعيين SSID WiFi
      set_fact:
        wifi_ssid: "Network_{{ cpe_inventory_id }}"

    - name: توليد كلمة مرور WiFi
      set_fact:
        word_list:
          - apple
          - cloud
          - river
          - mountain
          - ocean

    - name: إنشاء WiFi PSK
```

```
      set_fact:
        random_word: "{{ word_list | random }}"
        random_number: "{{ 99999 | random(start=10000) }}"

    - name: دمج WiFi PSK
      set_fact:
        wifi_psk: "{{ random_word }}{{ random_number }}"

    # 7. توليد ملف التكوين
    - name: تعيين اسم ملف التكوين
      set_fact:
        config_name: "{{ cpe_hostname }}_{{ lookup('pipe', 'date
+%Y%m%d%H%M%S') }}.cfg"
        config_dest: "/tmp/{{ cpe_hostname }}_{{ lookup('pipe',
'date +%Y%m%d%H%M%S') }}.cfg"

    - name: إنشاء التكوين من القالب
      template:
        src: "templates/cpe_router_config.j2"
        dest: "{{ config_dest }}"

    # 8. قراءة التكوين الناتج
    - name: قراءة ملف التكوين
      ansible.builtin.slurp:
        src: "{{ config_dest }}"
      register: config_content

    # 9. تحديث المخزون بمعلومات التوفير
    - name: Patch مخزون CPE بالتكوين
      uri:
        url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ cpe_inventory_id }}"
        method: PATCH
        body_format: json
        headers:
          Authorization: "Bearer {{ access_token }}"
        body:
          {
            "itemtext3": "{{ wifi_ssid }}",
            "itemtext4": "{{ wifi_psk }}",
            "management_url": "{{ cpe_hostname }}",
            "management_username": "{{ cpe_username }}",
            "management_password": "{{ cpe_password }}",
            "config_content": "{{ config_content.content |
```

```
b64decode }}",
            "inventory_notes": "تم التوفير: {{ lookup('pipe',
'date +%Y-%m-%d') }}"
          }
        status_code: 200

    # 10. إرسال التكوين إلى فريق الدعم
    - name: إرسال التكوين عبر البريد الإلكتروني إلى الدعم
      uri:
        url: "https://api.mailjet.com/v3.1/send"
        method: POST
        body_format: json
        headers:
          Content-Type: "application/json"
        body:
          {
            "Messages": [{
              "From": {
                "Email": "provisioning@example.com",
                "Name": "نظام التوفير"
              },
              "To": [{
                "Email": "support@example.com",
                "Name": "فريق الدعم"
              }],
              "Subject": "تكوين CPE - {{ cpe_hostname }}",
              "Attachments": [{
                "ContentType": "text/plain",
                "Filename": "{{ config_name }}",
                "Base64Content": "{{ config_content.content }}"
              }]
            }]
          }
        user: "{{ mailjet_api_key }}"
        password: "{{ mailjet_api_secret }}"
        force_basic_auth: true
        status_code: 200
```

نمط التجديد التلقائي

تكوين رسوم متكررة تلقائية أو تجديدات باستخدام خطط العمل في CGRateS.

```
# التعبئة الذي يقوم بإعداد التجديد التلقائي جزء من Playbook

# 1. تطبيع معلمة auto_renew
- name: تطبيع auto_renew إلى قيمة منطقية
  set_fact:
    auto_renew_bool: "{{ (auto_renew | string | lower) in ['true',
'1', 'yes'] }}"

# 2. إنشاء إجراء للتجديد التلقائي
- name: إنشاء إجراء للتجديد التلقائي
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.SetActions",
        "params": [{
          "ActionsId": "Action_AutoTopup_{{ service_uuid }}_{{
product_id }}",
          "Overwrite": true,
          "Actions": [
            {
              "Identifier": "*http_post",
              "ExtraParameters": "{{ crm_config.crm.base_url
}}/crm/provision/simple_provision_addon/service_id/{{ service_id
}}/product_id/{{ product_id }}"
            },
            {
              "Identifier": "*cdrlog",
              "BalanceType": "*generic",
              "ExtraParameters": "
{\"Category\":\"^activation\",\"Destination\":\"Auto Renewal\"}"
            }
          ]
        }]
      }
    status_code: 200
  register: action_response
  when: auto_renew_bool

# 3. إنشاء خطة العمل الشهرية
- name: إنشاء خطة العمل للتجديد الشهري
```

```
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.SetActionPlan",
        "params": [{
          "Id": "ActionPlan_Monthly_{{ service_uuid }}_{{
product_id }}",
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "ActionPlan": [{
            "ActionsId": "Action_AutoTopup_{{ service_uuid }}_{{
product_id }}",
            "Years": "*any",
            "Months": "*any",
            "MonthDays": "*any",
            "WeekDays": "*any",
            "Time": "*monthly",
            "StartTime": "*now",
            "Weight": 10
          }],
          "Overwrite": true,
          "ReloadScheduler": true
        }]
      }
    status_code: 200
  when: auto_renew_bool

# 4. تعيين خطة العمل للحساب
- name: تعيين خطة العمل للحساب
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV2.SetAccount",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "ActionPlanIds": ["ActionPlan_Monthly_{{ service_uuid
}}_{{ product_id }}"],
          "ActionPlansOverwrite": true,
```

```
            "ReloadScheduler": true
          }]
        }
      status_code: 200
  when: auto_renew_bool

# 5. إزالة خطة العمل إذا تم تعطيل التجديد التلقائي
- name: إزالة خطة العمل من الحساب
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.RemoveActionPlan",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Id": "ActionPlan_Monthly_{{ service_uuid }}_{{
product_id }}"
        }]
      }
    status_code: 200
  ignore_errors: true
  when: not auto_renew_bool
```

المهام القابلة لإعادة الاستخدام

المهام القابلة لإعادة الاستخدام هي Playbooks صغيرة ومستقلة يمكن تضمينها بواسطة عدة Plays. إنها تعزز إعادة استخدام الكود والاتساق.

مهمة البريد الإلكتروني الترحيبي

`task_welcome_email.yaml` - ترسل بريدًا إلكترونيًا ترحيبيًا للعملاء الجدد.

```
# الأب Play تتوقع هذه المهمة أن يتم تعيين هذه المتغيرات بواسطة:
# - api_response_customer (تفاصيل العميل)
# - package_name (اسم المنتج)
# - monthly_cost (التكلفة المتكررة)
# - setup_cost (التكلفة لمرة واحدة)

- name: تعيين تكوين البريد الإلكتروني
  set_fact:
    mailjet_api_key: "{{ lookup('env', 'MAILJET_API_KEY') }}"
    mailjet_api_secret: "{{ lookup('env', 'MAILJET_SECRET') }}"
    email_from: "noreply@example.com"
    recipients: []

- name: تعيين موضوع البريد الإلكتروني واسم المرسل
  set_fact:
    email_subject: "مرحبًا بك في خدمتنا!"
    email_from_name: "فريق خدمة العملاء"

- name: إعداد قائمة المستلمين من جهات اتصال العميل
  loop: "{{ api_response_customer.json.contacts }}"
  set_fact:
    recipients: "{{ recipients + [{'Email': item.contact_email,
'Name': item.contact_firstname ~ ' ' ~ item.contact_lastname}] }}"

- name: الحصول على اسم أول جهة اتصال
  set_fact:
    first_contact: "{{
api_response_customer.json.contacts[0].contact_firstname }}"

- name: إرسال بريد إلكتروني ترحيبي
  uri:
    url: "https://api.mailjet.com/v3.1/send"
    method: POST
    body_format: json
    headers:
      Content-Type: "application/json"
    body:
      {
        "Messages": [{
          "From": {
            "Email": "{{ email_from }}",
            "Name": "{{ email_from_name }}"
          },
```

```
            "To": "{{ recipients }}",
            "Subject": "{{ email_subject }}",
            "TextPart": "عزيزي {{ first_contact }}, خدمتك جاهزة!.",
            "HTMLPart": "عزيزي {{ first_contact }},<br/><h3>مرحبًا!
</h3><br/>خدمة {{ package_name }} الآن نشطة بك الخاصة.<br/>التكلفة
الشهرية: ${{ monthly_cost }}<br/>رسوم الإعداد: ${{ setup_cost }}
<br/>إذا كانت لديك أي استفسارات، اتصل بـ support@example.com"
          }]
        }
      user: "{{ mailjet_api_key }}"
      password: "{{ mailjet_api_secret }}"
      force_basic_auth: true
      status_code: 200
    register: email_response
```

المهام بعد التوفير

`post_provisioning_tasks.yaml` - عمليات التنظيف والإشعارات القياسية التي تُنفذ بعد كل توفير.

```
# يتم تضمين هذا الملف في نهاية معظم Playbooks للتوفير
# يتعامل مع العمليات الشائعة بعد التوفير

- include_tasks: task_notify_ocs.yaml
```

قد تحتوي `task_notify_ocs.yaml` على:

```
- name: إعلام OCS بإكمال التوفير
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "APIerSv1.ReloadCache",
        "params": [{
          "ArgsCache": "*all"
        }]
      }
    status_code: 200
  ignore_errors: true
```

العمليات الشائعة

العمل مع المخزون

استرجاع تفاصيل المخزون:

```
- name: الحصول على معرف مخزون بطاقة SIM
  set_fact:
    sim_inventory_id: "{{ hostvars[inventory_hostname]['SIM Card']
| int }}"
  when: "'SIM Card' in hostvars[inventory_hostname]"

- name: الحصول على تفاصيل بطاقة SIM
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ sim_inventory_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    return_content: yes
  register: sim_response

- name: استخراج تفاصيل SIM
  set_fact:
    iccid: "{{ sim_response.json.iccid }}"
    imsi: "{{ sim_response.json.imsi }}"
    ki: "{{ sim_response.json.ki }}"
```

تعيين المخزون للعميل:

```
- name: تعيين SIM للعميل
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ sim_inventory_id }}"
    method: PATCH
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "customer_id": {{ customer_id }},
        "service_id": {{ service_id }},
        "item_state": "Assigned"
      }
    status_code: 200
```

عمليات التاريخ والوقت

الحصول على التاريخ/الوقت الحالي:

```
- name: الحصول على التاريخ والوقت الحالي بتنسيق ISO 8601
  command: date --utc +%Y-%m-%dT%H:%M:%S%z
  register: current_date_time

- name: الحصول على تاريخ اليوم فقط
  set_fact:
    today: "{{ lookup('pipe', 'date +%Y-%m-%d') }}"
```

حساب التواريخ المستقبلية:

```
- name: حساب تاريخ انتهاء بعد 30 يومًا
  command: "date --utc +%Y-%m-%dT%H:%M:%S%z -d '+30 days'"
  register: expiry_date

- name: حساب التاريخ بعد 90 يومًا
  command: "date --utc +%Y-%m-%d -d '+{{ days }} days'"
  register: future_date
  vars:
    days: 90
```

توليد قيم عشوائية

UUIDs والمعرفات:

```
- name: توليد UUID
  set_fact:
    uuid: "{{ 99999999 | random | to_uuid }}"

- name: توليد معرف الخدمة
  set_fact:
    service_uuid: "SVC_{{ uuid[0:8] }}"
```

كلمات مرور عشوائية:

```
- name: توليد كلمة مرور آمنة
  set_fact:
    password: "{{ lookup('pipe', 'cat /dev/urandom | tr -dc a-zA-Z0-9 | head -c 16') }}"
```

عبارات مرور سهلة التذكر:

```
- name: تعيين قائمة الكلمات
  set_fact:
    words:
      - alpha
      - bravo
      - charlie
      - delta
      - echo

- name: توليد عبارة مرور
  set_fact:
    word: "{{ words | random }}"
    number: "{{ 99999 | random(start=10000) }}"

- name: دمجها في عبارة مرور
  set_fact:
    passphrase: "{{ word }}{{ number }}"
```

العمل مع CGRateS/OCS

إنشاء الحسابات:

```
- name: إنشاء حساب فوترة
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV2.SetAccount",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "ActionPlanIds": [],
          "ActionPlansOverwrite": true,
          "ExtraOptions": {
            "AllowNegative": false,
            "Disabled": false
          },
          "ReloadScheduler": true
        }]
      }
    status_code: 200
  register: account_response
```

إضافة الأرصدة:

```
- name: إضافة رصيد البيانات
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Categories": "*any",
          "Balance": {
            "ID": "حزمة البيانات",
            "Value": 10737418240,
            "ExpiryTime": "+720h",
            "Weight": 10
          }
        }]
      }
    status_code: 200
```

تنفيذ الإجراءات:

```
- name: تنفيذ إجراء الشحن
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "APIerSv1.ExecuteAction",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "ActionsId": "Action_Standard_Charge"
        }]
      }
    status_code: 200
```

الحصول على معلومات الحساب:

```
- name: الحصول على تفاصيل الحساب
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV2.GetAccount",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}"
        }]
      }
    status_code: 200
  register: account_info
```

العمل مع ملفات تعريف السمات:

```
- name: الحصول على ملف تعريف السمة
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "APIerSv1.GetAttributeProfile",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "ID": "ATTR_{{ service_uuid }}"
        }]
      }
    return_content: yes
    status_code: 200
  register: attr_response
  ignore_errors: true

- name: استخراج قيمة السمة
  set_fact:
    phone_number: "{{ attr_response.json.result.Attributes |
json_query(\"[?Path=='*req.PhoneNumber'].Value[0].Rules\") | first
}}"
  when: attr_response is defined
```

المنطق الشرطي

التحقق مما إذا كانت المتغيرات موجودة:

```
- name: استخدام القيمة المخصصة أو الافتراضية
  set_fact:
    monthly_cost: "{{ custom_cost | default(50.00) }}"

- name: تشغيل فقط إذا تم تعريف المتغير
  debug:
    msg: "معرف الخدمة هو {{ service_uuid }}"
  when: service_uuid is defined
```

الشروط المنطقية:

```
- name: توفير المعدات
  include_tasks: configure_cpe.yaml
  when: provision_cpe | default(false) | bool

- name: تخطي إذا كان إلغاء التوفير
  assert:
    that:
      - action != "deprovision"
  when: action is defined
```

شروط متعددة:

```
- name: مهمة شرطية معقدة
  uri:
    url: "{{ endpoint }}"
    method: POST
  when:
    - service_uuid is defined
    - customer_id is defined
    - action != "deprovision"
    - enable_feature | default(true) | bool
```

الحلقات والتكرار

حلقات بسيطة:

```
- name: إنشاء أرصدة متعددة
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Account": "{{ service_uuid }}",
          "BalanceType": "{{ item.type }}",
          "Balance": {
            "Value": "{{ item.value }}"
          }
        }]
      }
  loop:
    - { type: "*voice", value: 3600 }
    - { type: "*data", value: 10737418240 }
    - { type: "*sms", value: 100 }
```

التكرار على استجابات API:

```
- name: الحصول على جميع مواقع العملاء
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/site/customer_id/{{
customer_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: sites_response

- name: تكوين المعدات في كل موقع
  debug:
    msg: "تكوين الموقع في {{ item.address_line_1 }}"
  loop: "{{ sites_response.json }}"
```

معالجة الأخطاء

استخدام ignore_errors:

```yaml
- name: إشعا SMS اختياري
  uri:
    url: "http://sms-gateway/send"
    method: POST
    body: {...}
  ignore_errors: true
```

التحقق من صحة الاستجابة:

```yaml
- name: التحقق من استجابة API
  assert:
    that:
      - response.status == 200
      - response.json.result == "OK"
    fail_msg: "فشلت مكالمة API: {{ response.json }}"
```

معالجة الأخطاء الشرطية:

```yaml
- name: محاولة الحصول على الخدمة الحالية
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/service/service_uuid/{{ service_uuid }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: service_lookup
  failed_when: false

- name: إنشاء الخدمة إذا لم تكن موجودة
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/service/"
    method: PUT
    body: {...}
  when: service_lookup.status == 404
```

أفضل الممارسات

تسمية المتغيرات

استخدم أسماء وصفية ومتسقة:

```
# جيد
service_uuid: "SVC_12345"
customer_name: "جون سميث"
monthly_cost: 49.99

# سيء
svc: "SVC_12345"
name: "جون سميث"
cost: 49.99
```

قم ببادئة المتغيرات حسب المصدر:

```
api_response_customer: {...}
api_response_product: {...}
cgr_account_info: {...}
```

التصحيح

طباعة المتغيرات لأغراض استكشاف الأخطاء:

```
- name: طباعة جميع المتغيرات
  debug:
    var: hostvars[inventory_hostname]

- name: طباعة متغير محدد
  debug:
    msg: "معرف الخدمة: {{ service_uuid }}"

- name: طباعة استجابة API
  debug:
    var: api_response_product.json
```

التحقق

تحقق دائمًا من استجابات API الحاسمة:

```
- name: إنشاء حساب
  uri:
    url: "{{ billing_endpoint }}"
    method: POST
    body: {...}
  register: response

- name: التحقق من إنشاء الحساب
  assert:
    that:
      - response.status == 200
      - response.json.result == "OK"
    fail_msg: "فشل إنشاء الحساب: {{ response.json }}"
```

عدم التكرار

صمم المهام لتكون قابلة للتشغيل بأمان مرة أخرى:

```
# تحقق مما إذا كان المورد موجودًا أولاً
- name: التحقق مما إذا كان الحساب موجودًا
  uri:
    url: "{{ ocs_endpoint }}/get_account"
    method: POST
    body: {"Account": "{{ service_uuid }}"}
  register: account_check
  failed_when: false

# إنشاء فقط إذا لم يكن موجودًا
- name: إنشاء حساب
  uri:
    url: "{{ ocs_endpoint }}/create_account"
    method: POST
    body: {...}
  when: account_check.status == 404
```

الأمان

لا تقم بتشفير بيانات الاعتماد:

```
# سيء
mailjet_api_key: "abc123def456"

# جيد - استخدم المتغيرات البيئية
mailjet_api_key: "{{ lookup('env', 'MAILJET_API_KEY') }}"

# جيد - استخدم ملف التكوين
mailjet_api_key: "{{ crm_config.email.api_key }}"
```

استخدم دائمًا HTTPS والمصادقة:

```
- name: استدعاء API خارجي
  uri:
    url: "https://api.example.com/endpoint"
    method: POST
    headers:
      Authorization: "Bearer {{ access_token }}"
    validate_certs: yes
```

الوثائق

وثق المنطق المعقد:

```
# حساب الرسوم النسبية لشهر جزئي
# إذا قام العميل بالتسجيل في اليوم الخامس عشر والفوترة في اليوم الأول،
# قم بفرض 50% من التكلفة الشهرية للأيام المتبقية
- name: حساب الأيام حتى نهاية الشهر
  command: "date -d 'last day of this month' +%d"
  register: days_in_month

- name: الحصول على اليوم الحالي
  command: "date +%d"
  register: current_day

- name: حساب المبلغ النسبي
  set_fact:
    days_remaining: "{{ (days_in_month.stdout | int) -
(current_day.stdout | int) }}"
    pro_rata_cost: "{{ (monthly_cost | float) * (days_remaining |
float) / (days_in_month.stdout | float) }}"
```

اختبار Playbooks

نهج الاختبار

1. **اختبار جاف أولاً**: اختبر مع الأنظمة غير الإنتاجية
2. **تحقق من المتغيرات**: استخدم مهام التصحيح للتأكد من وجود جميع المتغيرات المطلوبة
3. **تحقق من الاستجابات**: تحقق من استجابات API قبل المتابعة
4. **اختبار التراجع**: افشل المهام عمدًا للتحقق من عمل كتل الإنقاذ
5. **اختبار إلغاء التوفير**: اختبر باستخدام `action: "deprovision"` للتحقق من التنظيف

مثال على Playbook اختبار:

```yaml
- name: اختبار توفير الخدمة
  hosts: localhost
  gather_facts: no

  tasks:
    - name: التحقق من المتغيرات المطلوبة
      assert:
        that:
          - product_id is defined
          - customer_id is defined
          - access_token is defined
        fail_msg: "المتغيرات المطلوبة مفقودة"

    - name: اختبار الاتصال بـ API
      uri:
        url: "http://localhost:5000/crm/health"
        method: GET
      register: health_check

    - name: التحقق من فحص الصحة
      assert:
        that:
          - health_check.status == 200
```

الأخطاء الشائعة

التحويلات المفقودة للنوع:

```yaml
# خطأ - قد تكون سلسلة
customer_id: "{{ customer_id }}"

# صحيح - تأكد من أنها عدد صحيح
customer_id: {{ customer_id | int }}
```

عدم التعامل مع المتغيرات غير المعرفة:

```
# خطأ - يفشل إذا لم يكن معرفًا
service_uuid: "{{ service_uuid }}"

# صحيح - قدم افتراضي
service_uuid: "{{ service_uuid | default('') }}"
```

نسيان التحقق:

```
# خطأ - لا يتحقق من الاستجابة
- name: إنشاء حساب
  uri: ...
  register: response

# صحيح - تحقق من الاستجابة
- name: إنشاء حساب
  uri: ...
  register: response

- name: التحقق من الإنشاء
  assert:
    that:
      - response.json.result == "OK"
```

سير عمل التوفير

بشكل عام، سيعمل موظفو Omnitouch مع العميل على:

1. تحديد متطلبات المنتج
2. تطوير Playbooks Ansible اللازمة لأتمتة عملية التوفير
3. اختبار Playbooks في بيئة staging
4. نشرها في الإنتاج

هذا يضمن أن كل خدمة يتم نشرها بشكل متسق وموثوق، مما يقلل من مخاطر الأخطاء ويضمن إكمال جميع الخطوات الضرورية بالترتيب الصحيح.

متغيرات Ansible

تشمل المتغيرات الممررة إلى Playbooks في Ansible:

متغيرات المنتج
مشتقة من تكوينات منتج OmniCRM وتحدد كيفية إعداد الخدمة.

متغيرات المخزون
أو أرقام الهواتف IP، كتل عناوين SIM، محددة من المخزون، تشمل عناصر مثل المودمات، بطاقات المطلوبة للتوفير.

المتغيرات النظامية
تضاف تلقائيًا بواسطة OmniCRM:

- `product_id` - المنتج الذي يتم توفيره
- `customer_id` - العميل الذي يتلقى الخدمة
- `service_id` - الخدمة التي يتم تعديلها (للتعبئة/التغييرات)
- `access_token` - JWT لمصادقة API

إلغاء التوفير

لإلغاء توفير الخدمة **Playbooks Ansible** عندما لم تعد الخدمة مطلوبة، يتم أيضًا استخدام باستخدام نمط كتلة `rescue`. هذا:

- يزيل أي تكوينات
- يحرر المخزون مرة أخرى إلى المجموعة
- يحذف حسابات الفوترة
- يضمن الحفاظ على نظافة النظام

طرق إلغاء التوفير

هناك طريقتان رئيسيتان يتم تحفيز إلغاء التوفير:

1. التراجع التلقائي (فشل التوفير)
عندما يفشل أي مهمة في كتلة التوفير الرئيسية، يتم تنفيذ قسم الإنقاذ تلقائيًا لتنظيف التغييرات الجزئية.

2. إلغاء التوفير اليدوي
تعيين `action: "deprovision"` عند تشغيل Playbook يحفز كتلة الإنقاذ عمدًا لإزالة خدمة.

نمط إلغاء التوفير

هذا الهيكل للتنظيف الآمن OmniCRM في Playbooks تتبع جميع:

```yaml
- name: Playbook توفير الخدمة
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: الكتلة الرئيسية
      block:
        # جميع مهام التوفير تذهب هنا
        - name: إنشاء حساب OCS
          uri: ...

        - name: إنشاء سجل الخدمة
          uri: ...
          register: service_creation_response

        - name: إضافة معاملة
          uri: ...

      rescue:
        # مهام إلغاء التوفير/التراجع تذهب هنا
        - name: طباعة جميع المتغيرات لأغراض التصحيح
          debug:
            var: hostvars[inventory_hostname]

        # تنفيذ مهام التنظيف بترتيب عكسي
        - name: إزالة الحساب في OCS
          uri: ...

        - name: حذف الخدمة من CRM
          uri: ...

        - name: الفشل إذا لم يكن إلغاء التوفير مقصودًا
          assert:
            that:
              - action == "deprovision"
```

مثال كامل لإلغاء التوفير

إليك مثال كامل من `play_simple_service.yaml`:

```
rescue:
  # 1. معلومات التصحيح
  - name: طباعة جميع المتغيرات للتراجع/التنظيف
    debug:
      var: hostvars[inventory_hostname]

  - name: الحصول على تاريخ اليوم
    set_fact:
      today: "{{ lookup('pipe', 'date +%Y-%m-%d') }}"

  # 2. الحصول على معلومات الخدمة إذا كانت متاحة
  - name: لإلغاء API CRM محاولة الحصول على معلومات الخدمة من
التوفير
    uri:
      url: "http://localhost:5000/crm/service/service_id/{{
service_creation_response.json.service_id }}"
      method: GET
      headers:
        Authorization: "Bearer {{ access_token }}"
      return_content: yes
    register: api_response_service
    ignore_errors: True
    when: service_creation_response is defined and
service_creation_response.json is defined and
service_creation_response.json.service_id is defined

  - name: طباعة api_response_service
    debug:
      var: api_response_service
    when: api_response_service is defined

  # 3. تعيين service_uuid للتنظيف
  - name: لإلغاء التوفير تعيين حقائق service_uuid
    set_fact:
      service_uuid: "{{ api_response_service.json.service_uuid }}"
    when:
      - service_uuid is not defined
      - api_response_service is defined
      - api_response_service.json is defined

  # 4. إزالة حساب OCS
  - name: إزالة الحساب في OCS
    uri:
```

```
      url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
      method: POST
      body_format: json
      headers:
        Content-Type: "application/json"
        Authorization: "Bearer {{ access_token }}"
      body:
        {
          "method": "ApierV2.RemoveAccount",
          "params": [{
            "Tenant": "{{ crm_config.ocs.ocsTenant }}",
            "Account": "{{ service_uuid }}",
            "ReloadScheduler": true
          }]
        }
      status_code: 200
    register: response
    ignore_errors: True
    when: service_uuid is defined

  - name: طباعة الاستجابة من إزالة حساب OCS
    debug:
      var: response
    when: response is defined

  # 5. حذف سجل الخدمة من CRM
  - name: إذا تم إنشاؤها CRM حذف الخدمة من
    uri:
      url: "http://localhost:5000/crm/service/service_id/{{
service_creation_response.json.service_id }}"
      method: DELETE
      headers:
        Authorization: "Bearer {{ access_token }}"
      status_code: 200
    register: delete_service_response
    ignore_errors: True
    when: service_creation_response is defined and
service_creation_response.json is defined and
service_creation_response.json.service_id is defined

  - name: طباعة استجابة حذف الخدمة
    debug:
      var: delete_service_response
    when: delete_service_response is defined
```

```
  # 6. تحديد ما إذا كان هذا مقصودًا أم فشلًا
  - name: / تعيين الحالة إلى "نجاح" إذا كان إلغاء التوفير يدويًا
الفشل إذا كان التوفير فشلًا
    assert:
      that:
        - action == "deprovision"
      fail_msg: "فشل ��لتوفير وتم التراجع عنه"
      success_msg: "تم إلغاء توفير الخدمة بنجاح"
```

مفاهيم إلغاء التوفير الرئيسية

ignore_errors: True

تستخدم معظم مهام التنظيف `ignore_errors: True` (على سبيل لأن الموارد قد لا توجد المثال، إذا فشل التوفير قبل إنشائها).

التنفيذ الشرطي

استخدم جمل `when` لتنظيف الموارد فقط التي تم إنشاؤها:

```
when: service_creation_response is defined
```

ترتيب عكسي

يحدث التنظيف بترتيب عكسي لإنشاء:

1. حذف الموارد التابعة أولاً (المعاملات، تعيينات المخزون)
2. حذف سجل الخدمة
3. حذف حساب OCS/الفوترة
4. تحرير أي موارد محتجزة

التحقق النهائي

التحقق النهائي يميز بين:

- **إلغاء التوفير المقصود** (`action == "deprovision"`) → نجاح
- **فشل التوفير** (لم يتم تعيين الإجراء) → فشل

إلغاء التوفير لموارد مختلفة

حسابات OCS/CGRateS:

```
- name: إزالة الحساب في OCS
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV2.RemoveAccount",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "ReloadScheduler": true
        }]
      }
    status_code: 200
  ignore_errors: True
  when: service_uuid is defined
```

سجلات الخدمة في CRM:

```
- name: حذف الخدمة من CRM
  uri:
    url: "http://localhost:5000/crm/service/service_id/{{
service_id }}"
    method: DELETE
    headers:
      Authorization: "Bearer {{ access_token }}"
    status_code: 200
  ignore_errors: True
  when: service_id is defined
```

عناصر المخزون (إعادتها إلى المجموعة):

```
- name: تحرير بطاقة SIM المخزون مجموعة إلى أخرى مرة
  uri:
    url: "http://localhost:5000/crm/inventory/inventory_id/{{
sim_inventory_id }}"
    method: PATCH
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "customer_id": null,
        "service_id": null,
        "item_state": "Available"
      }
    status_code: 200
  ignore_errors: True
  when: sim_inventory_id is defined
```

خطط العمل (الرسوم المتكررة):

```
- name: إزالة خطة العمل من الحساب
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.RemoveActionPlan",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Id": "ActionPlan_Monthly_{{ service_uuid }}_{{
product_id }}"
        }]
      }
    status_code: 200
  ignore_errors: True
```

تفويضات الدفع:

```
- name: تحرير تفويض الدفع
  uri:
    url: "http://localhost:5000/crm/payments/release/{{
authorization_id }}"
    method: POST
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "metadata": {
          "release_reason": "فشل التوفير"
        }
      }
    return_content: yes
  ignore_errors: True
  when: authorization_id is defined
```

سير عمل إلغاء التوفير اليدوي

لإلغاء توفير خدمة يدويًا:

1. تحديد الخدمة المراد إلغاء توفيرها
2. تشغيل Playbook التوفير الأصلي مع `action: "deprovision"`
3. تدخل Playbook في كتلة الإنقاذ على الفور
4. تنفيذ جميع مهام التنظيف
5. يتم إزالة الخدمة بشكل نظيف

مثال على استدعاء API:

```
POST /crm/provision/run_playbook
{
  "product_id": 123,
  "customer_id": 456,
  "service_id": 789,
  "action": "deprovision"
}
```

سيناريوهات التنظيف الجزئي

السيناريو 1: تم إنشاء حساب OCS، فشل إنشاء الخدمة

- يوجد حساب OCS
- لا يوجد سجل خدمة
- يقوم قسم الإنقاذ بإزالة حساب OCS
- لا توجد خدمة لحذفها (تم تخطيها بأمان مع شرط `when`)

السيناريو 2: تم إنشاء الخدمة، فشلت المعاملة

- يوجد حساب OCS
- يوجد سجل خدمة
- لا توجد معاملة
- يقوم قسم الإنقاذ بإزالة كل من حساب OCS والخدمة
- لا توجد معاملة لإبطالها (لم يتم إنشاؤها أبدًا)

السيناريو 3: توفير كامل، فشل التقاط الدفع

- يوجد حساب OCS
- يوجد سجل خدمة
- تم تفويض الدفع ولكن لم يتم التقاطه
- يقوم قسم الإنقاذ:
  - بإطلاق تفويض الدفع (لم يتم تحميل العميل)
  - إزالة سجل الخدمة
  - إزالة حساب OCS

أفضل الممارسات لإلغاء التوفير

1. استخدم ignore_errors بشكل واسع
يجب أن يكون التنظيف متساهلاً. لا تفشل إذا لم يكن المورد موجودًا.

2. تحقق من وجود المورد باستخدام جمل when
حاول التنظيف فقط إذا تم إنشاء المورد:

```
when: service_creation_response is defined
```

3. طباعة معلومات التصحيح
قم دائمًا بتضمين مهام التصحيح للمساعدة في استكشاف الأخطاء في التوفير الفاشل:

```
- name: طباعة جميع المتغيرات لأغراض التصحيح
  debug:
    var: hostvars[inventory_hostname]
```

4. تنظيف بترتيب عكسي حسب الاعتماد
احذف الأطفال قبل الآباء:

- المعاملات قبل الخدمات
- الخدمات قبل حسابات OCS
- تعيينات المخزون قبل تحرير المخزون

5. التعامل مع تفويضات الدفع
قم دائمًا بإطلاق تفويضات الدفع في كتل الإنقاذ لتجنب تحميل العملاء مقابل التوفير الفاشل.

6. إعادة تحميل المجدولين بعد التنظيف
عند إزالة موارد OCS، قم بتضمين `ReloadScheduler: true` لضمان تحديث CGRateS على الفور.

إلغاء التوفير مقابل الحذف

إلغاء التوفير (عبر كتلة الإنقاذ):

- يزيل الخدمة من جميع الأنظمة
- يحرر المخزون
- يلغي الرسوم المتكررة
- يترك أثر تدقيق
- نهج موصى به

الحذف المباشر (عبر API):

- يحذف فقط سجل CRM
- لا ينظف حساب OCS
- لا يحرر المخزون
- يمكن أن يترك موارد يتيمة

- غير موصى به للإنتاج

التراجع ومعالجة الأخطاء

أثناء كل من التوفير وإلغاء التوفير للتعامل مع Ansible في **block/rescue** تستخدم ميزة الأخطاء بشكل سلس. إذا فشلت أي مهمة في أي نقطة أثناء التوفير، يتم تنفيذ قسم الإنقاذ تلقائيًا للتراجع عن التغييرات للعودة إلى حالة متسقة. يضمن ذلك الموثوقية ويقلل من مخاطر النشر الجزئي أو الفاشل.

مثال على معالجة الأخطاء

```
- name: الكتلة الرئيسية
  block:
    - name: الخطوة 1: إنشاء الحساب
      uri: ...
      register: response

    - name: التحقق من الخطوة 1
      assert:
        that:
          - response.status == 200

    - name: الخطوة 2: إنشاء الخدمة
      uri: ...

  rescue:
    # إذا فشلت أي تأكيد أو حدث خطأ في أي مهمة:
    # 1. ينتقل التنفيذ إلى كتلة الإنقاذ
    # 2. يتم تنفيذ مهام التنظيف
    # 3. تفشل Playbook مع رسالة خطأ
    - name: تنظيف التغييرات الجزئية
      uri: ...
```

للحصول على تفاصيل كاملة حول نظام التوفير، وسير العمل، والمصادقة، انظر `concepts_provisioning`.

OmniCRM API

لا توجد وظائف متاحة فقط في واجهة API - متاحة عبر OmniCRM جميع الوظائف داخل المستخدم.

هذا يسمح لك بدمج OmniCRM مع أنظمة أخرى أو أتمتة المهام.

الـ API هو API قائم على REST، ومؤمن باستخدام عدة طرق مصادقة بما في ذلك رموز JWT، مفاتيح API، وقوائم بيضاء لعناوين IP.

تم توثيق الـ API باستخدام Swagger، وهي أداة تسمح بقراءة وفهم واختبار وظائف الـ API بسهولة.

تتوفر وثائق الـ API على عنوان URL التالي:

<https://yourcrm/crm/docs/>

طرق المصادقة

يدعم OmniCRM ثلاث طرق للمصادقة، كل منها مصمم لحالات استخدام مختلفة:

1. **رموز JWT Bearer** - لجلسات المستخدم التفاعلية (واجهة الويب، تطبيقات الجوال)

2. **مفاتيح API** - للتكاملات من خادم إلى خادم وسكربت��ت الأتمتة
3. **قائمة بيضاء لعناوين IP** - للأنظمة الداخلية الموثوقة (خوادم التزويد، أدوات المراقبة)

مصادقة رموز JWT Bearer

هذه هي الطريقة الأساسية للمصادقة لجلسات المستخدم. يقوم المستخدمون بتسجيل الدخول باستخدام البريد الإلكتروني وكلمة المرور، ويتلقون رمز JWT، ويستخدمونه للطلبات اللاحقة.

حالات الاستخدام:

- مصادقة واجهة الويب
- مصادقة تطبيقات الجوال
- الوصول البرمجي قصير الأمد

كيفية المصادقة:

لتسجيل الدخول، أرسل جسم JSON بالهيكل التالي إلى `/crm/auth/login` كطلب POST:

```
{
    "email": "youruser@yourdomain.com",
    "password": "yourpassword"
}
```

سيعيد الـ API كائن JSON يحتوي على حقل `token`، والذي يستخدم لمصادقة جميع الطلبات المستقبلية. بالإضافة إلى ذلك، تتضمن الاستجابة `refresh_token` يمكن استخدامه لتجديد الرمز عند انتهاء صلاحيته، جنبًا إلى جنب مع أذونات المستخدم وأدواره.

يمكنك اختبار ذلك من صفحة Swagger عن طريق اختيار نقطة النهاية `/auth/login`، وملء اسم المستخدم وكلمة المرور، والنقر على زر `Try it out`.

في أعلى يمين صفحة Swagger. لتفويض الجلسة، انسخ قيمة الرمز وانقر على زر "Authorize". ألصق الرمز في حقل "Value"، مسبوقًا بـ `Bearer` وانقر على "Authorize".

الآن، ستتم مصادقة جميع الطلبات اللاحقة باستخدام هذا الرمز.

مصادقة مفتاح API

توفر مفاتيح API مصادقة آمنة وطويلة الأمد للتكاملات من خادم إلى خادم وسكربتات الأتمتة دون الحاجة إلى كلمات مرور المستخدمين.

حالات الاستخدام:

- أنظمة التزويد الآلي
- أدوات المراقبة والتنبيه
- التكامل مع الأنظمة الخارجية
- المهام المجدولة ووظائف cron

كيفية عمل مفاتيح API:

يتم تكوين مفاتيح API في ملف `crm_config.yaml`. كل مفتاح وترتبط بأدوار وأذونات محددة. و سلسلة عشوائية آمنة (حد أدنى 32 حرفًا) تقوم بمصادقة الطلبات عند تمريرها في رأس API `X-API-KEY`.

تكوين مفاتيح API:

يجب إضافة مفاتيح API إلى الخادم `crm_config.yaml` بواسطة مسؤول لديه وصول:

```
api_keys:
  your-secure-api-key-here-minimum-32-chars:
    roles:
      - admin
    description: "Provisioning automation system"
  another-api-key-for-monitoring-system:
    roles:
      - view_customer
      - view_service
    description: "Monitoring and alerting"
```

استخدام مفاتيح API:

قم بتضمين مفتاح API في رأس `X-API-KEY` لطلباتك:

```
curl -X GET "https://yourcrm.com/crm/customers" \
  -H "X-API-KEY: your-secure-api-key-here-minimum-32-chars"
```

مثال على Python:

```
import requests

crm_url = 'https://yourcrm.com'
api_key = 'your-secure-api-key-here-minimum-32-chars'

headers = {
    "Content-Type": "application/json",
    "X-API-KEY": api_key
}

# Get Customers
response = requests.get(crm_url + '/crm/customers',
headers=headers)
for customer in response.json()['data']:
    print(customer)
```

أفضل الممارسات:

- قم بإنشاء مفاتيح API باستخدام مولدات عشوائية آمنة من الناحية التشفيرية (`openssl rand -base64 48`)
- استخدم مفاتيح API مختلفة للأنظمة المختلفة
- وثق الغرض من كل مفتاح API في حقل `description`
- قم بتدوير مفاتيح API بشكل دوري
- لا تقم أبدًا بالتزام مفاتيح API في التحكم في الإصدارات
- خصص الحد الأدنى من الأذونات اللازمة لكل مفتاح API

مصادقة قائمة بيضاء لعناوين IP

تسمح قائمة بيضاء لعناوين IP لعناوين IP المحددة بالوصول إلى الـ API دون مصادقة. هذا مفيد للأنظمة الداخلية الموثوقة على الشبكات الخاصة.

حالات الاستخدام:

- خوادم التزويد الداخلية
- أنظمة مراقبة الشبكة على VLANs الإدارية
- برامج Ansible التي تعمل على بنية تحتية مسيطر عليها

تكوين قائمة بيضاء لعناوين IP:

أضف عناوين IP الموثوقة إلى `crm_config.yaml`:

```
ip_whitelist:
  - 192.168.1.100
  - 10.0.0.0/24
  - 172.16.50.10
```

اعتبارات الأمان:

- استخدم قائمة بيضاء لعناوين IP فقط على الشبكات الخاصة والمحمية
- لا تقم أبدًا بإدراج عناوين IP العامة في القائمة البيضاء
- استخدم أكثر نطاقات IP تحديدًا ممكنة
- وثق سبب إدراج كل IP في القائمة البيضاء
- قم بمراجعة عناوين IP المدرجة في القائمة البيضاء بانتظام

أمثلة على استدعاءات API باستخدام Python

إليك مثال على كيفية تسجيل الدخول واسترجاع قائمة العملاء باستخدام مصادقة رمز JWT:

```python
import requests

crm_url = 'https://yourcrm.com'
session = requests.Session()

print("Provisioning data to server: " + str(crm_url))

headers = {
    "Content-Type": "application/json"
}

# Get Auth Token
response = session.post(crm_url + '/crm/auth/login', json={
    "email": "youruser@yourdomain.com",
    "password": "yourpassword"
}, headers=headers)

print(response.status_code)
print(response.json())
assert response.status_code == 200

headers['Authorization'] = 'Bearer ' + response.json()['token']
print("Authenticated to CRM successfully")

# Get Customers
response = session.get(crm_url + '/crm/customers',
headers=headers)
for customer in response.json()['data']:
    print(customer)
```

مراقبة أداء API

يتم تتبع جميع طلبات API تلقائيًا مع مقاييس شاملة. للحصول على تفاصيل حول مقاييس أداء API بما في ذلك معدلات الطلبات، وأوقات الاستجابة، ومعدلات الأخطاء، وإحصائيات محددة لنقاط

النهاية، انظر Monitoring & Metrics.

نقطة نهاية المقاييس متاحة على `/crm/metrics` بتنسيق Prometheus للمراقبة والتنبيه.

الوثائق ذات الصلة

- نظام العمارة - تصميم النظام العام وعلاقات المكونات
- المراقبة والمقاييس - مرجع كامل للمقاييس بما في ذلك مقاييس API، وقاعدة البيانات، والتكامل
- تدفقات المصادقة - تدفقات المصادقة والتفويض التفصيلية
- RBAC - التحكم في الوصول القائم على الأدوار والأذونات

ملاحظات شحن خدمة / منتج CRM

ملاحظة ⓘ

للحصول على دليل شامل يغطي تعريف المنتج، التوفير، الإضافات، وإلغاء التوفير مع أمثلة أنسبل مفصلة واستراتيجية التسعير، انظر دليل دورة حياة المنتج الكامل.

للحصول على معلومات مفصلة حول توفير خدمات الهاتف المحمول باستخدام بطاقات SIM، انظر توفير بطاقة SIM.

نظرة عامة على المنتجات والخدمات

المنتج (عنصر القائمة):

المنتج هو مثل طبق محدد في قائمة مطعم، مثل "سباغيتي كاربونارا."

لديه وصف واضح، قائمة بالمكونات (مثل المعكرونة، الكريمة، البيض، الجبن، واللحم المقدد)، وسعر.

يحتوي المنتج بالمثل على تفاصيل ما هو متضمن: الميزات، المواصفات، في OmniCRM، والتسعير.

غالبًا، قد يرغب ال��ملاء في تعديلات، مثل "لا بصل" أو "أضف جبن إضافي" إلى وجبتهم. داخل OmniCRM، يتوافق هذا مع تخصيص خدمة قبل إنشائها. مستوى التخصيصات أو التعديلات على الخدمة يعود لك (المشغل) لتحديده.

في OmniCRM، قد يقوم العملاء أو الموظفون بتعديل منتج ليتناسب بشكل أفضل مع احتياجات عميل معين، مثل ترقية سرعة الإنترنت الخاصة بهم أو إضافة ميزات معينة. يتم عكس هذا التخصيص في الخدمة المحددة المقدمة.

المنتج هو في الأساس عرض يمكن للعملاء اختياره للطلب منه، مشابه لقراءة واختيار طبق من القائمة.

كتالوج المنتجات (قائمة المطعم):

كتالوج المنتجات هو مثل القائمة الكاملة في مطعم، والتي تسرد جميع الأطباق المتاحة، من المقبلات إلى الحلويات.

إنه المجموعة الكاملة لكل ما يقدمه المطعم (أو في حالتك، مزود الخدمة).

في سياق الأعمال، يوفر كتالوج المنتجات للعملاء جميع المنتجات المتاحة، حتى يتمكنوا من اختيار ما يناسب احتياجاتهم.

الخدمة (الطبق المُعد):

عندما يطلب عميل عنصرًا من القائمة، يتم إعداد الطبق في المطبخ. هذا يشبه إنشاء خدمة من منتج.

في OmniCRM، عندما يختار العميل منتجًا، يتم إنشاء نسخة من ذلك المنتج وتسليمه كخدمة.

يتم تخصيصه وإعداده خصيصًا لذلك العميل، تمامًا مثل وجبة معدة لزبون.

على سبيل المثال، عندما يختار شخص ما "خطة الإنترنت البرونزية" من كتالوج المنتجات، يقوم أي، يتم :(المودمات والمنافذ، IP عناوين) نظام التوفير "بطهي" نسخة من تلك الخطة من المكونات تفعيل الخطة وتسليمها إلى العميل المحدد.

المنتجات المجمعة (وجبات كومبو):

قد يقدم كتالوج المنتجات أيضًا حزم، مثل وجبة كومبو تتضمن مقبلات، طبق رئيسي، وحلوى معًا بسعر خاص.

في OmniCRM، تجمع المنتجات المجمعة بين عدة منتجات فردية في حزمة مريحة واحدة، مثل "حزمة أساسيات المنزل" التي تشمل خدمات الإنترنت، الكابل، والهاتف بسعر مخفض.

بمجرد اختيارها، يتم تحويل هذه الحزمة إلى خدمات متعددة مصممة خصيصًا للعميل.

تعريفات المنتجات

المنتج هو نموذج يُستخدم لإنشاء خدمة / إضافة / خصم / إضافة، إلخ.

داخل التعريف ندرج:

- معلومات حول المنتج (الميزات، المكونات، الشروط والأحكام، مدة العقد، الأيقونة، إلخ) التي يتم عرضها على مستخدم CRM (العميل أو *الموظف*).
- المنطق التجاري حول من يمكنه شراء المنتج (*تجاري* أو *سكني*)، إذا كان يعتمد على وجود خدمة رئيسية مُقدمة (مثل الإضافات المحمولة المتاحة فقط للعملاء الذين لديهم خدمة محمولة)، إذا كان يمكن طلبه مباشرة من قبل عميل عبر الرعاية الذاتية أو فقط من قبل وكيل خدمة العملاء، ومتى يمكن شراء المنتج (السماح بوجود منتج متاح فقط لفترة زمنية محددة).

- يتم تحديدها (SIM مثل المودمات أو بطاقات) عندما يجب تضمين عناصر المخزون ورقم هاتف SIM كقائمة عناصر المخزون، على سبيل المثال الخدمة أدناه تتطلب بطاقة ليتم تعيينهما:

  `['SIM Card', 'Phone Number']` هذه تتوافق مع عناصر المخزون المحددة في CRM.

- الإشارة إلى كتاب أنسبل لتوفير الخدمة لعب توفير بالإضافة إلى المتغيرات التي يجب تمريرها إلى أنسبل. هذه المتغيرات التي يجب تمريرها سحرية، حيث قد تكون متغيرات التي يتم تعريفها بواسطة المنتج الذي نضيفه إليه، أو قد تكون مثل `service_id` مثل `MSISDN` و `ICCID` حيث قمنا باختيار عناصر المخزون التي يتم تمريرها عند تعيين المخزون. يتم التعامل مع التجميع في لعب التوفير لاحتواء خدمات متعددة، على سبيل المثال، قد يوفر منتج الإنترنت المنزلي المجمّع TV & Voice خدمة لكل منها.

فئات المنتجات وأنواع الخدمات

تستخدم المنتجات حقلين تصنيفيين للمساعدة في تنظيم وتصفية العروض:

فئات المنتجات

يتحكم حقل `category` في مكان عرض المنتجات في واجهة المستخدم. تشمل القيم الشائعة:

- **مستقل** - يظهر كخيار خدمة أساسي عند إنشاء خدمة جديدة
- **إضافة** - يظهر عند الإضافة إلى خدمة موجودة

- **حزمة** - يظهر كخيار خدمة مجمعة (يتم توفيرها مثل إضافة إلى الخدمات الموجودة)
- **ترويج** - عروض ترويجية خاصة

هذه الفئات تنظيمية بحتة ولا تحدد ما يتم توفيره. يتم تحديد سلوك التوفير الفعلي بالكامل بواسطة كتاب أنسبل المشار إليه في `provisioning_play`.

على سبيل المثال: - منتج `مستقل` عادةً ما ينشئ كائن خدمة جديدة - منتج `إضا فة` أو `حزمة` يضاف عادةً إلى خدمة موجودة - لكن هذا يعود إلى المنفذ الذي يكتب الكتابة - يمكنك إنشاء عدة كائنات خدمة من إضافة، أو تعديل الخدمات الموجودة من منتج مستقل إذا لزم الأمر

تتحكم الفئة ببساطة في تدفق واجهة المستخدم وأين يرى العملاء/الموظفون خيار المنتج.

أنواع الخدمات

يصنف حقل `service_type` نوع الخدمة المقدمة.

هذه محددة بالكامل بواسطة المستخدم، لكن القيم الشائعة تشمل:

- **محمول** - خدمات الهاتف المحمول مع الصوت، الرسائل القصيرة، والبيانات
- **ثابت** - خدمات الإنترنت الثابتة اللاسلكية أو السلكية
- **صوت ثابت** - خدمات الصوت الثابتة (الخط الثابت، VoIP)
- **نقطة الاتصال** - أجهزة نقطة اتصال محمولة أو مستأجرة
- **دونجل** - خدمات مودم USB أو دونجل
- **صوت** - خدمات صوتية فقط
- **بيانات** - خدمات بيانات فقط

مثل الفئات، يمكن تخصيص أنواع الخدمات بناءً على عروضك. تساعد في:

- تصفية أي إضافات تنطبق على أي خدمات أساسية
- تنظيم المنتجات في بوابة العملاء
- مطابقة متطلبات المخزون
- تحديد سير العمل للتوفير

مثال: يمكن للعميل الذي ل��يه خدمة `محمول` رؤية إضافات المحمول، بينما يرى العميل الذي لديه خدمة `ثا بت` إضافات الخط الثابت.

إدارة المنتجات

تتم إدارة المنتجات من خلال صفحة إدارة المنتجات، حيث يمكنك عرض، البحث، تصفية، وتحرير جميع المنتجات المتاحة.

واجهة نموذج المنتج

النقر على أي منتج يفتح واجهة محسّنة مع علامات تبويب تنظم جميع إعدادات المنتج في مجموعات منطقية لتسهيل التنقل والتحرير.

تتميز واجهة نموذج إدارة المنتج بخمس علامات تبويب منظمة:

1. **المعلومات الأساسية** - المعلومات الأساسية للمنتج (الاسم، الشعار، الفئة، الأيقونة، الميزات، الشروط)
2. **التسعير** - جميع الحقول المتعلقة بالتكلفة بما في ذلك التكاليف المتكررة، تكاليف الإعداد، ونسبة الضريبة
3. **التكوين** ⚙ - إعدادات التجديد، أنواع العملاء، والاعتمادات
4. **التوفير** - إعدادات كتاب أنسبل ومتطلبات المخزون
5. **التوفر** - نطاقات التاريخ والطوابع الزمنية للنظام

تنظيم علامة تبويب التسعير:

تجمع علامة تبويب التسعير حقول التكلفة في أقسام منطقية:

- **التكاليف المتكررة** - تكاليف التجزئة والجملة الشهرية جنبًا إلى جنب
- **تكاليف الإعداد** - رسوم التفعيل أو الإعداد لمرة واحدة للتجزئة والجملة
- **الضريبة** - إعداد نسبة الضريبة مع حساب تلقائي

ميزات وضع التحرير:

- **اختيار الأيقونة** - البحث واختيار أيقونات FontAwesome بصريًا
- **اختيار عناصر المخزون** - اختيار من أنواع عناصر المخزون المتاحة
- **اختيار التاريخ/الوقت** - اختيار سهل لنوافذ التوفر
- **تنسيق العملة** - تلقائي $ بادئة لحقول التكلفة
- **محددات منسدلة** - خيارات محددة مسبقًا للفئات والحقول المنطقية

اختيار الأيقونة:

عند تحرير حقل الأيقونة، تظهر واجهة اختيار أيقونة قابلة للبحث تتيح لك تصفح واختيار بصريًا من آلاف أيقونات FontAwesome.

الميزات: * البحث عن الأيقونات حسب الكلمة الرئيسية (مثل "مفتاح"، "محمول"، "واي فاي") * معاينة مظهر الأيقونة في الوقت الحقيقي * عرض اسم فئة الأيقونة للرجوع إليها * اختيار منسدل للوصول السريع

علامة تبويب التكوين:

تنظم علامة تبويب التكوين إعدادات سلوك المنتج في مجموعات منطقية.

أقسام التكوين:

- **إعدادات التجديد:**
  - التجديد التلقائي - سلوك التجديد الافتراضي (مطالبة/نعم/لا)
  - السماح بالتجديد التلقائي - ما إذا كان يمكن للعملاء تمكين التجديد التلقائي
  - أيام العقد - الحد الأدنى لطول العقد (مثل 30 شهراً، 365 سنوياً)
- **أنواع العملاء:**
  - سكني - متاح للعملاء المستهلكين
  - تجاري - متاح للعملاء التجاريين
- **الاعتمادات:**
  - يعتمد على القائمة - معرفات المنتجات أو أنواع الخدمات المطلوبة قبل إضافة هذا المنتج
  - يستخدم للاعتمادات الإضافية (مثل، تتطلب الإضافات المحمولة خدمة محمولة نشطة)

علامة تبويب التوفير:

تتعامل علامة تبويب التوفير مع أتمتة أنسبل ومتطلبات المخزون.

حقول التوفير:

- **لعب التوفير:**
  - اسم كتاب أنسبل (بدون امتداد .yaml)
  - يجب أن يكون موجودًا في دليل OmniCRM-API/Provisioners/plays/
  - يتم استدعاؤه عند إنشاء الخدمة أو تحديثه�أو إلغاء توفيرها
- **متغيرات JSON للتوفير:**
  - المتغيرات الافتراضية المرسلة إلى كتاب أنسبل كـ JSON
  - يمكن تجاوزها أثناء التوفير
  - يتلقى الكتاب هذه بالإضافة إلى customer_id، product_id، service_id، access_token
- **قائمة عناصر المخزون:**
  - محدد متعدد يظهر أنواع عناصر المخزون المتاحة
  - أمثلة: بطاقة SIM، عنوان IPv4، رقم الهاتف، مودم راوتر
  - يختار العميل/الموظف عناصر محددة من المخزون المتاح أثناء الطلب
  - يتم تمرير معرفات المخزون المحددة إلى الكتاب مع نوع المخزون كاسم المتغير

علامة تبويب التوفر:

تتحكم علامة تبويب التوفر في متى يمكن شراء المنتج وتعرض بيانات التعريف النظامية.

إعدادات التوفر:

- **متاح من:**
  - التاريخ/الوقت عندما يصبح المنتج متاحًا للشراء
  - اتركه فارغًا للتوفر الفوري
  - مفيد للإعلان المسبق عن المنتجات الجديدة
- **متاح حتى:**
  - التاريخ/الوقت عندما لم يعد المنتج متاحًا للشراء
  - اتركه فارغًا للتوفر غير المحدود
  - مثالي للعروض المحدودة الوقت أو المنتجات التي انتهت صلاحيتها
- **بيانات التعريف النظامية (للقراءة فقط):**
  - تم الإنشاء - الطابع الزمني عندما تم إنشاء المنتج لأول مرة
  - آخر تعديل - الطابع الزمني لأحدث تحديث
  - يتم الحفاظ عليه تلقائيًا بواسطة النظام

إجراءات النموذج:

- **وضع العرض:**
  - إغلاق - إلغاء النموذج
  - نسخ المنتج - إنشاء نسخة مع لاحقة "_clone"
  - تحرير المنتج - التبديل إلى وضع التحرير
- **وضع التحرير/الإنشاء:**
  - إلغاء - discard التغييرات وإغلاق
  - حفظ التغييرات - إنشاء أو تحديث المنتج (زر كبير للتأكيد)

حقول المنتج

يحتوي نموذج المنتج على جميع المعلومات اللازمة لتعريف عرض وكيف يجب أن يتم توفيره. يتم إدارة هذه الحقول من خلال واجهة نموذج إدارة المنتج الموضحة أعلاه.

المعلومات الأساسية

- **product_id** - معرف فريد يتم تعيينه تلقائيًا بواسطة النظام
- **product_name** - الاسم المعروض للعملاء والموظفين في واجهة المستخدم
- **product_slug** - معرف فريد يُستخدم في عناوين URL واستدعاءات API (حروف صغيرة، بدون مسافات، استخدم الشرطات)
- **category** - يتحكم في مكان ظهور هذا المنتج في واجهة المستخدم:
  - `مستقل` - يظهر كخيار خدمة أساسية عند إنشاء خدمة جديدة
  - `إضا فة` - يظهر عند الإضافة إلى خدمة موجودة
  - `حزمة` - يظهر كخيار خدمة مجمعة
  - `ترو يج` - عروض ترويجية خاصة
- **service_type** - نوع الخدمة المقدمة (مث��، محمول، ثابت، صوت ثابت، نقطة اتصال، دونجل، صوت، بيانات). يُستخدم لتصفية أي إضافات تنطبق على أي خدمات.
- **comment** - ملاحظات داخلية حول المنتج للرجوع إليها من قبل الموظفين فقط (لا تظهر للعملاء)
- **icon** - فئة أيقونة FontAwesome المعروضة في واجهة المستخدم (مثل، `fa-solid fa-sim-card`)

حقول التسعير

- **retail_cost** - الرسوم الشهرية المتكررة المفروضة على العميل (تعيين إلى 0 للشراء لمرة واحدة أو المنتجات المدفوعة مسبقًا)
- **wholesale_cost** - التكلفة الشهرية الخاصة بك لتوفير هذه الخدمة (تُستخدم لحساب الهوامش)
- **retail_setup_cost** - رسوم التفعيل أو الإعداد لمرة واحدة المفروضة على العميل
- **wholesale_setup_cost** - التكلفة لمرة واحدة الخاصة بك لإعداد الخدمة
- **tax_percentage** - نسبة الضريبة المطبقة على هذا المنتج (مثل، 10 لـ 10%، 12.5 لـ 12.5%). تعيين إلى 0 للمنتجات المعفاة من الضريبة. يتم تطبيق معدل الضريبة هذا تلقائيًا على المعاملات التي تم إنشاؤها من هذا المنتج.

تطبيق الضريبة:

عند إنشاء معاملة من هذا المنتج، يتم نسخ نسبة الضريبة تلقائيًا إلى المعاملة ويتم حساب مبلغ الضريبة. على سبيل المثال:

- منتج مع ضريبة 10%، $50.00 تكلفة التجزئة → المعاملة لديها $5.00 ضريبة
- منتج مع ضريبة 0% (معفى من الضريبة) → المعاملة لديها $0.00 ضريبة
- تجاوز المعاملة اليدوية → يمكن للموظف تغيير نسبة الضريبة لكل معاملة

رؤية العميل والوصول

- **enabled** - ما إذا كان هذا المنتج نشطًا ومتاحة للشراء (تعيين إلى false لإخفاء دون حذف)
- **residential** - ما إذا كان يمكن للعملاء السكنيين (المستهلكين) شراء هذا المنتج
- **business** - ما إذا كان يمكن للعملاء التجاريين (التجاريين) شراء هذا المنتج
- **customer_can_purchase** - ما إذا كان يمكن للعملاء الشراء الذاتي عبر البوابة (false) أو إذا كان يمكن للموظفين فقط إضافته (true)
- **available_from** - التاريخ/الوقت عندما يصبح هذا المنتج متاحًا للشراء (اختياري)
- **available_until** - التاريخ/الوقت عندما لم يعد هذا المنتج متاحًا للشراء (اختياري، مفيد للعروض المحدودة الوقت)

العقد والتجديد

- **contract_days** - الحد الأدنى لطول العقد بالأيام (مثل، 30 شهراً شهريًا، 365 سنويًا، 0 بدون حد أدنى للعقد)

- **auto_renew** - سلوك التجديد الافتراضي:
  - `prompt` - يسأل العميل في كل مرة ما إذا كان يريد التجديد
  - `true` - يتم التجديد تلقائيًا دون سؤال
  - `false` - يتطلب التجديد اليدوي
- **allow_auto_renew** - تعيين إلى) ما إذا كان يمكن للعملاء تمكين التجديد التلقائي false للشراء لمرة واحدة)

المحتوى الموجه للعملاء

- **terms** - الشروط والأحكام المعروضة للعملاء قبل الشراء (تتضمن القيود، قواعد انتهاء الصلاحية، شروط الاستخدام)
- **features_list** - قائمة الميزات والمكونات المعروضة للعملاء (تنسيق قائمة بايثون: `['الميزة 1', 'الميزة 2']`)

تكوين التوفير

- **provisioning_play** - اسم كتاب أنسبل الذي يوفر هذه الخدمة (بدون امتداد .yaml). يجب أن يكون موجودًا في `OmniCRM-API/Provisioners/plays/` الدليل.
- **provisioning_json_vars** - المتغيرات الافتراضية المرسلة إلى كتاب أنسبل كـ JSON. يمكن تجاوزها عند التوفير. يتلقى الكتاب هذه مع `customer_id`، `product_id`، `service_id`، و `access_token`.
- **inventory_items_list** - قائمة عناصر المخزون المطلوبة لهذا المنتج (مثل، `['SIM Card', 'Mobile Number']`). عندما يطلب العميل ، سيُطلب منه اختيار عناصر معينة من المخزون المتاح. يتم تمرير معرفات المخزون المحددة إلى كتاب التوفير مع نوع المخزون كاسم المتغير.
- **relies_on_list** - قائمة معرفات المنتجات أو أنواع الخدمات التي يجب أن تكون موجودة قبل إضافة هذا المنتج. تُستخدم للاعتمادات الإضافية (مثل، تتطلب الإضافات المحمولة وجود خدمة محمولة نشطة).

بيانات التعريف النظامية

- **created** - الطابع الزمني عندما تم إنشاء المنتج (يتم تعيينه تلقائيًا)
- **last_modified** - الطابع الزمني عندما تم تحديث المنتج آخر مرة (يتم تحديثه تلقائيًا)

أمثلة تعريفات المنتجات

منتج مستقل (محمولة SIM بطاقة)

```
{
  "product_id": 1,
  "product_slug": "Mobile-SIM",
  "product_name": "بطاقة SIM محمولة فقط",
  "category": "مستقل",
  "service_type": "محمول",
  "provisioning_play": "play_psim_only",
  "provisioning_json_vars": "{\"iccid\": \"\", \"msisdn\": \"\"}",
  "inventory_items_list": "['SIM Card', 'Mobile Number']",
  "retail_cost": 0,
  "retail_setup_cost": 0,
  "wholesale_cost": 3,
  "wholesale_setup_cost": 1,
  "contract_days": 0,
  "residential": true,
  "business": true,
  "enabled": true,
  "customer_can_purchase": true,
  "icon": "fa-solid fa-sim-card",
  "features_list": "['رقم هاتف أسترالي (04xxx)', 'أسرع السرعات', 'تغطية أفضل على البر الرئيسي والتجوال']",
  "terms": "يجب تفعيله خلال 6 أشهر. يتم فقدان جميع الرصيد إذا لم يتم استخدام الخدمة لمدة 12 شهرًا.",
  "comment": "بطاقة SIM المحمولة فعلية للاستخدام مع الهواتف المحمولة"
}
```


وينشئ خدمة جديدة (ورقم الهاتف SIM بطاقة) يتطلب هذا المنتج المستقل عنصرين من المخزون عند توفيره.

منتج إضافة (خطة بيانات شهرية)


```
{
  "product_slug": "example-mobile-prepaid-mini",
  "product_name": "خطة موبايل مصغرة مثال",
  "category": "إضافة",
  "service_type": "محمول",
  "provisioning_play": "play_topup_charge_then_action",
  "provisioning_json_vars": "",
  "inventory_items_list": "[]",
  "retail_cost": 30,
  "retail_setup_cost": 0,
  "wholesale_cost": 5.84,
  "contract_days": 30,
  "residential": true,
  "business": false,
  "enabled": true,
  "customer_can_purchase": true,
  "auto_renew": "prompt",
  "icon": "fa-solid fa-sim-card",
  "features_list": "['8GB مكالمات ', 'من البيانات فائقة السرعة
ورسائل نصية غير محدودة لمستخدمي موبا️ل مثال', '100 دقيقة مكالمات
إلى أستراليا', '100 رسالة نصية إلى أستراليا', 'انتهاء صلاحية 30
يوم']",
  "terms": "تنتهي صلاحية الرصيد بعد 30 يومًا. بمجرد استخدام البيانات
أو الصوت أو الرسائل النصية، ستحتاج إلى إعادة الشحن للاستمرار في
استخدام الخدمة.",
  "comment": "أصغر خطة لدينا للمستخدمين الخفيفين"
}
```


لا يتطلب هذا المنتج الإضافي مخزونًا ويتم تطبيقه على خدمة موجودة. يقوم بشحن العميل ويضيف أرصدة/رصيد إلى خدمتهم.

منتج حزمة (حزمة كبار السن)


```
{
  "product_slug": "Bundle-Seniors",
  "product_name": "حزمة كبار السن",
  "category": "حزمة",
  "service_type": "ثابت",
  "provisioning_play": "play_seniors_package",
  "provisioning_json_vars": "{\"IPTV_Service_ID\":
\"SeniorBundle\"}",
  "inventory_items_list": "['مودم راوتر']",
  "retail_cost": 30,
  "retail_setup_cost": 0,
  "wholesale_cost": 10,
  "wholesale_setup_cost": 11,
  "contract_days": 180,
  "residential": true,
  "business": false,
  "enabled": true,
  "icon": "fa-solid fa-person-walking-with-cane",
  "features_list": "['رفع ','تنزيل بسرعة 20 ميغابت في الثانية
,'بسرعة 5 ميغابت في الثانية','بيانات غير محدودة','صوت منزلي
','رسوم تركيب بقيمة 60£ +5£ شهريًا','إضافي: التلفزيون'']",
  "terms": "عقد لمدة 6 أشهر، يجب إظهار بطاقة المواطن المسن
للتأهل",
  "comment": "خدمة GPON بسرعة 20 ميغابت/2 ميغابت + IPTV + هاتف"
}
```


يوفر هذا المنتج المجمّع خدمات متعددة (هاتف + IPTV + إنترنت) باستخدام كتاب واحد. يتطلب عنصرًا واحدًا من المخزون (مودم راوتر).

منتج إضافة (إعادة شحن بسيطة)

```
{
  "product_slug": "Mobile-Topup-5",
  "product_name": "إعادة شحن PAYG 5£ بقيمة",
  "category": "إضافة",
  "service_type": "محمول",
  "provisioning_play": "play_topup_monetary",
  "provisioning_json_vars": "{\"service_id\": \"\"}",
  "inventory_items_list": "[]",
  "retail_cost": 5,
  "retail_setup_cost": 0,
  "wholesale_cost": 0,
  "contract_days": 0,
  "residential": true,
  "business": false,
  "enabled": true,
  "customer_can_purchase": true,
  "icon": "fa-solid fa-coins",
  "features_list": "['رصيد بقيمة 5£', 'صالح لمدة 180 يومًا']",
  "terms": "صالح لمدة 180 يومًا أو حتى يتم استخدام جميع الرصيد.
راجع موقعنا للحصول على الأسعار الكاملة",
  "comment": "للاستخدام في المكالمات، الرسائل النصية والبيانات 5£"
}
```

تضيف هذه الإضافة ببساطة رصيدًا نقديًا إلى خدمة موجودة. لا يتطلب مخزونًا، ويستخدم `service_id` لتحديد الخدمة التي يجب إعادة شحنها.

كيفية تمرير المتغيرات إلى أنسبل

فهم كيفية تدفق المتغيرات من تعريف المنتج عبر API إلى كتاب أنسبل أمر حاسم لكتابة كتب توفير فعالة.

مصادر المتغيرات والدمج

عند إنشاء وظيفة توفير، تأتي المتغيرات من مصادر متعددة وتُدمج معًا بهذا الترتيب (المصادر اللاحقة تتجاوز السابقة):

1. **متغيرات provisioning_json_vars للمنتج** - المتغيرات الافتراضية من تعريف المنتج
2. **جسم الطلب** - المتغ��رات المرسلة في استدعاء API (يمكن أن تتجاوز الافتراضات الخاصة بالمنتج)
3. **المتغيرات التي أضافها النظام** - تتم إضافتها تلقائيًا بواسطة نظام التوفير
4. **اختيارات المخزون** - معرفات عناصر المخزون المحددة (إذا لم يكن `inventory_items_list` فارغًا)

عملية دمج المتغيرات

يعمل النظام على دمج المتغيرات من جميع المصادر، مع تجاوز المصادر اللاحقة للمصادر السابقة. وهذا يسمح بتخصيص مرن في وقت التوفير.

على سبيل المثال، إذا كان منتجك يحتوي على:

```
"provisioning_json_vars": "{\"monthly_cost\": 50, \"data_gb\": 100}"
```

وكان طلب API الخاص بك يتضمن:

```
{
  "product_id": 10,
  "customer_id": 456,
  "monthly_cost": 45,
  "custom_param": "value"
}
```

ستكون المتغيرات النهائية `extra_vars` المرسلة إلى أنسبل:

```
{
    "monthly_cost": 45,          # تم تجاوزها من الطلب
    "data_gb": 100,              # من provisioning_json_vars
    "product_id": 10,            # من الطلب
    "customer_id": 456,          # من الطلب
    "custom_param": "value",     # من الطلب
    "access_token": "eyJ..."     # أضيفت بواسطة النظام
}
```

المتغيرات التي أضافها النظام

يضيف نظام التوفير تلقائيًا:

- `access_token` - رمز JWT لمصادقة استدعاءات API مرة أخرى إلى CRM (مقدم لمصادقة `refresh_token` أو تم إنشاؤه من API، مفتاح/IP مباشرة لمصادقة المستخدم)
- `initiating_user` - معرف المستخدم الذي قام بتفعيل التوفير (أو أول مسؤول للأنظمة الآلية)
- أي حقول من جسم الطلب (`product_id`، `customer_id`، `service_id`، إلخ.)

متغيرات المخزون

عندما يتطلب منتج عناصر مخزون (مثل، `inventory_items_list: "['SIM Card', 'Mobile Number']"`)، تعمل العملية كما يلي:

1. **تطلب الاختيار/واجهة المستخدم API** - يختار المستخدم عناصر المخزون المحددة من المخزون المتاح
2. **تتم إضافة معرفات المخزون إلى المتغيرات** - تتم إضافة معرفات عناصر المخزون المحددة مع نوع المخزون كاسم المتغير
3. **يمكن للكتاب الوصول إلى معرفات المخزون** - يمكن لكتاب التوفير بعد ذلك استرداد تفاصيل المخزون الكاملة من API CRM

على سبيل المثال، إذا اختار المستخدم: - بطاقة SIM مع inventory_id: 789 - رقم الهاتف مع inventory_id: 101

تتضمن المتغيرات المرسلة إلى الكتاب: - `SIM Card`: 789 - `Mobile Number`: 101

يمكن للكتاب بعد ذلك استخدام هذه المعرفات لاسترداد سجلات المخزون الكاملة (ICCID، IMSI، MSISDN، إلخ.) من API CRM واستخدام تلك المعلومات لتوفير الخدمة على معدات الشبكة.

كيفية استلام أنسبل المتغيرات

يمرر نظام التوفير جميع المتغيرات المدمجة إلى كتاب أنسبل كـ `extravars`. داخل الكتاب، تتوفر هذه المتغيرات من خلال نظام المتغيرات القياسي لأنسبل ويمكن استخدامها في المهام.

يمكن الإشارة إلى المتغيرات مباشرة في مهام الكتاب باستخدام صيغة `{{ variable_name }}`. على سبيل المثال، `{{ product_id }}`، `{{ customer_id }}`، `{{ monthly_cost`

`}}`، إلخ.

المتغيرات المرسلة إلى المنتجات الإضافية

عند توفير منتج إضافي، يمرر النظام تلقائيًا:

- `product_id` - معرف المنتج الإضافي الذي يتم توفيره
- `customer_id` - العميل الذي يمتلك الخدمة
- `service_id` - معرف الخدمة التي تتم إضافة هذا الإضافة إليها (حاسم للإضافات)
- `access_token` - رمز المصادقة لاستدعاءات API
- أي متغيرات من `provisioning_json_vars`
- أي متغيرات إضافية من طلب API

مثال على تدفق توفير الإضافة

عندما يضيف عميل إضافة "إعادة شحن بقيمة 5£" إلى خدمة الهاتف المحمول الخاصة به (service_id: 123)، يتلقى الكتاب المتغيرات بما في ذلك:

- `product_id`: 45 (منتج إعادة الشحن)
- `customer_id`: 456 (العميل)
- `service_id`: 123 (الخدمة التي يجب إضافة الرصيد إليها)
- `access_token`: رمز المصادقة
- `provisioning_json_vars` للمنتج بالإضافة إلى أي متغيرات من

يستخدم الكتاب بعد ذلك هذه المتغيرات لـ:

1. **استرداد تفاصيل الخدمة** من API CRM باستخدام `service_id`
2. **استخراج UUID الخدمة** ومعلومات أخرى من سجل الخدمة
3. **إضافة رصيد إلى نظام الفوترة** (OCS) باستخدام UUID الخدمة
4. **تسجيل المعاملة** في CRM لأغراض الفوترة

يسمح هذا التدفق للإضافة بتحديد بالضبط أي خدمة يجب تعديلها وتطبيق التغييرات بشكل مناسب.

الفرق بين المتغيرات المستقلة والإضافية

المنتجات المستقلة تتلقى:

- `product_id` - المنتج الذي يتم توفيره
- `customer_id` - العميل الذي يطلب الخدمة
- معرفات عناصر المخزون (مثل، `SIM Card`، `Mobile Number`) إذا كان المنتج يتطلبها
- `access_token` - لمصادقة API

تتلقى **المنتجات الإضافية**:

- `product_id` - المنتج الإضافي الذي يتم توفيره
- `customer_id` - العميل الذي يمتلك الخدمة
- `service_id` - **معرف الخدمة الموجودة التي يجب تعديلها** (هذا هو الفرق الرئيسي)
- `access_token` - لمصادقة API

`service_id` - هذا يخبر الكتاب أي خدمة موجودة يجب تعديلها أو إضافة الف��ق الرئيسي هو إليها.

المنتجات المجمعة

تُوفر المنتجات المجمعة مثل الإضافات ولكن قد ينشئ كتابها سجلات خدمة متعددة. تتلقى نفس المتغيرات مثل الإضافات، بما في ذلك:

- `product_id` - المنتج المجمّع
- `customer_id` - العميل
- `service_id` - الخدمة الرئيسية (إذا كان ذلك مناسبًا)
- معرفات عناصر المخزون (مثل، `مودم راوتر`) إذا كانت مطلوبة
- `access_token` - لمصادقة API

خدمات متعددة ذات صلة (`play_seniors_package`، مثل) ثم ينشئ كتاب الحزمة (إنترنت، IPTV، هاتف) ويربطها معًا.

الخدمات

الخدمة هي نسخة من منتج تنتمي إلى عميل، يتم تحصيل الرسوم عليها.

الذي يتعامل مع توليد الرسوم (نظام الشحن عبر الإنترنت) `OCS` إنها في الأساس رابط إلى حساب ويدير الأرصدة CGRateS بواسطة OCS والأرصدة والاستخدامات الفعلية للحساب. يتم تشغيل النقدية، والأرصدة الوحد��ية (البيانات، الصوت، الرسائل القصيرة)، وخطط العمل للتجديد التلقائي، و ThresholdSلحدود الإنفاق.

إضافة خدمة: اختيار المنتج والتصفية

عند إضافة خدمة إلى عميل (إما خدمة مستقلة جديدة أو إضافة إلى خدمة موجودة)، يعرض النظام المنتجات المتاحة في واجهة دوارة. يتم تصفية المنتجات المعروضة بناءً على عدة معايير:

تصفية المنتجات للخدمات المستقلة

عند إنشاء خدمة جديدة لعميل، تعرض واجهة المستخدم المنتجات المصفاة حسب:

1. **نوع العميل** - يتم تصنيف المنتجات على أنها:
   - **فردية (سكنية)**: منتجات حيث `residential = true` أو `business = false`
   - **تجارية**: منتجات حيث `business = true`
2. **الفئة** - يتم فصل المنتجات إلى:
   - **خطط الخدمة**: منتجات مع `category` = `مستقل` أو `حزمة`
   - **الإضافات**: منتجات مع `category` = `إضافة` (تظهر في دوارة منفصلة)
3. **التوفر** - يتم عرض المنتجات فقط إذا:
   - `enabled = true` - المنتج نشط وغير معطل
   - `available_until` و `available_from` - التاريخ الحالي بين المنتج ضمن نافذة توفره
   - `customer_can_purchase = true` - (إذا كان العميل يشتري بنفسه) المنتج يسمح بالشراء المباشر من قبل العميل

> **ⓘ ملاحظة**
>
> **تصفية على مستوى API**: يقوم API تلقائيًا بتصفية المنتجات حسب حالة التفعيل وتواريخ التوفر على مستويين:
>
> - **نقاط نهاية الشراء/الاختيار** (`/crm/product/`) - تُستخدم بواسطة إضافة النموذج وقائمة الخطط لاختيار المنتجات. يتم تصفيتها تلقائيًا لعرض المنتجات المفعلة فقط ضمن نطاق تواريخ توفرها. يضمن ذلك أن العملاء والموظفين يمكنهم فقط اختيار المنتجات المتاحة حاليًا للشراء.

- **نقاط نهاية الإدارة** (`/crm/product/paginated`) - تُستخدم بواسطة صفحة إدارة المنتج. تعرض جميع المنتجات بما في ذلك المعطلة وخارج تواريخ التوفر، مما يسمح للمسؤولين بإدارة الكتالوج الكامل للمنتجات بما في ذلك المنتجات غير النشطة.

مرر `include_disabled=true` إلى نقطة نهاية المنتج الأساسية لتجاوز التصفية (للاستخدام الإداري فقط).

تعرض واجهة المستخدم دوارات منفصلة لـ:

- **خطط الخدمة الفردية** - منتجات سكنية للعملاء المستهلكين
- **خطط الخدمة التجارية** - منتجات تجارية للعملاء التجاريين
- **الإضافات الفردية** - حزم إضافية سكنية
- **الإضافات التجارية** - حزم إضافية تجارية

تصفية المنتجات لخدمات الإضافات

عند إضافة إضافة إلى **خدمة موجودة**، يتم تطبيق تصفية إضافية:

1. **مطابقة نوع الخدمة** - يتم عرض الإضافات فقط التي تتطابق مع `service_type`:
   - إذا كانت الخدمة الموجودة تحتوي على `service_type = "محمول"`، يتم عرض الإضافات فقط التي تحتوي على `service_type = "محمول"`
   - يضمن ذلك أن عملاء المحمول يرون فقط إضافات المحمول، وعملاء الإنترنت يرون فقط إضافات الإنترنت، إلخ.
2. **التحقق من الاعتمادات** - إذا كانت الإضافة تحتوي على `relies_on_list`:
   - يتحقق النظام مما إذا كان لدى العميل المنتجات/الخدمات المطلوبة
   - يتم عرض الإضافات التي تم الوفاء بمتطلباتها فقط
3. **تصفية نفس نوع العميل** - لا تزال الإضافات مصفاة حسب `residential` مقابل `business` لتتناسب مع نوع العميل

سيناريو تصفية مثال

لعميل تجاري لديه خدمة محمولة موجودة (`service_type = "محمول"`):

- **المنتجات المستقلة المعروضة**: جميع المنتجات التجارية المستقلة/المجمعة (`business = true`، `category != "إضافة"`)

- **المنتجات الإضافية المعروضة**: فقط إضافات المحمول التجارية (`business = true`، `category = "إضافة"`، `service_type = "محمول"`)
- **المنتجات المخفية**: المنتجات السكنية، الإضافات لأنواع خدمات أخرى (الإنترنت، الصوت، إلخ)، المنتجات المعطلة

حقول الخدمة

يحتوي نموذج الخدمة على حقول تتبع نسخة الخدمة المقدمة وعلاقتها بالعميل، المنتج، ونظام الفوترة.

المعلومات الأساسية للخدمة

- **service_id** - معرف فريد يتم تعيينه تلقائيًا بواسطة النظام (للقراءة فقط)
- **customer_id** - رابط إلى العميل الذي يمتلك هذه الخدمة (للقراءة فقط بعد الإنشاء)
- **product_id** - رابط إلى المنتج الذي تم إنشاء هذه الخدمة منه (للقراءة فقط بعد الإنشاء)
- **service_name** - الاسم المعروض للعملاء (قابل للتحرير)
- **service_type** - نوع الخدمة: محمول، إنترنت، VoIP، IPTV، حزمة، إلخ. (قابل للتحرير)
- **service_uuid** - معرف فريد يُستخدم في OCS/CGRateS للفوترة (للقراءة فقط، يتم إنشاؤه تلقائيًا)
- **icon** - فئة أيقونة FontAwesome (قابل للتحرير) للعرض في بوابة الرعاية الذاتية

حالة الخدمة والتواريخ

- **service_status** - الحالة الحالية: نشطة، غير نشطة، معلقة، إلخ. (قابل للتحرير)
- **service_provisioned_date** - عندما تم توفير الخدمة لأول مرة (تم تعيينه تلقائيًا، للقراءة فقط)
- **service_active_date** - عندما أصبحت الخدمة نشطة (قابل للتحرير)
- **service_deactivate_date** - عندما تنتهي الخدمة أو سيتم إلغاء تنشيطها (قابل للتحرير)
- **contract_end_date** - تاريخ انتهاء الالتزام بالعقد (قابل للتحرير)

الفوترة والتسعير

- **retail_cost** - الرسوم الشهرية المتكررة على العميل (قابل للتحرير)

- **wholesale_cost** - التكلفة الخاصة بك لتوفير الخدمة (قابل للتحرير)
- **service_billed** - ما إذا كانت هذه الخدمة تظهر في الفواتير (قابل للتحرير، الافتراضي: true)
- **service_taxable** - ما إذا كانت الضرائب تنطبق على هذه الخدمة (قابل للتحرير، الافتراضي: true)
- **invoiced** - ما إذا كانت الخدمة قد تم فوترتها (تم تعيينها تلقائيًا بواسطة نظام الفوترة)
- **promo_code** - الرمز الترويجي المستخدم عند إنشاء الخدمة (قابل للتحرير)

رؤية العميل

- **service_visible_to_customer** - ما إذا كان يمكن للعميل رؤية هذه الخدمة في بوابة الرعاية الذاتية (قابل للتحرير، الافتراضي: true)
- **service_usage_visible_to_customer** - ما إذا كان يمكن للعميل عرض تفاصيل الاستخدام/الرصيد (قابل للتحرير، الافتراضي: true)

تكوين التوفير

- **provisioning_play** - كتاب أنسبل المستخدم لتوفير هذه الخدمة (موروث من المنتج، للقراءة فقط)
- **provisioning_json_vars** - المتغيرات المرسلة إلى كتاب التوفير (موروثة من المنتج، للقراءة فقط)
- **deprovisioning_play** - كتاب أنسبل الذي يتم تشغيله عند إلغاء توفير الخدمة (للقراءة فقط)
- **deprovisioning_json_vars** - المتغيرات لكتاب إلغاء التوفير (للقراءة فقط)

علاقات الخدمة

- **bundled_parent** - إذا كانت هذه الخدمة جزءًا من حزمة، فإن service_id للخدمة الرئيسية (للقراءة فقط)
- **site_id** - رابط إلى الموقع/المكان الفعلي حيث يتم تقديم الخدمة (قابل للتحرير)

ملاحظات وبيانات التعريف

- **service_notes** - ملاحظات داخلية حول الخدمة للرجوع إليها من قبل الموظفين (قابل للتحرير)
- **created** - الطابع الزمني عندما تم إنشاء الخدمة (تم تعيينه تلقائيًا، للقراءة فقط)
- **last_modified** - الطابع الزمني لأحدث تحديث (يتم تحديثه تلقائيًا، للقراءة فقط)

الحقول القابلة للتحرير مقابل الحقول للقراءة فقط

قابلة للتحرير عبر API/UI:

يمكن تحديث الخدمات عبر `PATCH /crm/service/{service_id}` مع هذه الحقول:

- service_name، service_type، service_status
- service_notes
- retail_cost، wholesale_cost
- service_billed، service_taxable
- service_visible_to_customer، service_usage_visible_to_customer
- service_active_date، service_deactivate_date، contract_end_date
- icon، promo_code، site_id

للقراءة فقط (تم تعيينها تلقائيًا):

لا يمكن تعديل هذه الحقول مباشرة بعد الإنشاء:

- service_id، customer_id، product_id
- service_uuid (تم إنشاؤه أثناء التوفير)
- service_provisioned_date
- provisioning_play، provisioning_json_vars
- deprovisioning_play، deprovisioning_json_vars
- bundled_parent
- invoiced (يدار بواسطة نظام الفوترة)
- created، last_modified (يدار تلقائيًا)

توفير المنتجات إلى خدمات

تتحول عملية التوفير منتجًا (نموذج) إلى خدمة (نسخة محددة للعميل) من خلال سلسلة من الخطوات المنسقة التي تشمل واجهة الويب، API، وكتب أنسبل.

تدفق التوفير على مستوى عالٍ

1. **إعداد ما قبل التوفير** - يتم إنشاء المنتج في API مع تكوين التوفير، وكتابة واختبار كتب أنسبل المقابلة

2. "**اختيار الخدمة** - من صفحة العميل، يختار الموظف أو العميل "إضافة خدمة
3. **تصفية المنتجات** - يتم تصفية المنتجات المعروضة بناءً على:
    - نوع العميل (سكني/تجاري)
    - الخدمات الموجودة (`relies_on_list` للاعتبارات الاعتمادية في)
    - تواريخ التوفر (`available_from`/`available_until`)
    - علامات `enabled` و `customer_can_purchase`
4. **التخصيص** - خيار لتجاوز متغيرات التوفير (لتعديلات الأسعار، التكوينات المخصصة، إلخ.)
5. **اختيار المخزون** - إذا كان المنتج يتطلب المخزون (`inventory_items_list` ليس فارغًا)، يختار المستخدم عناصر محددة (أي رقم هاتف، SIM مثل، أي بطاقة)
6. **بدء التوفير** - عند النقر على زر "توفير"، يقوم بإنشاء وظيفة توفير API

تدفق التكامل التفصيلي بين أنسبل API

عند توفير خدمة، تحدث التسلسل التالي:

الخطوة 1: إنشاء وظيفة التوفير

يتلقى API طلب التوفير وينشئ وظيفة توفير مع:

- `provisioning_play` - اسم كتاب أنسبل (مثل، `play_psim_only`)
- `provisioning_json_vars` - سلسلة JSON من المتغيرات من المنتج أو التي تم تجاوزها بواسطة الطلب
- `customer_id` - معرف العميل الذي يطلب الخدمة
- `product_id` - معرف المنتج الذي يتم توفيره
- `service_id` - معرف الخدمة الموجودة للإضافات (اختياري)
- اختيارات المخزون - معرفات عناصر المخزون المحددة

الخطوة 2: تجميع المتغيرات

يعمل خدمة التوفير على دمج المتغيرات من مصادر متعددة بهذا الترتيب:

1. متغيرات `provisioning_json_vars` (الافتراضات من تعريف المنتج)
2. معلمات جسم الطلب (يمكن أن تتجاوز الافتراضات الخاصة بالمنتج)
3. المتغيرات التي أضافها النظام:
    - `access_token` - رمز JWT لمصادقة API مرة أخرى إلى CRM

    - `initiating_user` - معرف المستخدم الذي قام بتفعيل التوفير
    - `customer_id`، `product_id`، `service_id`
4. اختيارات المخزون - تتم إضافتها كأزواج `{inventory_type: inventory_id}`

مثال على المتغيرات المدمجة:

```
{
    "customer_id": 123,
    "product_id": 456,
    "service_id": 789,          # فقط للإضافات
    "SIM Card": 1001,           # من اختيار المخزون
    "Mobile Number": 1002,      # من اختيار المخزون
    "monthly_cost": 30,         # من provisioning_json_vars
    "data_gb": 50,              # من provisioning_json_vars
    "access_token": "eyJ...",   # تمت إضافتها للنظام لاستدعاءات API
    "initiating_user": 5        # تمت إضافتها للنظام
}
```

الخطوة 3: إنشاء سجل التوفير

يتم إنشاء سجل `Provision` مع في قاعدة البيانات:

- `provision_id` - معرف فريد للتتبع
- `provisioning_play` - اسم ملف الكتاب
- `provisioning_json_vars` - المتغيرات المدمجة كسلسلة JSON
- `task_count` - عدد المهام في الكتاب (مستخرج من YAML)
- `provisioning_status` - رمز الحالة (يتم تعيينه أولاً إلى 1 = قيد التشغيل، ثم يتم تحديثه إلى 0 = نجاح، 2 = فشل، أو قد يبقى 1 إذا كان لا يزال في التقدم)
- `product_id`، `customer_id`، `service_id` - مراجع السياق

الخطوة 4: تنفيذ الكتاب في الخلفية

يولد API خيطًا في الخلفية يقوم بـ:

1. تحميل YAML الكتاب من `OmniCRM-API/Provisioners/plays/{playbook_name}.yaml`
2. استدعاء `ansible_runner.run()` مع:
    - `playbook` - مسار ملف YAML المحمل

- `extravars` - جميع المتغيرات المدمجة (تم تمريرها إلى أنسبل)
- `inventory` - تعيين إلى `'localhost,'` (تنفيذ محلي)
- `event_handler` - معالج مخصص لالتقاط أحداث تنفيذ المهام

3. مراقبة تنفيذ الكتاب في الوقت الحقيقي

الخطوة 5: التقاط الأحداث والتسجيل (`ProvisioningEventHandler`)

مع تنفيذ كل مهمة أنسبل، يتم التقاط الأحداث وتخزينها كسجلات `Provision_Event`:

- `event_name` - اسم المهمة من الكتاب
- `event_number` - رقم التسلسل
- `provisioning_status` - رمز الحالة الذي يشير إلى نتيجة المهمة:
  - **0** = نجاح - تم إكمال المهمة بنجاح
  - **1** = قيد التشغيل - المهمة قيد التنفيذ حاليًا
  - **2** = فشل - فشل حرج يوقف التوفير
  - **3** = فشل (تم تجاهله) - فشلت المهمة ولكن تم تجاهل الأخطاء (في الكتاب `ignore_errors: true`)
- `provisioning_result_json` - نتائج المهمة مع البيانات الحساسة محجوبة

يقوم معالج الأحداث تلقائيًا بإزالة كلمات المرور، المفاتيح، الأسرار، وغيرها من البيانات الحساسة من السجلات.

الخطوة 6: تنفيذ كتاب أنسبل (`Provisioners/plays/*.yaml`)

يعمل كتاب أنسبل محليًا وعادةً ما يقوم بتنفيذ هذه الإجراءات:

1. استدعاء - **استرداد تعريف المنتج** GET إلى `/crm/product/product_id/{{ product_id }}` باستخدام `{{ access_token }}`

2. استدعاء - **استرداد معلومات العميل** GET إلى `/crm/customer/customer_id/{{ customer_id }}`

3. استدعاء - (إذا لزم الأمر) **معالجة عناصر المخزون** GET إلى `/crm/inventory/inventory_id/{{ inventory_id }}` لكل عنصر محدد لاسترداد التفاصيل الكاملة (ICCID، MSISDN، إلخ، أرقام السلاسل.)

4. إجراء استدعاءات - **تكوين الأنظمة الخارجية** API إلى:

   - HSS (خادم المشتركين المنزليين) لتوفير المشتركين

    - IMS (نظام الوسائط المتعددة IP) لتسجيل الصوت
    - CGRateS/OCS لإنشاء الحساب، تكوين الشحن، خطط الأسعار
    - خوادم ENUM لتخطيط أرقام الهواتف
    - معدات الشبكة (المودمات، المفاتيح، إلخ.)

5. **إضافة تكاليف الإعداد** (إذا لزم الأمر) - POST إلى `/crm/transaction/` لتسجيل الرسوم لمرة واحدة

6. **تحصيل الرسوم من العميل** - POST إلى OCS/CGRateS لتحصيل retail_setup_cost إذا تم تكوينه

7. **إنشاء حساب OCS** - POST إلى OCS/CGRateS لإنشاء حساب فوترة مع UUID

8. **تكوين الرسوم المتكررة** OCS/CGRateS - إنشاء إجراءات وخطط إجراءات في للرسوم الشهرية المتكررة

9. **إنشاء سجل الخدمة** - PUT/POST إلى `/crm/service/` لإنشاء سجل الخدمة في CRM:

```
{
  "customer_id": 123,
  "product_id": 456,
  "service_name": "بطاقة SIM 0412345678 - محمولة",
  "service_uuid": "generated-uuid-for-ocs",
  "service_status": "نشط",
  "service_type": "محمول",
  "retail_cost": 30,
  "wholesale_cost": 5,
  "provisioning_play": "play_psim_only",
  "provisioning_json_vars": "{...}"
}
```

10. **تعيين المخزون** - PATCH إلى `/crm/inventory/inventory_id/{{ inventory_id }}` لتحديد المخزون كـ "مخصص" للخدمة

11. **إرسال الإشعارات** (اختياري) - بريد إلكتروني أو رسالة نصية للعميل مع تفاصيل الخدمة

الخطوة 7: الانتهاء وتحديث الحالة

عند انتهاء الكتاب:

- **نجاح**: يتم تحديث `Provision.provisioning_status` إلى **0** (نجاح)
- **فشل حرج**: يتم تحديث `Provision.provisioning_status` إلى **2** (فشل)، ويتم إرسال بريد إلكتروني بالفشل إلى `crm_config.provisioning.failure_list`
- **الفشل غير الحرج**: المهام التي تفشل مع `ignore_errors: true` يتم وضع علامة عليها بالحالة **3** (فشل ولكن تم تجاهله) ولا توقف التوفير

الخدمة المقدمة م��ئية الآن في CRM (إذا نجح التوفير) ونشطة للعميل.

الاختلافات الرئيسية: التوفير المستقل مقابل الإضافات مقابل الحزم

المنتجات المستقلة (`category: مستقل`):

- تتلقى `customer_id` و `product_id`
- تتطلب عادةً عناصر المخزون (أرقام الهواتف، المودمات، SIM بطاقات)
- تنشئ سجل خدمة **جديد** عبر API PUT `/crm/service/`
- توفر موارد جديدة على الأنظمة الخارجية (معدات الشبكة، OCS، HSS)
- مثال: تفعيل بطاقة SIM جديدة، اتصال إنترنت جديد

المنتجات الإضافية (`category: إضافة`):

- تتلقى `customer_id`، `product_id` و **`service_id`** (الخدمة الموجودة لتعديلها)
- عادةً لا تتطلب المخزون (أو مخزونًا ضئيلًا)
- **تعدل الخدمة الموجودة** أو تضيف رسومًا إلى حساب OCS الموجود
- قد تنفذ إجراءات على OCS (إضافة حزمة بيانات، إضافة رصيد، تمكين ميزة)
- لا تنشئ سجلات خدمة جديدة (أو تنشئ سجلات خدمة فرعية مرتبطة بالأصل)
- مثال: إعادة شحن خطة بيانات شهرية، حزمة تجوال دولية، رصيد إضافي

المنتجات المجمعة (`category: حزمة`):

- مشابهة للإضافات من حيث المتغيرات المستلمة
- قد تتطلب بعض عناصر المخزون (مثل، مودم للحزمة المنزلية)
- تنشئ **سجلات خدمة متعددة** مرتبطة (إنترنت + تلفزيون + هاتف)
- توفر موارد متعددة عبر أنظمة مختلفة

- تربط الخدمات معًا في CRM للفوترة/الإدارة الموحدة
- مثال: حزمة منزلية (إنترنت + IPTV + هاتف VoIP)

متطلبات كتاب التوفير

لكي يعمل الكتاب بشكل صحيح، يجب أن:

1. **يكون موجودًا في** `OmniCRM-API/Provisioners/plays/{playbook_name}.yaml`
2. **يقبل المتغيرات** عبر `extravars` لأنسبل (يتم الوصول إليها كـ `{{ variable_name }}`)
3. **يصادق على استدعاءات API** باستخدام `Authorization: Bearer {{ access_token }}` رأس
4. **يتعامل مع الفشل بشكل جيد** `rescue` و `ignore_errors` حيثما كان ذلك مناسبًا
5. **ينشئ سجل الخدمة** للمنتجات المستقلة، أو يعدل الخدمة الموجودة للإضافات
6. **يخصص المخزون** إذا تم اختيار عناصر المخزون
7. **يعود برسائل خطأ ذات مغزى** عبر وحدة `fail` عند حدوث أخطاء حرجة

المتغيرات الشائعة المتاحة في الكتب

تتلقى كل كتاب هذه المتغيرات:

- `customer_id` - عدد صحيح، العميل الذي يطلب الخدمة
- `product_id` - عدد صحيح، المنتج الذي يتم توفيره
- `service_id` - عدد صحيح (فقط للإضافات/الحزم)، الخدمة الموجودة لتعديلها
- `access_token` - سلسلة، رمز JWT لمصادقة API إلى CRM
- `initiating_user` - عدد صحيح، المستخدم الذي قام بتفعيل التوفير
- بالإضافة إلى أي معرفات عناصر المخزون: `{{ inventory_type }}: inventory_id`
- بالإضافة إلى أي متغيرات من `provisioning_json_vars`
- بالإضافة إلى أي متغيرات تم تمريرها في طلب التوفير

يمكن أن تستخدم الكتب هذه لت:

- استرداد تفاصيل المنتج الكاملة: `GET /crm/product/product_id/{{ product_id }}`
- استرداد تفاصيل العميل: `GET /crm/customer/customer_id/{{ customer_id }}`
- استرداد تفاصيل المخزون: `GET /crm/inventory/inventory_id/{{ SIM_Card }}`
- إنشاء المعام��ات: `POST /crm/transaction/`
- إنشاء الخدمات: `PUT /crm/service/`
- تحديث الخدمات: `PATCH /crm/service/{{ service_id }}`
- تخصيص المخزون: `PATCH /crm/inventory/inventory_id/{{ inventory_id }}`

مثال: تدفق كتاب إضافة بسيطة

لإضافة إعادة شحن بيانات محمولة:

1. يتلقى الكتاب: `customer_id`، `product_id`، `service_id`، `access_token`
2. استرداد تفاصيل الخدمة: `GET /crm/service/{{ service_id }}` للحصول على `service_uuid`
3. استرداد تفاصيل المنتج: `GET /crm/product/product_id/{{ product_id }}` للحصول على التسعير و كمية البيانات
4. تحصيل الرسوم من العميل في OCS: `POST إلى CGRateS` لخصم retail_cost من الرصيد
5. إضافة رصيد بيانات في OCS: `POST إلى CGRateS` لإضافة رصيد البيانات مع انتهاء الصلاحية
6. تسجيل المعاملة في CRM: `POST /crm/transaction/` مع تفاصيل الشحن
7. الانتهاء بنجاح

لأغراض التصحيح `Provision_Event` و `Provision` يتم تتبع العملية بأكملها في جداول والتدقيق.

مشاركة OCS

يتعامل `OCS` (نظام الشحن عبر الإنترنت)، الذي يتم تنفيذه عبر CGRateS، مع جميع التحصيلات، OCS، يعمل كمرجع لحساب CRM في الوقت الحقيقي وتتبع الاستخدام للخدمات. يسجل خدمة

الذي يدير:

- رسوم الاشتراك، DID **الرسوم المتكررة** - الرسوم الشهرية، إيجار
- **التحصيلات المعتمدة على الاستخدام** - مكالمات الصوت بالدقيقة، البيانات لكل ميغابايت، رسوم الرسائل القصيرة لكل رسالة
- بيانات) **إدارة الأرصدة** - الأرصدة النقدية (رصيد مدفوع مسبقًا) والأرصدة الوحدوية (دقائق الصوت، عدد الرسائل القصيرة، GB)
- مثل، إنفاق $30) **تحويل الأرصدة** - تحويل الأرصدة النقدية إلى أرصدة وحدوية للحصول على حزمة بيانات 10 GB)
- Thresholds **حالة الحساب** - نشط، معلق، معطل بناءً على حدود الائتمان و

على بيانات التعريف والتكوين (العميل، المنتج، التسعير، الرؤية)، بينما CRM تحتوي سجل خدمة على الحالة الفوترة الحية (الأرصدة، الاستخدام، الرسوم). OCS يحتوي

استرداد استخدام الخدمة والأرصدة

وعرضها على العملاء والموظفين في OCS/CGRateS يتم استرداد معلومات استخدام الخدمة من الوقت الحقيقي.

كيفية استرداد الاستخدام

عند طلب استخدام خدمة (أو عبر واجهة المستخدم API)، تحدث التدفق التالي:

1. **طلب API** - تستدعي الواجهة الأمامية `GET /crm/service/{service_id}` أو تعرض تفاصيل الخدمة في واجهة المستخدم

2. **بحث الخدمة** - يسترد API سجل الخدمة من قاعدة البيانات، يستخرج `service_uuid`

3. **استدعاءات API CGRateS** - يقوم النظام بإجراء مكالمتين إلى CGRateS:

    i. **Get_Balance(service_uuid)** - يسترد رصيد الحساب مع `BalanceMap`
        - يعيد الأرصدة منظمة حسب النوع: البيانات، الصوت، الرسائل القصيرة، النقدية، بيانات الدونجل
        - يتضمن كل رصيد: ID، القيمة، تاريخ الانتهاء، الوزن، DestinationIDs

- يضيف النظام حقول قابلة للقراءة البشرية: `custom_Name_hr`, `custom_Expiration`, `custom_Description_String`

ii. **Get_ActionPlans(service_uuid)** - يسترد خطط الإجراءات النشطة للتجديد التلقائي (المغطاة في القسم التالي)

4. يتم دمج بيانات CGRateS في استجابة الخدمة - **دمج الاستجابة**:

```
{
  "service_id": 123,
  "service_name": "خدمة المحمول",
  "service_uuid": "abc-123-def",
  "cgrates": {
    "BalanceMap": {
      "DATA": [{
        "ID": "DATA_10GB",
        "Value": 5368709120,
        "ExpirationDate": "2025-02-01T00:00:00Z",
        "custom_Name_hr": "10 حزمة بيانات GB",
        "custom_Expiration": "1 فبراير، 2025",
        "custom_Description_String": "5 جيجابايت المتبقية"
      }],
      "VOICE": [{
        "ID": "VOICE_UNLIMITED",
        "Value": 999999999,
        "custom_Name_hr": "مكالمات غير محدودة",
        "custom_Description_String": "دقائق غير محدودة"
      }],
      "MONETARY": [{
        "ID": "PREPAID_CREDIT",
        "Value": 25.50,
        "custom_Description_String": "رصيد $25.50"
      }]
    },
    "ActionPlans": [...]
  }
}
```

5. **عرض واجهة المستخدم** - تعرض مكونات الواجهة الأمامية بيانات الاستخدام:

- **ServiceUsage.js** - المكون الرئيسي لعرض الاستخدام مع تحديث تلقائي كل 3 ثوانٍ

- **UsageCard.js** - بطاقات ملخص لكل نوع رصيد
- **UsageProgress.js** - أشرطة التقدم التي تظهر النسبة المئوية المستخدمة/المتبقية
- يتم ترميز الأرصدة بالألوان وتنسيقها لسهولة القراءة

هيكل بيانات الاستخدام

يحتوي كل رصيد في `BalanceMap` على:

حقول CGRateS الأصلية:

- `ID` - معرف فريد للرصيد (مثل، "DATA_10GB_2025_01")
- `Value` - مقدار الرصيد:
  - للبيانات: بايت (5368709120 = 5 جيجابايت)
  - للصوت: ثوانٍ (3600 = 1 ساعة)
  - للرسائل القصيرة: عدد (100 = 100 رسالة)
  - للنقدية: وحدات العملة (25.50 = 25.50$)
- `ExpirationDate` - الطابع الزمني ISO 8601 عندما تنتهي صلاحية الرصيد
- `Weight` - الأولوية لاستهلاك الرصيد (يتم استهلاك الوزن الأعلى أولاً)
- `DestinationIDs` - الوجهات التي ينطبق عليها هذا الرصيد (مثل ["AU", "INTERNATIONAL"])

حقول قابلة للقراءة البشرية (تمت إضافتها بواسطة CRM):

- `custom_Name_hr` - اسم قابل للقراءة البشرية مستخرج من ID
- `custom_Expiration` - تاريخ انتهاء الصلاحية بتنسيق (مثل، "15 يناير، 2025" أو "في 11 يومًا")
- `custom_Description_String` - وصف الرصيد القابل للقراءة البشرية:
  - البيانات: "5 جيجابايت المتبقية" أو "10 جيجابايت إجمالي"
  - الصوت: "60 دقيقة المتبقية" أو "غير محدودة"
  - الرسائل القصيرة: "50 رسالة نصية المتبقية"
  - النقدية: "25.50$ رصيد"

التحكم في رؤية الاستخدام

يتم التحكم في رؤية استخدام الخدمة بواسطة حقلين:

- **service_visible_to_customer** - إذا كان false، يتم إخفاء الخدمة تمامًا من بوابة الرعاية الذاتية للعميل
- **service_usage_visible_to_customer** - إذا كان false، تكون الخدمة مرئية ولكن تفاصيل الاستخدام/الرصيد مخفية (يمكن للعميل رؤية أنه لديه الخدمة، ولكن ليس مقدار ما استخدمه)

يسمح ذلك للمشغلين بـ:

- إخفاء الخدمات الداخلية/التجريبية من العملاء
- عرض وجود الخدمة دون الكشف عن الاستخدام (مفيد للخطط غير المحدودة أو الخدمات الحساسة)
- عرض الاستخدام بشكل شفاف بالكامل (الافتراضي)

تحديثات الاستخدام في الوقت الحقيقي

تقوم واجهة الويب بتحديث بيانات الاستخدام تلقائيًا:

- **الفترة**: كل 3 ثوانٍ (يمكن تكوينها في مكون ServiceUsage)
- **الطريقة**: تستدعي `GET /crm/service/{service_id}` التي fetch البيانات الحية من CGRateS
- **الكفاءة**: يتم تحديث فقط عرض الخدمات النشطة؛ تستخدم قوائم العرض البيانات المخزنة

يضمن ذلك أن العملاء والموظفين يرون تحديثات الأرصدة في الوقت القريب عند حدوث الاستخدام.

الرسوم المتكررة / التجديد التلقائي

`OCS` أولاً كإجراءات داخل DID تُنشأ الرسوم المتكررة، مثل رسوم الخدمة الشهرية، أو رسوم لكل وتأخذ الشكل `Action_ServiceUUID_ServiceName_WhatitDoes`.

من الاستخدام، ستبدو الإجراء كما يلي: لنفترض أن لدينا خدمة GPON 1TB بقيمة 60$ شهريًا تشمل

`Action_kj49-adsf-1234-9742_60_GPON_1TB_MonthlyExpiry`

1. إعادة تعيين الرصيد النقدي إلى 0$
2. إرسال POST HTTP إلى `/simple_provision` على CRM لتوفير شيء ما

3. 1 إضافة رصيد لاستخدام TB ينتهي في شهر واحد

إذا أردنا جعل MRC متكررة (نعم)، فسننشئ `ActionPlan` باسم `ActionPlan_{{ service_uuid }}_Monthly_Charge` والذي سيكون له الوقت المحدد على *شهري* ليتم تنشيطه كل شهر، وتعيين `ActionPlan` للحساب.

يمكننا تعيين تاريخ انتهاء الصلاحية بناءً على معلمة *السنة / الأشهر / الأيام* لتحديد متى ستتوقف MRC أيضًا، على سبيل المثال لخدمة ثابتة لمدة 12 شهرًا تتوقف بعد هذه النقطة.

`ActionPlans` فريدة لكل خدمة، فإنها لا تشارك أي شيء مع أي خدمات نظرًا لأن الإجراءات و أخرى.

هذا يعني أنه يمكننا توفيرها بقيم معدلة، ولن يؤثر ذلك على أي خدمات أخرى.

الإضافات والإضافات

تتم معالجة الإضافات / الإضافات مثل شراء بيانات إضافية، حزم التجوال، دقائق دولية، إلخ، بنفس الطريقة. يتم إنشاء إجراء للقيام بما هو مطلوب، مثل تحصيل قيمة نقدية ثم منح رصيد وحدوي مع انتهاء محدد.

لإضافته تلقائيًا إلى الحساب، نقوم فقط بتفعيل `ActionPlans` بدلاً من استخدام `ExecuteAction` للإجراء الذي أنشأناه للتو مرة واحدة من داخل أنسبل.

إضافة أرصدة نقدية مدفوعة مسبقًا

بقيمة 10$، يتم إضافتها كرصيد نقدي، PAYG بالنسبة للأرصدة النقدية المدفوعة مسبقًا، مثل خطة ولكن مع أولوية أعلى.

حد الائتمان على هذه الخدمات للرصيد الافتراضي سيكون 0$.

حدود الائتمان / منع الإنفاق المفرط

`ThresholdS` تُستخدم على كل حساب لتحديد الحد الأقصى للإنفاق لفترة زمنية معينة.

بالنسبة للعملاء المدفوعين مسبقًا / PAYG، 0$ يكون هذا.

التفاعل مع OCS عبر CRM

لكل خدمة يمكنك رؤية `Balances` و `ActionPlans` و `Actions` و `ThresholdS` من `OCS` من داخل API CRM.

ويتم تنفيذها عبر كتب أنسبل. يمكن، حسب الحاجة من API CRM `ActionPlans` يمكن إزالة عن طريق إضافة إضافة/خدمة وتنفيذها عبر كتب، حسب الحاجة من CRM، إضافة `ActionPlans` أنسبل.

ومن استهلاك الخدمات `ActionPlans` مما سيتوقف عن تنفيذ، `OCS` يمكن تعطيل حسابات.

وفقًا للسياسة للمنتج `ThresholdS` بالنسبة لحدود الائتمان، يتم تعيين قيمة.

عرض وإدارة ActionPlans في CRM

يتم عرض وإدارة ActionPlans (تكوينات التجديد التلقائي) من خلال واجهة CRM، مما يسمح للموظفين والعملاء برؤية التجديدات التلقائية القادمة وإدارتها.

كيفية استرداد ActionPlans وعرضها

عند عرض خدمة في CRM، يتم استرداد ActionPlans وعرضها تلقائيًا:

1. **استدعاء API** - عند استدعاء `GET /crm/service/{service_id}`، يقوم API:
    - باسترداد سجل الخدمة من قاعدة البيانات
    - استخراج `service_uuid` (معرف حساب OCS)
    - استدعاء API CGRateS لاسترداد ActionPlans بواسطة UUID الخدمة
    - يستدعي هذا داخليًا `Get_ActionPlans(service_uuid)` على CGRateS
2. **هيكل بيانات ActionPlan** - يحتوي كل ActionPlan تم إرجاعه على:

```
{
    "ActionPlanId": "ServiceID_abc-123-
def__ProductID_456__...",
    "PlanName": "خطة التجديد الشهرية",
    "NextExecTime": "2025-02-01T00:00:00Z",
    "custom_NextExecTime_hr": "في 11 يومًا",
    "ActionPlanId_split_dict": {
        "ServiceID": "abc-123-def",
        "ProductID": 456,
        "CustomerID": 789,
        ...
    }
}
```

- `ActionPlanId` - معرف فريد يحتوي على معلومات مشفرة عن الخدمة/المنتج/العميل
- `PlanName` - اسم خطة العمل (عادةً اسم كتاب التجديد)
- `NextExecTime` - الطابع الزمني ISO عندما ستنفذ خطة العمل التالية
- `custom_NextExecTime_hr` - الوقت القابل للقراءة البشرية حتى التنفيذ (مثل، "في 11 يومًا"، "غدًا"، "1 فبراير، 2025")
- `ActionPlanId_split_dict` - قاموس بمكونات مفصولة من ActionPlanId

3. يتم عرض ActionPlans في مكون **عرض واجهة المستخدم** **ActionPlansTable**:

   أعمدة الجدول:

   - `ProductID` من **اسم المنتج** - يتم استرداده من خلال البحث عن ActionPlanId
   - `retail_cost` من تعريف المنتج **التكلفة** - تظهر
   - `custom_NextExecTime_hr` (قابل للقراءة البشرية) **تاريخ التجديد** - يعرض
   - **الإجراءات** - زران:
     - **تجديد الآن** - توفير الإضافة/التجديد على الفور (يتجاوز الانتظار للتنفيذ المجدول)
     - **إزالة التجديد التلقائي** - يلغي التجديد التلقائي

عندما لا توجد ActionPlans:

- يظهر الجدول رسالة: "لا يوجد تجديد تلقائي مفعل لهذه الخدمة"
- يمكن للعميل إضافة إضافات متجددة تلقائيًا لتمكين التجديد التلقائي

إدارة ActionPlans

يمكن للموظفين والعملاء إدارة ActionPlans من خلال واجهة المستخدم:

إزالة ActionPlan (إلغاء التجديد التلقائي):

1. انقر على زر "إزالة التجديد التلقائي" في ActionPlansTable
2. تظهر نافذة تأكيد: "هل أنت متأكد أنك تريد إزالة هذا التجديد التلقائي؟"
3. عند التأكيد، يستدعي الواجهة الأمامية: `DELETE /crm/oam/remove_action_plan/{action_plan_id}`
4. يقوم API بإزالة ActionPlan من CGRateS عبر `ocs.Remove_ActionPlan()`
5. يتم تسجيل النشاط: "تمت إزالة ActionPlan {ActionPlanId} من الخدمة {service_id}"
6. تختفي ActionPlan من الجدول

تجديد على الفور (تجديد يدوي):

1. انقر على زر "تجديد الآن" في ActionPlansTable
2. تظهر نافذة تأكيد: "هل أنت متأكد أنك تريد تجديد هذا الآن؟"
3. عند التأكيد، يقوم النظام:
   - باستخراج product_id من ActionPlanId
   - إنشاء وظيفة توفير جديدة لذلك المنتج
   - توفير الإضافة على الفور (تشغيل كتاب التوفير)
   - تتلقى الخدمة فوائد الإضافة دون الانتظار للتجديد المجدول
4. تظهر نافذة حالة التوفير تقدم
5. عند النجاح، يتم تحديث الأرصدة على الفور

استمرارية الخصم على الإضافات عبر التجديدات:

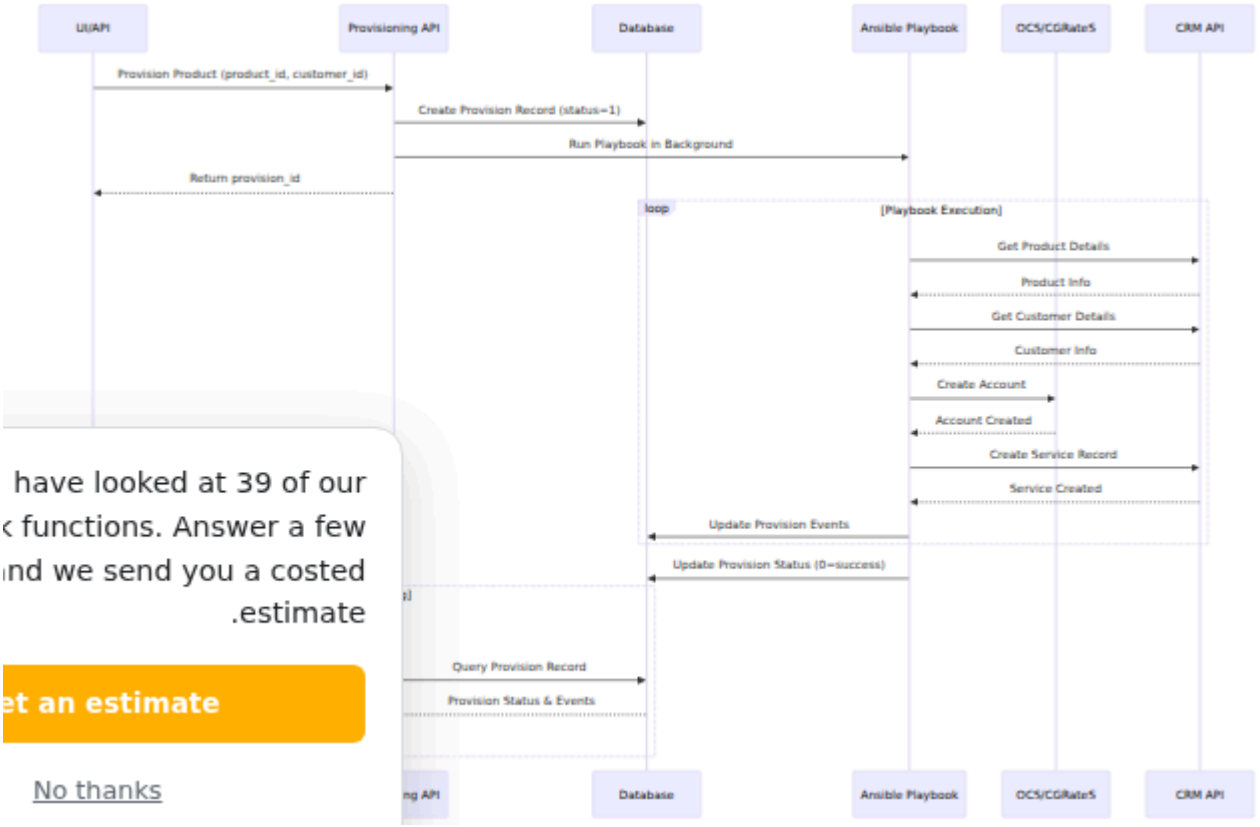
يتم تخزين الخصومات في نقطة البيع المختارة عند إضافة إضافة إلى خدمة بناءً على `(service_id, product_id)` في `service_product_discount` وإعادة تطبيقها عند التجديد التلقائي و"تجديد الآن".

- يقوم الخص

نظام التزويد

يستخدم OmniCRM **Ansible** لأتمتة التزويد، والتكوين، وإلغاء تزويد خدمات العملاء. تم تصميم نظام التزويد ليكون مرنًا، مما يسمح بتدفقات عمل معقدة مع الحفاظ على الاتساق والموثوقية.

انظر أيضًا: تزويد بطاقة SIM لمعلومات مفصلة حول تزويد بطاقات SIM المحمولة، بما في ذلك بطاقات SIM الفعلية و eSIMs.



> **ملاحظة ⓘ**
>
> للحصول على جولة كاملة حول رحلة المنتج إلى الخدمة مع أمثلة مفصلة على دفاتر التشغيل الخاصة بـ Ansible، واستراتيجيات التسعير، والسيناريوهات الواقعية، انظر دليل دورة حياة المنتج الكامل.

نظرة عامة

بإنشاء **وظيفة تزويد** تقوم بتنفيذ OmniCRM عند طلب منتج أو الحاجة إلى تكوين خدمة، يقوم Ansible. دفتر تشغيل أو أكثر من دفاتر تشغيل تتفاعل هذه دفاتر التشغيل مع أنظمة خلفية مختلفة (OCS/CGRateS، معدات الشبكة، APIs، إلخ) بالكامل لتزويد الخدم��.

يدعم نظام التزويد تدفقين رئيسيين:

1. **التزويد القياسي** - يتم تشغيله بواسطة الموظفين أو العملاء من خلال UI/API
2. **التزويد البسيط** - يتم تشغيله بواسطة أنظمة خارجية مثل OCS للعمليات الآلية

قيم حالة التزويد

يمكن أن تحتوي وظائف التزويد والمهام الفردية على الحالات التالية:

- **الحالة 0 (نجاح)** - اكتملت وظيفة التزويد بنجاح
- **الحالة 1 (تشغيل)** - وظيفة التزويد أو المهمة قيد التنفيذ حاليًا
- **الحالة 2 (فشل - حرج)** - حدث فشل حرج أدى إلى فشل التزويد
- **الحالة 3 (فشل - تم تجاهله)** - فشلت مهمة ولكن كانت تحتوي على `ignore_errors: true`، لذا استمر التزويد

إشعارات بريد إلكتروني إلى قائمة إشعارات الفشل OmniCRM عند فشل وظيفة التزويد، يرسل المكونة مع معلومات خطأ مفصلة.

كيف تدفع المنتجات التزويد

تعريف **المنتج** هو المخطط لما يتم تزويده وكيف. عندما يختار المستخدم منت��ا للتزويد، يقوم النظام بقراءة عدة حقول رئيسية من تعريف المنتج لتحديد ما يجب القيام به.

حقول المنتج المستخدمة في التزويد

يحتوي تعريف المنتج على:

- `provisioning_play` - اسم دفتر تشغيل Ansible الذي سيتم تنفيذه (بدون امتداد .yaml)
- `provisioning_json_vars` - سلسلة JSON تحتوي على المتغيرات الافتراضية التي سيتم تمريرها إلى Ansible
- `inventory_items_list` - قائمة بأنواع المخزون التي يجب تعيينها (على سبيل المثال، `['SIM Card', 'Mobile Number']`)
- `product_id`، `product_name`، حقول التسعير - يتم تمريرها تلقائيًا إلى دفتر التشغيل

مثال على تعريف المنتج

```
{
  "product_id": 1,
  "product_slug": "Mobile-SIM",
  "product_name": "Mobile SIM Only",
  "provisioning_play": "play_psim_only",
  "provisioning_json_vars": "{\"iccid\": \"\", \"msisdn\": \"\"}",
  "inventory_items_list": "['SIM Card', 'Mobile Number']",
  "retail_cost": 0,
  "retail_setup_cost": 0,
  "wholesale_cost": 3,
  "wholesale_setup_cost": 1
}
```

من المنتج إلى وظي��ة التزويد

عند بدء التزويد، يقوم النظام:

1. **تحميل دفتر التشغيل** المحدد في `provisioning_play`

   يبحث النظام عن `OmniCRM-API/Provisioners/plays/play_psim_only.yaml`

2. **دمج المتغيرات** من مصادر متعددة في `extra_vars`:

i. **من provisioning_json_vars**: `{"iccid": "", "msisdn": ""}`
ii. **من جسم الطلب**/API: أي متغيرات إضافية يقدمها المستخدم
iii. **من حقول المنتج**: `product_id`، `customer_id`، إلخ.
iv. **من المصادقة**: `access_token` أو الإعداد لـ `refresh_token`

3. **تعيين المخزون** بناءً على `inventory_items_list`

اختيار المخزون UI/API قبل تشغيل دفتر التشغيل، يطلب:

- **بطاقة SIM** - يختار المستخدم بطاقة SIM من المخزون متاحة
- **رقم الهاتف المحمول** - يختار المستخدم رقم هاتف متاح

مع نوع المخزون كمفتاح `extra_vars` يتم إضافة معرفات المخزون المحددة إلى:

```
extra_vars = {
    "product_id": 1,
    "customer_id": 456,
    "SIM Card": 789,        # inventory_id of selected SIM
    "Mobile Number": 101,   # inventory_id of selected
phone number
    "iccid": "",            # From provisioning_json_vars
    "msisdn": "",           # From provisioning_json_vars
    "access_token": "eyJ..."
}
```

4. **تمرير كل شيء إلى Ansible** عبر `hostvars[inventory_hostname]`

داخل دفتر التشغيل، المتغيرات متاحة كـ:

```
- name: Get inventory_id for SIM Card
  set_fact:
    inventory_id_sim_card: "{{
hostvars[inventory_hostname]['SIM Card'] | int }}"
  when: "'SIM Card' in hostvars[inventory_hostname]"
```

كيف تستخدم دفاتر التشغيل متغيرات المخزون

بمجرد أن تحتوي دفتر التشغيل على معرفات المخزون، تسترجع التفاصيل الكاملة للمخزون من API:

```
- name: Get SIM Card Details from Inventory
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_sim_card }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    return_content: yes
  register: sim_card_response

- name: Extract ICCID and IMSI from inventory
  set_fact:
    iccid: "{{ sim_card_response.json.iccid }}"
    imsi: "{{ sim_card_response.json.imsi }}"

- name: Get Phone Number Details from Inventory
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_phone_number }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    return_content: yes
  register: phone_number_response

- name: Extract MSISDN
  set_fact:
    msisdn: "{{ phone_number_response.json.msisdn }}"
```

يمكن بعد ذلك استخدام هذه القيم لـ:

- تزويد بطاقة SIM على HSS مع IMSI
- تكوين رقم الهاتف في نظام الفوترة
- تعيين عناصر المخزون للعميل
- إنشاء سجل الخدمة مع هذه التفاصيل

المحمولة SIM مثال من العالم الحقيقي: تزويد بطاقة

إليك كيف يستخدم بيانات المنتج والمخزون، `play_psim_only.yaml` من:

```
- name: Get Product information from CRM API
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/product/product_id/{{
product_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    return_content: yes
  register: api_response_product

- name: Set package facts from product
  set_fact:
    package_name: "{{ api_response_product.json.product_name }}"
    package_comment: "{{ api_response_product.json.comment }}"
    setup_cost: "{{ api_response_product.json.retail_setup_cost
}}"
    monthly_cost: "{{ api_response_product.json.retail_cost }}"

- name: Set inventory_id_sim_card if SIM Card was selected
  set_fact:
    inventory_id_sim_card: "{{ hostvars[inventory_hostname]['SIM
Card'] | int }}"
  when: "'SIM Card' in hostvars[inventory_hostname]"

- name: Set inventory_id_phone_number if Mobile Number was
selected
  set_fact:
    inventory_id_phone_number: "{{ hostvars[inventory_hostname]
['Mobile Number'] | int }}"
  when: "'Mobile Number' in hostvars[inventory_hostname]"

- name: Get SIM Card details from inventory
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_sim_card }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    return_content: yes
  register: sim_inventory_response

- name: Get Phone Number details from inventory
  uri:
```

```
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_phone_number }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    return_content: yes
  register: phone_inventory_response

- name: Extract values from inventory
  set_fact:
    iccid: "{{ sim_inventory_response.json.iccid }}"
    imsi: "{{ sim_inventory_response.json.imsi }}"
    msisdn: "{{ phone_inventory_response.json.msisdn }}"
    ki: "{{ sim_inventory_response.json.ki }}"
    opc: "{{ sim_inventory_response.json.opc }}"

- name: Provision subscriber on HSS
  uri:
    url: "http://{{ hss_server }}/subscriber/{{ imsi }}"
    method: PUT
    body_format: json
    body:
      {
        "imsi": "{{ imsi }}",
        "msisdn": "{{ msisdn }}",
        "ki": "{{ ki }}",
        "opc": "{{ opc }}",
        "enabled": true
      }
    status_code: 200

- name: Assign inventory to customer
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_sim_card }}"
    method: PATCH
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "customer_id": {{ customer_id }},
        "item_state": "Assigned"
```

```
    }
  status_code: 200
```

هذا يوضح التدفق الكامل:

1. تعريف المنتج يحدد `provisioning_play: "play_psim_only"`
2. يتطلب المنتج `inventory_items_list: ['SIM Card', 'Mobile Number']`
3. يختار المستخدم عناصر المخزون أثناء التزويد
4. يتم تمرير معرفات المخزون إلى دفتر التشغيل كـ `extra_vars`
5. يسترجع دفتر التشغيل تفاصيل المخزون الكاملة من API
6. يستخدم دفتر التشغيل بيانات المخزون لتكوين معدات الشبكة
7. يحدد دفتر التشغيل المخزو�� كمخصص للعميل

التراجع والتنظيف: نمط أفضل الممارسات

أفضل ممارسة حرجة: يجب أن يتعامل نفس دفتر التشغيل مع كل من التراجع عن الفشل في التزويد وإلغاء التزويد المتعمد باستخدام هيكل `block` و `rescue` الخاص بـ Ansible.

هيكل دفتر التشغيل

من `play_psim_only.yaml`:

```yaml
- name: OmniCore Service Provisioning 2024
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Main block
      block:
        # --- مهام التزويد ---
        - name: Get Product information
          uri: ...

        - name: Create account in OCS
          uri: ...

        - name: Provision subscriber on HSS
          uri: ...

        - name: Create service record
          uri: ...

        # ... المزيد من مهام التزويد ...

      rescue:
        # --- مهام التنظيف ---
        # يتم تشغيل هذا القسم عندما:
        # 1. تفشل أي مهمة في الكتلة (التراجع)
        # 2. action == "deprovision" (تنظيف متعمد)

        - name: Get Inventory items linked to this service
          uri:
            url: "{{ crm_config.crm.base_url
}}/crm/inventory/customer_id/{{ customer_id }}"
            method: GET
          register: inventory_api_response
          ignore_errors: True

        - name: Return inventory to pool
          uri:
            url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ item.inventory_id }}"
            method: PATCH
            body_format: json
```

```
      body:
        service_id: null
        customer_id: null
        item_state: "Used"
    with_items: "{{ inventory_api_response.json.data }}"
    ignore_errors: True

  - name: Delete Account from Charging
    uri:
      url: "http://{{ crm_config.ocs.OCS }}/jsonrpc"
      method: POST
      body:
        {
          "method": "ApierV1.RemoveAccount",
          "params": [{
              "Tenant": "{{ crm_config.ocs.ocsTenant }}",
              "Account": "{{ service_uuid }}"
          }]
        }
    ignore_errors: True

  - name: Delete Attribute Profile
    uri:
      url: "http://{{ crm_config.ocs.OCS }}/jsonrpc"
      method: POST
      body:
        {
          "method": "APIerSv1.RemoveAttributeProfile",
          "params": [{
              "ID": "ATTR_ACCOUNT_{{ service_uuid }}"
          }]
        }
    ignore_errors: True

  - name: Remove Resource Profile
    uri: ...
    ignore_errors: True

  - name: Remove Filters
    uri: ...
    ignore_errors: True

  - name: Deprovision Subscriber from HSS
    uri:
```

```
            url: "{{ item.key }}/subscriber/{{
item.value.subscriber_id }}"
            method: DELETE
          loop: "{{ hss_subscriber_data | dict2items }}"
          ignore_errors: True
          when:
            - deprovision_subscriber | bool == true

        - name: Patch Subscriber to Dormant State
          uri:
            url: "{{ item.key }}/subscriber/{{
item.value.subscriber_id }}"
            method: PATCH
            body:
              {
                "enabled": true,
                "msisdn": "9999{{ imsi[-10:] }}",  # رقم وهمي
                "ue_ambr_dl": 9999999,              # مرتفع بشكل غير
قابل للاستخدام
                "ue_ambr_ul": 9999999
              }
          loop: "{{ hss_subscriber_data | dict2items }}"
          when:
            - deprovision_subscriber | default(false) | bool ==
false

        # التأكيد النهائي يحدد النجاح أو الفشل
        - name: Set status to "Success" if Manual deprovision /
Fail if failed provision
          assert:
            that:
              - action == "deprovision"
```

لماذا يعتبر هذا النمط أفضل ممارسة

1. عدم تكرار الشيفرة

تتعامل نفس مهام التنظيف مع كلا السيناريوهين:

- إذا فشلت أي مهمة في `block`، يتم تنفيذ قسم `rescue` **فشل التزويد (التراجع)**: تلقائيًا

- **إلغاء التزويد المتعمد**: عند استدعائه مع `action: "deprovision"`، ينتقل دفتر التشغيل مباشرة إلى `rescue`

2. ضمان التنظيف الكامل

عندما يفشل التزويد في منتصف الطريق، يضمن قسم الإنقاذ:

- حذف جميع حسابات OCS التي تم إنشاؤها
- إزالة جميع إدخالات معدات الشبكة المكونة
- إعادة المخزون المعين إلى المجموعة
- حذف أو تعيين مشتركي HSS إلى حالة خاملة
- عدم بقاء أي تزويد جزئي في أي نظام

هذا يمنع الموارد "اليتيمة" التي:

- تستهلك المخزون دون أن يتم تتبعها
- تنشئ حسابات فوترة غير مرتبطة بالخدمات
- تسبب الارتباك أثناء استكشاف الأخطاء وإصلاحها
- تهدر موارد الشبكة

3. التعامل مع الفشل بشكل سلس مع ignore_errors

لاحظ أن كل مهمة تنظيف تستخدم `ignore_errors: True`. هذا مقصود لأن:

- أثناء التراجع، قد لا تكون بعض الموارد قد تم إنشاؤها بعد
- نريد محاولة جميع مهام التنظيف حتى لو فشلت بعضها
- يحدد التأكيد النهائي النجاح/الفشل العام

على سبيل المثال، إذا فشل التزويد عند "إنشاء حساب في OCS"، ستحاول عملية التنظيف:

- حذف حساب OCS (ستفشل، لكن سيتم تجاهلها)
- إزالة ملفات تعريف السمات (ستفشل، لكن سيتم تجاهلها)
- إعادة المخزون (تنجح)
- حذف مشترك HSS (قد لا يوجد، يتم تجاهلها)

4. تمييز إلغاء التزويد عن التراجع

التأكيد النهائي في نهاية `rescue` ذكي:

```
- name: Set status to "Success" if Manual deprovision / Fail if
failed provision
  assert:
    that:
      - action == "deprovision"
```

هذا يعني:

- **إذا كانت ``action == "deprovision"``**: يمر التأكيد، ينجح دفتر التشغيل (الحالة 0)
- **إذا كانت ``action`` غير محددة أو != "deprovision"**: يفشل التأكيد، يفشل دفتر التشغيل (الحالة 2)

لذا فإن نفس كود التنظيف ينتج عنه حالات وظائف تزويد مختلفة اعتمادًا على النية.

5. تنظيف مشروط بناءً على نوع الخدمة

تستخدم بعض مهام التنظيف شروطًا للتعامل مع سيناريوهات مختلفة:

```
- name: Deprovision Subscriber from HSS
  uri: ...
  when:
    - deprovision_subscriber | bool == true

- name: Patch Subscriber to Dormant State
  uri: ...
  when:
    - deprovision_subscriber | default(false) | bool == false
```

هذا يسمح بتنظيف مرن:

- **الحذف الكامل**: عندما تكون بطاقات SIM مخصصة للعملاء (`deprovision_subscriber: true`)
- **الحالة الخامدة**: عندما تكون بطاقات SIM قابلة لإعادة الاستخدام ويجب أن تبقى في HSS (`deprovision_subscriber: false`)

كيفية استخدام هذا النمط

للتزويد:

```
{
  "product_id": 1,
  "customer_id": 456,
  "provisioning_play": "play_psim_only"
}
```

إذا فشل التزويد، يحد`rescue`ث التراجع التلقائي عبر ��.

لإلغاء التزويد:

```
{
  "service_id": 123,
  "service_uuid": "Service_abc123",
  "action": "deprovision",
  "provisioning_play": "play_psim_only"
}
```

يتخطى دفتر التشغيل مباشرة إلى قسم `rescue`، وينجح، وينفذ جميع مهام التنظيف.

ملخص الفوائد

مصدر واحد للحقيقة: يتعامل دفتر تشغيل واحد مع التزويد وإلغاء التزويد **عمليات ذرية**: إما أن تكون مخصصة بالكامل أو يتم تنظيفها بالكامل **لا موارد يتيمة**: التزويد الفاشل لا يترك أي أثر **صيانة أسهل**: التغييرات في منطق التزويد تنطبق تلقائيًا على التنظيف **تقليل الأخطاء**: لا فرصة لتزامن كود التزويد وإلغاء التزويد **قابل للاختبار**: يمكن اختبار منطق إلغاء التزويد عن طريق التشغيل مع `action: "deprovision"`

يجب اتباع هذا النمط في جميع دفاتر تشغيل التزويد لضمان الموثوقية والاتساق.

تجاوز متغيرات المنتج

يمكن تجاوز `provisioning_json_vars` في وقت التزويد. على سبيل المثال، قد يعرف منتج ما:

```
{
  "provisioning_json_vars": "{\"monthly_cost\": 50,
\"data_limit_gb\": 100}"
}
```

ولكن عند التزويد، يمكنك تجاوز هذه القيم:

```
{
  "product_id": 1,
  "customer_id": 456,
  "monthly_cost": 45,
  "data_limit_gb": 150
}
```

القيم المتجاوزة. وهذا يسمح بـ `extra_vars` ستستخدم المتغيرات المدمجة:

- تسعير مخصص للعملاء المحددين
- حدود بيانات مختلفة بناءً على العروض الترويجية
- الاختبار بمعلمات مختلفة دون تعديل المنتج

المنتجات بدون مخزون

لا تتطلب جميع المنتجات مخزونًا. على سبيل المثال، قد يحتوي ملحق بيانات أو تبديل ميزة على:

```
{
  "product_id": 10,
  "product_name": "Extra 10GB Data",
  "provisioning_play": "play_local_data_addon",
  "provisioning_json_vars": "{\"data_gb\": 10}",
  "inventory_items_list": "[]"
}
```

في هذه الحالة، يتلقى دفتر التشغيل:

```
extra_vars = {
    "product_id": 10,
    "customer_id": 456,
    "service_id": 123,  # الخدمة التي سيتم إضافة البيانات إليها
    "data_gb": 10,
    "access_token": "eyJ..."
}
```

يضيف دفتر التشغيل ببساطة البيانات إلى الخدمة الموجودة دون الحاجة إلى أي عناصر مخزون.

سير عمل التزويد القياسي

يتم بدء التزويد القياسي عندما:

- يضيف موظف خدمة إلى عميل من واجهة المستخدم
- يطلب عميل خدمة من خلال بوابة الخدمة الذاتية
- يتم استدعاء API مباشرة مع `PUT /crm/provision/`

عندما تضغط على "تزويد"

إليك التدفق الكامل الذي يحدث عندما ينقر المستخدم على زر "تزويد":

1. تعرض واجهة المستخدم اختيار المنتج

يختار المستخدم منتجًا من كتالوج المنتجات. يحتوي المنتج على:

- `provisioning_play` - أي دفتر تشغيل Ansible يجب تشغيله
- `inventory_items_list` - المخزون المطلوب (على سبيل المثال، `['SIM Card', 'Mobile Number']`)
- `provisioning_json_vars` - المتغيرات الافتراضية

2. مختار المخزون (إذا لزم الأمر)

إذا كانت `inventory_items_list` غير فارغة، تظهر نافذة منبثقة تعرض قوائم منسدلة لكل نوع مخزون. يجب على المستخدم اختيار عناصر المخزون المتاحة قبل المتابعة.

3. تم النقر على زر التزويد

ترسل JavaScript طلب `PUT /crm/provision/`:

```
PUT /crm/provision/
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "product_id": 42,
  "customer_id": 123,
  "SIM Card": 5001,
  "Mobile Number": 5002
}
```

4. يستقبل API الطلب

نقطة نهاية التزويد:

- تتحقق من المصادقة (رمز Bearer، مفتاح API، قائمة بيضاء IP أو)
- تتحقق من أن المستخدم لديه إذن `CREATE_PROVISION`
- تستخرج `initiating_user` من الرمز
- تحمل تعريف المنتج من قاعدة البيانات
- تسترجع مسار دفتر التشغيل: `OmniCRM-API/Provisioners/plays/play_psim_only.yaml`

5. دمج المتغيرات

يجمع النظام المتغيرات من مصادر متعددة:

```
# من المنتج
product_vars = json.loads(product['provisioning_json_vars'])
# من جسم الطلب
request_vars = request_body_json
# المضافة من النظام
system_vars = {
    'product_id': 42,
    'customer_id': 123,
    'access_token': auth_token,  # انظر قسم المصادقة أدناه
    'initiating_user': 7
}
# المدمجة النهائية
extra_vars = {**product_vars, **request_vars, **system_vars}
```

6. إنشاء سجل التزويد

تم إنشاء سجل قاعدة البيانات مع الحالة 1 (تشغيل):

```
provision = {
    'provision_id': 456,
    'customer_id': 123,
    'product_id': 42,
    'provisioning_play': 'play_psim_only',
    'provisioning_json_vars': json.dumps(extra_vars),
    'provisioning_status': 1,  # تشغيل
    'task_count': 85,
    'initiating_user': 7,
    'created': '2025-01-10T14:30:00Z'
}
```

7. يتم إنشاء خيط في الخلفية

ينشئ النظام خيطًا في الخلفية لتنفيذ دفتر التشغيل بشكل غير متزامن:

- دفتر التشغيل: `play_psim_only.yaml`
- المتغيرات: `extra_vars` المدمجة
- معرف التزويد: 456 (لتتبع الحالة)
- رمز التحديث: يتم تمريره لتحديث الرمز أثناء التنفيذ

8. تعيد API الاستجابة على الفور

تُعاد الاستجابة إلى واجهة المستخدم مع `provision_id`:

```
{
  "provision_id": 456,
  "provisioning_status": 1,
  "message": "Provisioning job created"
}
```

9. تبدأ واجهة المستخدم في الاستعلام عن التحديثات

تبدأ واجهة المستخدم في الاستعلام عن `GET /crm/provision/provision_id/456` كل 3 ثوانٍ للتحقق من الحالة. تتضمن الاستجابة:

```
{
  "provision_id": 456,
  "provisioning_status": 1,
  "task_count": 12,
  "provisioning_result_json": [
    {
      "event_number": 1,
      "event_name": "Get Product information from CRM API",
      "provisioning_status": 0,
      "timestamp": "2024-01-15T10:30:05"
    },
    {
      "event_number": 2,
      "event_name": "Assign SIM Card from inventory",
      "provisioning_status": 1,
      "timestamp": "2024-01-15T10:30:07"
    }
  ]
}
```


10. تنفذ Ansible في الخلفية

تشغل دفتر التشغيل المهام بشكل متسلسل:

- كل إكمال مهمة ينشئ سجل في قاعدة البيانات `Provision_Event`
- تتضمن الحدث: اسم المهمة، الحالة (0=نجاح، 2=فشل، 3=فشل ولكن تم تجاهله)، نتيجة JSON
- تعرض واجهة المستخدم تقدمًا في الوقت الحقيقي يظهر المهام المكتملة والمهمة التي يتم تشغيلها حاليًا
- تظهر المهام الفاشلة رسائل خطأ في تفاصيل الحدث

التتبع في واجهة المستخدم:

بينما يكون التزويد قيد التشغيل (الحالة 1)، يمكن للمستخدمين عرض:

- **صفحة تفاصيل الخدمة** - تظهر شارة حالة التزويد (تشغيل/نجاح/فشل)
- **سجل النشاط** - يسرد جميع أحداث التزويد مع الطوابع الزمنية
- **عرض تفاصيل التزويد** - يظهر تقدم المهمة تلو الأخرى مع توسيع/طي للتفاصيل

مثال على العرض:

حالة التزويد: تشغيل (8 من 12 مهمة مكتملة)

✓ Get Product information from CRM API ✓ Fetch Customer details ✓ Assign SIM Card from inventory (ICCID: 8991101200003204510) ✓ Assign Mobile Number (555-0123) ⟳ Create account in OCS/CGRateS (قيد التقدم...) Configure network policies Create service record ...

11. يكتمل التزويد

تُحدد الحالة النهائية:

- `provisioning_status: 0` - نجاح
- `provisioning_status: 2` - فشل (خطأ حرج)

تتوقف واجهة المستخدم عن الاستعلام وتعرض النتيجة:

- **النجاح**: علامة تحقق خضراء، الخدمة محددة نشطة، يمكن للمستخدم عرض تفاصيل الخدمة
- **الفشل**: علامة X حمراء، يتم عرض رسالة خطأ، خيار لإعادة المحاولة أو الاتصال بالدعم
- **إشعار البريد الإلكتروني**: إذا حدث فشل، يتم إرسال بريد إلكتروني إلى `provisioning.failure_list` في التكوين

المصادقة والتفويض

تتبع المستخدم

تتبع كل وظيفة تزويد المستخدم الذي بدأها:

- **بدء المستخدم**: يتم تعيين حقل `initiating_user` إلى معرف المستخدم من رمز JWT الخاص به
- **مصادقة مفتاح API**: تستخدم معرف أول مستخدم إداري
- **مصادقة قائمة بيضاء IP**: تستخدم معرف أول مستخدم إداري

التحقق من الأذونات

يتحقق النظام من الأذونات قبل السماح بالتزويد:

- يحتاج الموظفون إلى إذن `CREATE_PROVISION`

- إذن) يمكن للعملاء فقط تزويد الخدمات لحسابهم الخاص (`VIEW_OWN_PROVISION`)

كيف تصادق Ansible مع API CRM

تحتاج دفاتر تشغيل Ansible إلى إجراء مكالمات API إلى CRM مصدقة مرة أخرى (للحصول على تفاصيل المنتج، إنشاء الخدمات، تحديث المخزون، إلخ). **رموز** تتم معالجة المصادقة من خلال **Bearer** المرسلة إلى دفتر التشغيل.

يعتمد مصدر التزويد API على طريقة المصادقة المستخدمة لاستدعاء `access_token`:

الطريقة 1: تسجيل دخول المستخدم (رمز Bearer)

عندما يقوم المستخدم بتسجيل الدخول عبر واجهة الويب:

1. يقوم المستخدم بالمصادقة: `POST /crm/auth/login`
2. يتلقى رمز JWT `access_token` (قصير العمر، 15-30 دقيقة) و `refresh_token` (طويل العمر)
3. يقوم بإجراء طلب التزويد مع رمز Bearer في الرأس
4. يستخرج API التزويد الرمز من رأس `Authorization: Bearer ...`
5. يخزن الرمز للاستخدام أثناء التزويد
6. يمرره إلى Ansible كمتغير `access_token`

تدفق التنفيذ:

يستخرج النظام رمز Bearer من رأس المصادقة، يتحقق منه ويفك تشفيره، ثم يخزنه للاستخدام أثناء التزويد. ثم يتم تمريره إلى دفتر التشغيل كمتغير إضافي حتى يتمكن من مصادقة مكالمات API اللاحقة.

الطريقة 2: مفتاح API (رأس X-API-KEY)

لأنظمة الأتمتة التي تستخدم مفاتيح API:

1. يقوم النظام بإجراء الطلب: `PUT /crm/provision/` مع `X-API-KEY: your-api-key...` الرأس
2. يتحقق API التزويد من مفتاح API مقابل `crm_config.yaml`
3. **ينشئ رمز JWT جديد في الوقت الفعلي** لأول مستخدم إداري
4. يخزن الرمز للاستخدام في التزويد
5. يمرره إلى Ansible

لماذا إنشاء رمز؟

مفاتيح API تتوقع دفاتر التشغيل استدعاء نقاط نهاية API باستخدام JWTs. وليست مصادقة JWT. لذا:

- API تحقق من مفتاح
- JWT مؤقت قم بإنشاء `admin`�إذا كان صالحًا وله دور
- JWT استخدم معرف أول مستخدم إداري كموضوع
- API مصدقة يسمح الرمز لدفتر التشغيل بإجراء مكالمات

تدفق التنفيذ:

يتحقق النظام من مفتاح API مقابل المفاتيح المكونة. إذا كان صالحًا وله دور إداري، فإنه ينشئ JWT مؤقتًا باستخدام هوية مستخدم إداري، ثم يخزنه للاستخدام في التزويد.

الطريقة 3: قائمة بيضاء IP

لأنظمة داخلية موثوقة على الشبكات الخاصة:

1. يقوم النظام بإجراء الطلب من IP (على سبيل المثال، 192.168.1.100) مدرج في القائمة البيضاء
2. يتحقق API من التزويد من IP العميل مقابل `ip_whitelist` في `crm_config.yaml`
3. **ينشئ رمز** **JWT** لأول مستخدم إداري إذا كان مدرجًا في القائمة البيضاء، **جديدًا**
4. يخزن الرمز للاستخدام في التزويد
5. Ansible يمرره إلى

تدفق التنفيذ:

يتحقق النظام من IP العميل مقابل القائمة البيضاء. إذا كان مسموحًا، فإنه ينشئ JWT لمستخدم إداري ويخزنه للاستخ��ام في التزويد.

استخدام الرمز في دفاتر التشغيل

في دفتر التشغيل الرمز API تتضمن كل مكالمة:

```
- name: Get Product Details
  uri:
    url: "http://localhost:5000/crm/product/product_id/{{
product_id }}"
    headers:
      Authorization: "Bearer {{ access_token }}"

- name: Create Service Record
  uri:
    url: "http://localhost:5000/crm/service/"
    method: PUT
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      customer_id: "{{ customer_id }}"
      service_name: "Mobile Service"
```

انتهاء صلاحية الرمز والتحديث

(دقيقة انتهاء الصلاحية 15-30) `access_token` قد تتجاوز دفاتر التشغيل الطويلة (5-10 دقائق). بالنسبة للتزويدات التي يبدأها المستخدم، يمرر النظام كلاً من `access_token` و `refresh_token`:

```
refresh_token = request.cookies.get('refresh_token')
run_playbook(playbook_path, extra_vars, provision_id,
refresh_token=refresh_token)
```

إذا انتهت صلاحية `access_token`، يمكن لمشغل دفتر التشغيل:

1. اكتشاف استجابة 401 غير مصرح بها
2. استدعاء `POST /crm/auth/refresh` مع `refresh_token`
3. استلام `access_token` جديد
4. إعادة محاولة الطلب الفاشل

يمكن أن تحتوي الرموز المولدة على انتهاء صلاحية، بالنسبة لمصادقة مفتاح API/IP whitelist أطول (1-2 ساعة) لأن هذه أنظمة مؤتمتة موثوقة.

عملية التزويد

1. **إنشاء الوظيفة**

   عند استلام طلب التزويد، يقوم النظام:

   - يتحقق من الطلب ويتحقق من الأذونات
   - يحمل دفتر التشغيل الخاص بـ Ansible المحدد في تعريف المنتج
   - ينشئ سجل `Provision` (تشغيل) في قاعدة البيانات مع الحالة 1
   - يستخرج المتغيرات من تعريف المنتج وجسم الطلب
   - يلتقط رموز المصادقة للوصول إلى API

2. **معالجة الرموز**

   تحتاج دفاتر تشغيل Ansible إلى المصادقة مع API CRM لاسترجاع البيانات وإجراء التغييرات. يتعامل نظام التزويد مع هذا بطريقتين:

   - **رمز Bearer (JWT)**: بالنسبة للتزويد الذي يبدأه المستخدم، يتم استخدام `refresh_token` من الطلب لإنشاء رموز وصول جديدة ��ثناء تنفيذ دفتر التشغيل
   - **مفتاح API/IP Auth**: بالنسبة للأنظمة المؤتمتة، يتم تمرير `access_token` مباشرة إلى دفتر التشغيل

3. **التنفيذ في الخلفية**

   يعمل دفتر التشغيل في خيط خلفي. يتيح ذلك لـ API العودة على الفور بينما يستمر التزويد بشكل غير متزامن.

   أثناء التنفيذ:

   - ينشئ سجل `Provision_Event` كل إكمال/فشل مهمة
   - يقوم معالج الحدث بمراقبة الفشل الحرج مقابل الفشل المتجاهل
   - يتم كتابة تحديثات الحالة في الوقت الحقيقي إلى قاعدة البيانات
   - يمكن لواجهة المستخدم الاستعلام عن التحديثات عبر `GET /crm/provision/provision_id/<id>`

4. **تنفيذ دفتر التشغيل**

   عادةً ما تقوم دفتر تشغيل Ansible بتنفيذ هذه العمليات:

- يسترجع معلومات المنتج من API
- يسترجع معلومات العميل من API
- يعين عناصر المخزون (بطاقات SIM، عناوين IP، إلخ، أرقام الهواتف.)
- ينشئ حسابات في OCS/OCS
- ي��وم بتكوين معدات الشبكة
- ينشئ سجل الخدمة في API CRM
- يضيف معاملات تكلفة الإعداد
- يرسل رسائل بريد إلكتروني/SMS ترحيبية للعملاء

5. **معالجة الأخطاء**

   تستخدم دفاتر تشغيل Ansible أقسام `block` و `rescue` للتراجع:

   - إذا فشلت مهمة حرجة، يقوم قسم الإنقاذ بإزالة التزويد الجزئي
   - يتم وضع علامات على المهام التي تحتوي على `ignore_errors: true` كحالة 3 ولا تفشل الوظيفة
   - تؤدي الأخطاء الفتاكة (فشل الاتصال، أخطاء بناء جملة YAML) إلى إنشاء حدث خطأ خاص مع معلومات تصحيح الأخطاء

مثال: دفتر تشغيل التزويد القياسي

إليك مثال من `play_simple_service.yaml`:

```
- name: Simple Provisioning Play
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Main block
      block:
      - name: Get Product information from CRM API
        uri:
          url: "http://localhost:5000/crm/product/product_id/{{
product_id }}"
          method: GET
          headers:
            Authorization: "Bearer {{ access_token }}"
          return_content: yes
          validate_certs: no
        register: api_response_product

      - name: Set package facts
        set_fact:
          package_name: "{{ api_response_product.json.product_name
}}"
          setup_cost: "{{
api_response_product.json.retail_setup_cost }}"
          monthly_cost: "{{ api_response_product.json.retail_cost
}}"

      - name: Generate Service UUID
        set_fact:
          service_uuid: "Service_{{ 99999999 | random | to_uuid
}}"

      - name: Create account in OCS
        uri:
          url: "http://{{ crm_config.ocs.OCS }}/jsonrpc"
          method: POST
          body_format: json
          body:
            {
              "method": "ApierV2.SetAccount",
              "params": [{
                  "Tenant": "{{ crm_config.ocs.ocsTenant }}",
```

```
                  "Account": "{{ service_uuid }}",
                  "ActionPlanIds": [],
                  "ExtraOptions": { "AllowNegative": false,
"Disabled": false }
                }]
              }
            status_code: 200
          register: response

      - name: Add Service via API
          uri:
            url: "http://localhost:5000/crm/service/"
            method: PUT
            body_format: json
            headers:
              Authorization: "Bearer {{ access_token }}"
            body:
              {
                "customer_id": "{{ customer_id }}",
                "product_id": "{{ product_id }}",
                "service_name": "Service: {{ service_uuid }}",
                "service_uuid": "{{ service_uuid }}",
                "service_status": "Active",
                "retail_cost": "{{ monthly_cost | float }}"
              }
            status_code: 200
          register: service_creation_response

      - name: Add Setup Cost Transaction
          uri:
            url: "http://localhost:5000/crm/transaction/"
            method: PUT
            headers:
              Authorization: "Bearer {{ access_token }}"
            body_format: json
            body:
              {
                "customer_id": {{ customer_id | int }},
                "service_id": {{
service_creation_response.json.service_id | int }},
                "title": "{{ package_name }} - Setup Costs",
                "retail_cost": "{{ setup_cost | float }}"
              }
          register: api_response_transaction
```

```
    rescue:
      - name: Remove account in OCS
        uri:
          url: "http://{{ crm_config.ocs.OCS }}/jsonrpc"
          method: POST
          body_format: json
          body:
            {
              "method": "ApierV2.RemoveAccount",
              "params": [{
                  "Tenant": "{{ crm_config.ocs.ocsTenant }}",
                  "Account": "{{ service_uuid }}"
              }]
            }
          status_code: 200

      - name: Fail the provision
        assert:
          that:
            - false
```

يوضح هذا دفتر التشغيل التدفق النموذجي:

1. استرجاع تفاصيل المنتج من API CRM
2. خدمة فريدة UUID توليد
3. إنشاء حساب الفوترة في OCS
4. إنشاء سجل الخدمة عبر API CRM
5. إضافة معاملات تكلفة الإعداد
6. إذا فشل أي شيء، يقوم قسم `rescue` بإزالة حساب OCS

سير عمل التزويد البسيط

تم تصميم التزويد البسيط للأنظمة المؤتمتة التي تحتاج إلى تشغيل التزويد دون تفاعل المستخدم. أكثر حالات الاستخدام شيوعًا هي تشغيل OCS لتزويد الملحقات عبر ActionPlans.

نقاط نهاية التزويد البسيط

يوفر OmniCRM نقطتي نهاية للتزويد البسيط:

- `POST /crm/provision/simple_provision_addon/service_id/<id>/product_id/<id>`

  للتزويد الآلي للملحقات (على سبيل المثال، الرسوم المتكررة، الشحن التلقائي)

- `POST /crm/provision/simple_provision_addon_recharge/service_id/<id>/product_id/<id>`

  لعمليات الشحن السريعة التي تحتاج إلى ردود فعل فورية

المصادقة للتزويد البسيط

تستخدم نقاط نهاية التزويد البسيط **قائمة بيضاء** **IP** أو **مفاتيح** **API** للمصادقة:

- يتم التحقق من IP مصدر الطلب مقابل `ip_whitelist` في `crm_config.yaml`
- يمكن تقديم مفتاح API أو من `api_keys` في `crm_config.yaml`
- يتم إنشاء رمز وصول ويتم تمريره إلى دفتر التشغيل

مثال: رد نداء ActionPlan من OCS

يمكن تكوين OCS لاستدعاء نقطة نهاية التزويد البسيط عند تنفيذ الإجراءات المتكررة:

```
{
  "method": "ApierV1.SetActionPlan",
  "params": [{
    "Id": "ActionPlan_Service123_Monthly_Charge",
    "ActionsId": "Action_Service123_Add_Monthly_Data",
    "Timing": {
      "Years": [],
      "Months": [],
      "MonthDays": [1],
      "Time": "00:00:00Z"
    },
    "Weight": 10,
    "ActionTriggers": [
      {
        "ThresholdType": "*min_event_counter",
        "ThresholdValue": 1,
        "ActionsID": "Action_Service123_HTTP_Callback"
      }
    ]
  }]
}
```

تقوم الإجراء بإجراء POST HTTP إلى:

هذا يشغل دفتر التشغيل المرتبط (على سبيل المثال، `play_topup_no_charge.yaml`) الذي يضيف البيانات/الرصيد إلى الخدمة.

مثال: دفتر تشغيل الشحن البسيط

من `play_topup_monetary.yaml`:

```yaml
- name: Mobile Topup Monetary - 2024
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Get Product information from CRM API
      uri:
        url: "http://localhost:5000/crm/product/product_id/{{
product_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
      register: api_response_product

    - name: Get Service information from CRM API
      uri:
        url: "http://localhost:5000/crm/service/service_id/{{
service_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
      register: api_response_service

    - name: Set service facts
      set_fact:
        service_uuid: "{{ api_response_service.json.service_uuid
}}"
        customer_id: "{{ api_response_service.json.customer_id }}"
        package_name: "{{ api_response_product.json.product_name
}}"
        monthly_cost: "{{ api_response_product.json.retail_cost
}}"

    - name: Get Customer Payment Method
      uri:
        url: "http://localhost:5000/api/payments/methods/default?
customer_id={{ customer_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
```

```
      return_content: yes
    register: api_response_payment_method

- name: Charge customer
  uri:
    url: "http://localhost:5000/api/payments/charge"
    method: POST
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "customer_id": "{{ customer_id | int }}",
        "amount": "{{ monthly_cost }}",
        "currency": "USD",
        "metadata": {
          "description": "{{ package_name }} topup",
          "service_id": "{{ service_id | int }}"
        }
      }
  register: api_response_payment

- name: Add monetary balance to OCS
  uri:
    url: "http://{{ crm_config.ocs.OCS }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
            "Tenant": "{{ crm_config.ocs.ocsTenant }}",
            "Account": "{{ service_uuid }}",
            "BalanceType": "*monetary",
            "Balance": {
                "Value": "{{ monthly_cost | float * 100 }}",
                "ExpiryTime": "+4320h"
            }
        }]
      }
    status_code: 200

- name: Add Transaction to CRM
  uri:
```

```
        url: "http://localhost:5000/crm/transaction/"
        method: PUT
        headers:
          Authorization: "Bearer {{ access_token }}"
        body_format: json
        body:
          {
            "customer_id": {{ customer_id | int }},
            "service_id": {{ service_id | int }},
            "title": "{{ package_name }}",
            "retail_cost": "{{ monthly_cost | float }}"
          }

    - name: Send Notification SMS
      uri:
        url: "http://sms-gateway/SMS/plaintext/{{ api_key }}"
        method: POST
        body_format: json
        body:
          {
            "source_msisdn": "YourCompany",
            "destinatination_msisdn": "{{ customer_phone }}",
            "message_body": "Thanks for topping up {{ monthly_cost
}}!"
          }
        status_code: 201
      ignore_errors: True
```

يقوم هذا دفتر التشغيل بـ:

1. الحصول على تفاصيل الخدمة والمنتج من API
2. استرجاع طريقة دفع العميل
3. شحن العميل عبر API Stripe
4. إضافة رصيد نقدي إلى OCS
5. تسجيل المعاملة في CRM
6. إرسال رسالة SMS تأكيد (مع `ignore_errors: True` حتى لا تفشل الأخطاء الوظيفة)

سلاسل التزويد

سلاسل التزويد. تنفذ سلسلة OmniCRM لمنتجات معقدة تتطلب خطوات تزويد متعددة، يدعم عدة دفاتر تشغيل بشكل متسلسل، تمرر السياق بينها.

حالة استخدام نموذجية: خدمة مجمعة تقوم بتزويد:

1. خدمة الإنترنت الأساسية (تنشئ سجل الخدمة الأساسي)
2. ملحق IPTV (يستخدم `service_id` من الخطوة 1)
3. ملحق IP ثابت (يستخدم `service_id` من الخطوة 1)

تقوم خدمة التزويد تلقائيًا:

- استعلام قاعدة البيانات عن `service_id` الذي تم إنشاؤه بواسطة أول دفتر تشغيل
- حقنها في `extra_vars` لدفاتر التشغيل اللاحقة
- تتبع كل دفتر تشغيل كسجل `Provision` منفصل

أسباب الفشل وتصحيح الأخطاء

عندما يفشل التزويد، يلتقط النظام معلومات مفصلة للمساعدة في تشخيص المشكلة.

سيناريوهات الفشل الشائعة

فشل المهام الحرجة (الحالة 2)

تسبب هذه الفشل في وظيفة التزويد بالكامل:

- مكالمات API التي تعيد رموز حالة غير متوقعة
- فشل التأكيدات (على سبيل المثال، `assert: that: response.status == 200`)
- العناصر المطلوبة المفقودة من المخزون
- معدات الشبكة غير قابلة للوصول
- بيانات اعتماد غير صالحة أو رموز منتهية الصلاحية
- أخطاء OCS/OCS

الفشل المتجاهل (الحالة 3)

تُسجل هذه ولكن لا تفشل الوظيفة:

- فشل الإشعارات الاختيارية عبر SMS/البريد الإلكتروني
- عمليات البحث عن البيانات غير الحرجة (المعلمة بـ `ignore_errors: True`)
- عمليات التنظيف أثناء إلغاء التزويد

الأخطاء الفتاكة

تمنع هذه دفاتر التشغيل من التشغيل على الإطلاق:

- أخطاء بناء جملة YAML في دفتر التشغيل
- متغيرات غير معرفة في دفتر التشغيل
- ملفات دفتر التشغيل المفقودة
- فشل الاتصال مع وحدة التحكم Ansible

عندما يحدث خطأ فتاك، ينشئ النظام حدث خطأ خاص يحتوي على:

- رمز خروج Ansible
- stdout الكامل (يحتوي على تفاصيل أخطاء البناء)
- stderr الكامل (يحتوي على أخطاء وقت التشغيل)
- قائمة بالأسباب الشائعة لهذا النوع من الفشل
- جميع المتغيرات المرسلة إلى دفتر التشغيل

رسائل البريد الإلكتروني لإشعار الأخطاء

عندما يفشل التزويد (الحالة 2)، يتم إرسال بريد إلكتروني تلقائيًا إلى قائمة إشعارات الفشل المكونة (`provisioning.failure_list` في `crm_config.yaml`).

يتضمن البريد الإلكتروني:

- معلومات العميل
- تفاصيل المنتج/الخدمة
- نتائج المهام الملونة:
  - **أخضر**: المهام الناجحة
  - **برتقالي**: المهام الفاشلة ولكن تم تجاهلها
  - **أحمر**: الفشل الحرج
- بالنسبة للفشل الحرج: إخراج تصحيح كامل بما في ذلك أجسام الطلب/الاستجابة

- إخراج: رسائل الخطأ، والأسباب الشائعة، Ansible بالنسبة للأخطاء الفتاكة

مراقبة وظائف التزويد

يوفر OmniCRM مقاييس شاملة لمراقبة أداء التزويد. للحصول على تفاصيل كاملة حول مقاييس التزويد بما في ذلك مدة الوظيفة، ومعدلات النجاح، وعدد المهام، وتتبع الأخطاء، انظر المراقبة والمقاييس.

API حالة التزويد

للتحقق من حالة وظيفة التزويد:

```
GET /crm/provision/provision_id/<id>
Authorization: Bearer <token>
```

تتضمن الاستجابة:

```
{
  "provision_id": 123,
  "customer_id": 456,
  "customer_name": "John Smith",
  "product_id": 10,
  "provisioning_status": 0,
  "provisioning_play": "play_psim_only",
  "playbook_description": "OmniCore Service Provisioning 2024",
  "task_count": 85,
  "provisioning_result_json": [
    {
      "event_number": 1,
      "event_name": "Get Product information from CRM API",
      "provisioning_status": 1,
      "provisioning_result_json": "{...}"
    },
    {
      "event_number": 2,
      "event_name": "Create account in OCS",
      "provisioning_status": 1,
      "provisioning_result_json": "{...}"
    }
  ]
}
```

قائمة وظائف التزويد

للحصول على ق��ئمة مجزأة بجميع وظائف التزويد:

```
GET /crm/provision/?
page=1&per_page=20&sort=provision_id&order=desc
Authorization: Bearer <token>
```

يدعم التصفية:

```
GET /crm/provision/?filters={"provisioning_status":
[2]}&search=Mobile
Authorization: Bearer <token>
```

يعيد هذا فقط الوظائف الفاشلة (الحالة 2) حيث تحتوي الوصف على "Mobile".

أفضل الممارسات

تصميم دفتر التشغيل

- **استخدم دائمًا block/rescue**: تأكد من إمكانية التراجع عن التزويد الجزئي
- **استخدم ignore_errors بحذر**: فقط للعمليات الاختيارية حقًا
- **قم بتسجيل المتغيرات المهمة**: استخدم مهام `debug` لتسجيل القيم الرئيسية لاستكشاف الأخطاء وإصلاحها
- **تحقق من الاستجابات**: استخدم `assert` للتحقق من أن استجابات API كما هو متوقع
- **التماثل**: صمم دفاتر التشغيل لتكون قابلة للتشغيل بأمان مرة أخرى

المصادقة

- **التزويد الذي يبدأه المستخدم**: استخدم دائمًا `refresh_token` لـ دفاتر التشغيل الطويلة
- **التزويد الآلي**: استخدم قائمة بيضاء IP أو مفاتيح API مولدة مع رموز وصول
- **انتهاء صلاحية الرمز**: يضمن `refresh_token` تجديد رموز الوصول حسب الحاجة

معالجة الأخطاء

- **قدم السياق**: قم بتضمين `customer_id`، `service_id`، وتفاصيل العملية في رسائل الخطأ
- **أبلغ بشكل مناسب**: تؤدي الأخطاء الحرجة إلى إشعارات عبر البريد الإلكتروني، ولكن لا تزعج للأخطاء المتوقعة
- **معلومات تصحيح الأخطاء**: التقط أجسام الطلب/الاستجابة الكاملة في سجلات `Provision_Event`

الأمان

- **تحقق من المدخلات**: تحقق من `customer_id`، `product_id`، `service_id` قبل التزويد

- **التحقق من الأذونات**: تحقق من أن المستخدمين يمكنهم فقط التزويد للعملاء المصرح لهم
- **البيانات الحساسة**: استخدم نظام الحذف لإزالة كلمات المرور/المفاتيح من السجلات
- **قائمة بيضاء IP**: قيد نقاط نهاية `simple_provision` فقط على الأنظمة الموثوقة

الأداء

- **التنفيذ في الخلفية**: لا تقم بحظر استجابات API في انتظار التزويد
- **فترات الاستعلام**: يجب على واجهة المستخدم الاستعلام عن تحديثات الحالة كل 2-5 ثوانٍ
- **المهام المتوازية**: استخدم التوازي الأصلي لـ Ansible للعمليات المستقلة
- **تحديثات قاعدة البيانات**: يقوم معالج الحدث بتحديث قاعدة البيانات في الوقت الحقيقي، لا حاجة للاستعلام أثناء التنفيذ

الوثائق ذات الصلة

- `concepts_ansible` - مفاهيم التزويد العامة لـ Ansible
- `concepts_sim_provisioning` - تزويد بطاقة SIM (و eSIMالفعلية)
- `concepts_api` - مصادقة واستخدام API CRM
- `concepts_products_and_services` - تعريفات المنتجات والخدمات
- `administration_inventory` - إدارة المخزون للتزويد
- المراقبة والمقاييس - مقاييس أداء التزويد والمراقبة

توفير بطاقة SIM

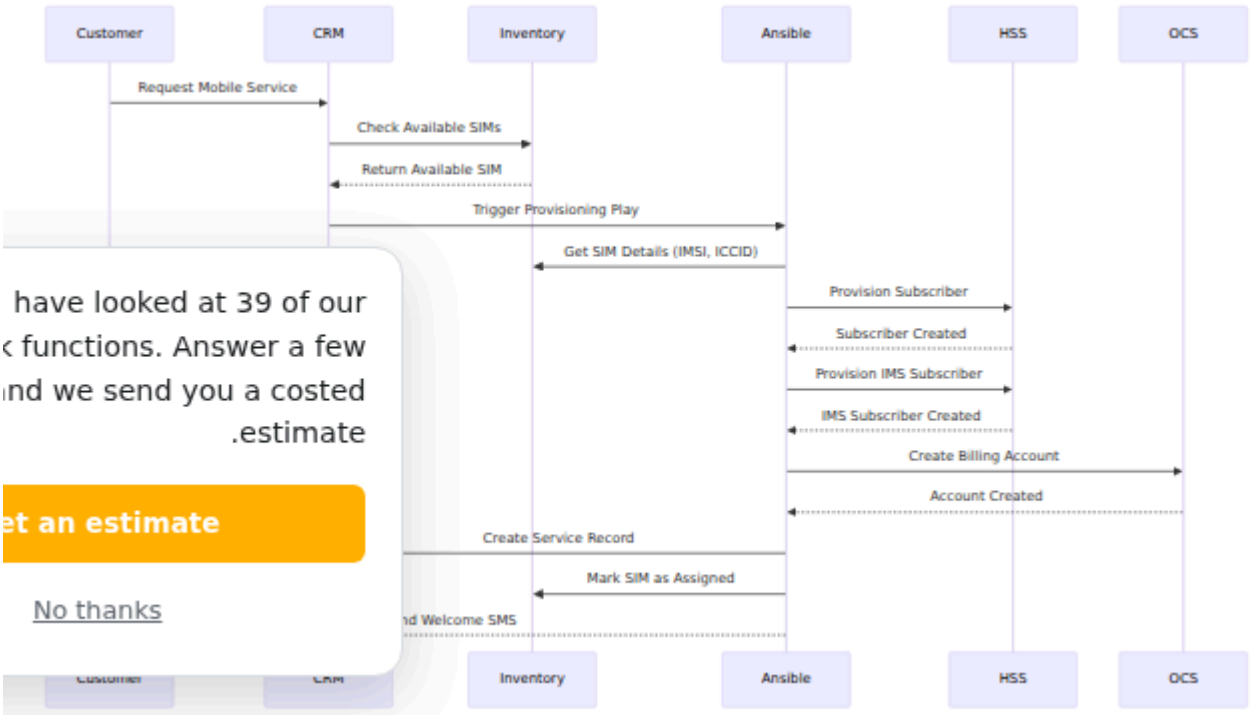
يوفر OmniCRM **بطاقات** SIM دعمًا شاملاً لتوفير كل من **SIM** و **eSIMs الفعلية** (بطاقات SIM المدمجة) و MVNOs. يتعامل النظام مع دورة الحياة الكاملة من إدارة المخزون إلى التفعيل، والتخصيص، وإلغاء التخصيص.

انظر أيضًا: نظام التوفير لمفاهيم التوفير العامة، المخزون لإدارة المخزون، دفاتر تشغيل Ansible لأتمتة التوفير.

نظرة عامة

يتضمن توفير بطاقة SIM في OmniCRM عدة أنظمة متكاملة تعمل معًا:

1. **إدارة المخزون** - تتبع بطاقات SIM المتاحة (بطاقات eSIM الفعلية وملفات)
2. **تكامل HSS/IMS** - توفير بيانات اعتماد المشترك وخدمات الصوت
3. **تكامل OCS** - إعداد الفوترة والرسوم للخدمة
4. **أتمتة Ansible** - تنسيق سير عمل التوفير
5. **دعم التسجيل الذاتي** - يمكّن العملاء من تفعيل بطاقاتهم SIM الخاصة



توفير بطاقة SIM الفعلية

تعد بطاقات SIM الفعلية بطاقات SIM قابلة للإزالة تقليديًا يقوم العملاء بإدخالها في أجهزتهم. يدير HSS/IMS الشبكة للوصول إلى هذه من خلال نظام المخزون ويوفرها إلى OmniCRM.

سير عمل بطاقة SIM الفعلية

1. إعداد المخزون

يجب أولاً تحميل بطاقات SIM في نظام المخزون:

- **ICCID** (معرف بطاقة الدائرة المتكاملة) - معرف بطاقة SIM فريد
- **IMSI** (معرف المشترك المحمول الدولي) - هوية المشترك للشبكة

يتم تخزين هذه في حقول عنصر المخزون:

- `itemtext1`: ICCID
- `itemtext2`: IMSI
- `itemtext3`: نوع SIM (فعلي/eSIM) - اختياري

مثال على عنصر مخزون بطاقة SIM الفعلية:

```
{
  "inventory_id": 1001,
  "item": "SIM Card",
  "itemtext1": "89612345678901234567",  // ICCID
  "itemtext2": "310120123456789",       // IMSI
  "itemtext3": "Physical",               // SIM Type
  "item_location": "Warehouse A, Shelf 3",
  "item_state": "New",
  "wholesale_cost": 2.50,
  "retail_cost": 10.00
}
```

للحصول على شرح كامل لجميع حقول المخزون (itemtext1-20، قيم item_state، حقول العنوان، management_url، إلخ)، انظر نظرة عامة على المخزون - حقول عنصر المخزون.

تخزين بيانات الاعتماد المصادقة

تُخزن بيانات الاعتماد المصادقة لبطاقات SIM في **HSS AuC (مركز المصادقة)**، في ولي�� في مخزون CRM:

- **Ki** (مفتاح المصادقة) - مفتاح سري للمصادقة (مخزن في HSS)
- **OPC** (رمز المشغل) - معلمة مصادقة خاصة بالمشغل (مخزنة في HSS)
- **PIN1/PIN2** - رموز PIN للمستخدم (مخزنة في HSS)
- **PUK1/PUK2** - رموز فتح PIN (مخزنة في HSS)

يتتبع مخزون CRM أي بطاقة SIM (حسب ICCID/IMSI) بينما يتعامل HSS مخصصة لأي عميل، التي تتطابق مع بطاقة SIM المصادقة الفعلية على الشبكة باستخدام بيانات اعتماد Ki/OPC الفعلية.

2. تخصيص الخدمة

عندما يطلب عميل خدمة موبايل:

1. يقوم الموظف أو العميل باختيار بطاقة SIM متاحة من المخزون
2. يتم اختيار رقم موبايل (MSISDN) من مخزون أرقام الهواتف
3. يتم تشغيل دفتر تشغيل التوفير للمنتج (مثل `play_psim_only.yaml`)

3. توفير HSS

يقوم دفتر تشغيل Ansible بتوفير المشترك إلى خادم المشترك المنزلي (HSS):

أولاً، يسترجع **auc_id** (مرجع لبيانات الاعتماد المصادقة المخزنة بالفعل في HSS):

```
- name: Get AuC ID for IMSI
  uri:
    url: "{{ item }}/auc/imsi/{{ imsi }}"
    method: GET
  loop: "{{ hss_peers }}"
  register: auc_lookup

- name: Extract auc_id
  set_fact:
    auc_id: "{{ auc_lookup.results[0].json.auc_id }}"
```

ثم ينشئ سجل المشترك الذي يشير إلى ذلك auc_id:

```
- name: Provision subscriber on HSS
  uri:
    url: "{{ item }}/subscriber/"
    method: PUT
    body_format: json
    body:
      enabled: true
      roaming_enabled: true
      auc_id: "{{ auc_id }}"
      msisdn: "{{ phone_number }}"
      imsi: "{{ imsi }}"
      ue_ambr_dl: 9999999
      ue_ambr_ul: 9999999
      apn_list: "1,2,3"
      default_apn: 1
  loop: "{{ hss_peers }}"
```

تظل بيانات الاعتماد المصادقة (Ki، OPC) في HSS AuC ولا تُعرض أبدًا على CRM أو دفاتر تشغيل التوفير.

4. توفير IMS (لخدمات الصوت)

لتمكين القدرة على إجراء المكالمات الصوتية، يتم إنشاء مشترك IMS:

```
- name: Create IMS subscriber for voice services
  uri:
    url: "{{ item }}/ims_subscriber/"
    method: PUT
    body_format: json
    body:
      imsi: "{{ imsi }}"
      msisdn: "{{ phone_number }}"
      msisdn_list: "{{ phone_number }}"
      ifc_path: "default_ifc.xml"
      sh_profile: "{{ sh_profile_xml }}"
  loop: "{{ hss_peers }}"
```

5. إعداد نظام الفوترة

يتم تكوين OCS (نظام الفوترة عبر الإنترنت) مع:

- إنشاء ح��اب
- تعيين IMSI/MSISDN عبر ملفات تعريف السمات
- قواعد التصفية لتحديد المشترك
- حدود الموارد (الجلسات المتزامنة)
- الرصيد الأولي

6. إنشاء الخدمة وتخصيص المخزون

أخيرًا:

- يتم إنشاء سجل خدمة في CRM
- يتم تخصيص عنصر مخزون بطاقة SIM للعميل (تم تعيينه `customer_id`)
- يتم ربط بطاقة SIM بالخدمة (تم تعيينه `service_id`)
- يتم تحديث حالة المخزون إلى `"Assigned"`

مصادقة بطاقة SIM الفعلية

تستخدم بطاقات SIM الفعلية بيانات اعتماد AuC (مركز المصادقة) المخزنة في HSS:

- تُستخدم 4G/5G للمصادقة على **OPC** و **Ki** (خوارزمية MILENAGE)
- يتم تحميل هذه البيانات مسبقًا في قاعدة بيانات HSS AuC أثناء استيراد SIM
- تحتوي بطاقة SIM الفعلية على قيم Ki/OPC المطابقة
- أثناء التوفير، يشير دفتر التشغيل إلى **auc_id** لربط المشترك بهذه البيانات
- تحدث المصادقة على الشبكة عبر تحدٍ واستجابة بين HSS و SIM
- لا ترى CRM الفعلية أو تتعامل مع الق��م Ki/OPC

توفير eSIM

قائمة على البرمجيات يمكن تنزيلها على SIM هي ملفات تعريف (المدمجة SIM بطاقات) eSIMs. يدعم OmniCRM توفير eSIM مع رموز تفعيل LPA فعلية. يدعم SIM الأجهزة المتوافقة دون بطاقات (مساعد ملف التعريف المحلي).

ميزات eSIM

رموز تفعيل LPA

تستخدم eSIMs رموز LPA للتفعيل:

```
LPA:1$smdp.example.com$ACTIVATION-CODE-ABC123XYZ
```

حيث:

- `LPA:1` - معرف إصدار LPA
- `smdp.example.com` - عنوان خادم SM-DP+ (إعداد بيانات الاشتراك)
- `ACTIVATION-CODE-ABC123XYZ` - رمز تفعيل فريد لهذا الملف الشخصي eSIM

تولید رمز QR

ينشئ OmniCRM تلقائيًا رموز QR من رموز تفعيل LPA:

- مخزنة في حقل `management_url` للمخزون
- يمكن مسحه ضوئيًا QR معروضة في واجهة المستخدم كرمز
- يقوم العملاء بمسحها باستخدام كاميرا أجهزتهم لتثبيت ملف تعريف eSIM
- لا حاجة لكتابة رموز التفعيل الطويلة يدويًا

مثال على عنصر مخزون eSIM:

```
{
  "inventory_id": 1002,
  "item": "eSIM",
  "itemtext1": "89612345678901234567",
  "itemtext2": "310120123456790",
  "itemtext3": "eSIM",
  "management_url": "LPA:1$smdp.example.com$ACTIVATION-CODE-
ABC123XYZ",
  "item_location": "Virtual Inventory",
  "item_state": "New",
  "wholesale_cost": 0.00,
  "retail_cost": 0.00
}
```

تلقائيًا وعرضه لتسهيل المسح QR عند عرض هذا العنصر في واجهة المستخدم، يتم إنشاء رمز.

سير عمل توفير eSIM

سير عمل توفير eSIM مشابه تقريبًا لسير عمل توفير بطاقة SIM الفعلية، مع بعض الاختلافات الرئيسية:

1. تخصيص eSIM تلقائي

إذا كان المنتج يتطلب eSIM فعلية، يمكن للنظام تخصيص SIM ولكن لم يتم اختيار بطاقة SIM متاحة تلقائيًا:

```
# البحث التلقائي عن eSIM
search_filters = {
    "customer_id": [None],
    "item_state": ["New"],
    "itemtext3": ["eSIM"]  # تصفية لنوع eSIM
}
available_esim = search_inventory("SIM Card",
filters=search_filters)
```

2. توفير HSS/IMS

يتم توفير eSIMs إلى HSS/IMS تمامًا مثل بطاقات SIM الفعلية:

- نفس توفير IMSI
- نفس إنشاء مشترك IMS
- نفس بيانات الاعتماد المصادقة (Ki، OPC)

3. تفعيل العميل

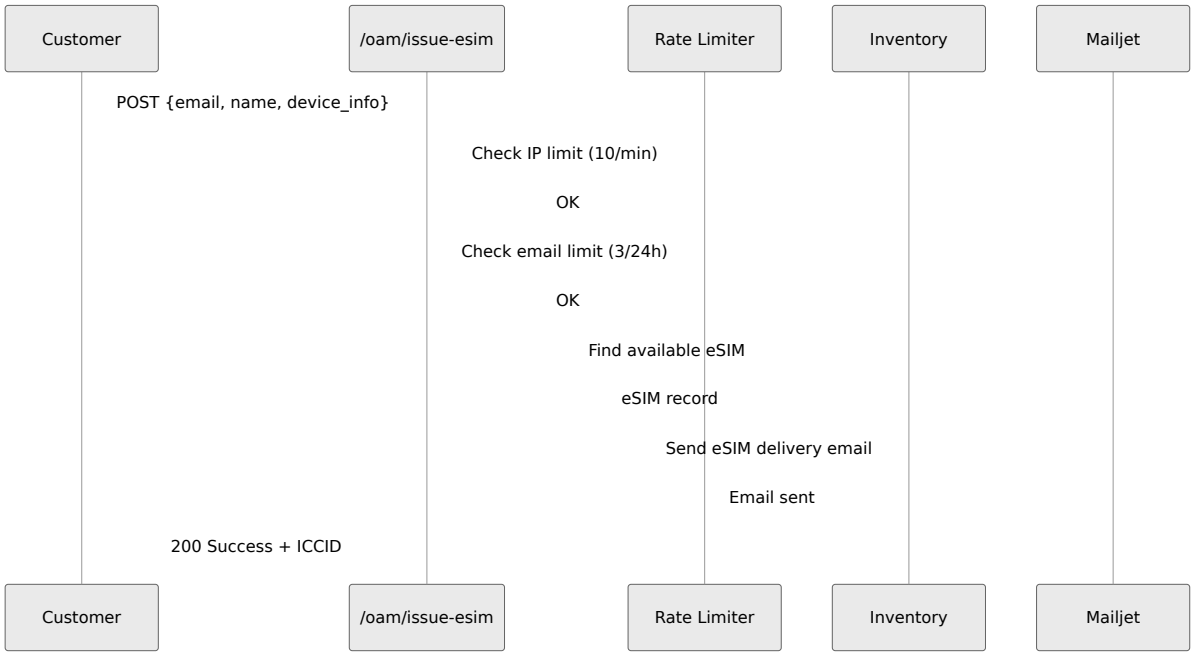
بعد التوفير:

1. يتلقى العميل تفاصيل تفعيل eSIM عبر البريد الإلكتروني أو في بوابة العميل
2. يتم عرض رمز QR للمسح
3. يقوم العميل بمسح رمز QR باستخدام جهازه
4. يقوم الجهاز بتنزيل ملف تعريف eSIM من خادم SM-DP+
5. يتم تثبيت eSIM وهي جاهزة للاستخدام

واجهة برمجة التطبيقات العامة لمصدر eSIM

يوفر OmniCRM نقطة نهاية API عامة لطلبات eSIM من طلب eSIM الذاتية، مما يمكّن العملاء مباشرة من موقع ويب أو تطبيق موبايل دون تدخل الموظفين.

نقطة النهاية: `POST /crm/oam/issue-esim`



معلمات الطلب

المعلمة	النوع	مطلوب	الافتراضي	ف
`email`	سلسلة	نعم	-	البريد تروني لعميل تسليم eSIM يكون ق بريد تروني صالح.
`name`	سلسلة	لا	مستمدة من البريد الإلكتروني	لعميل صيص البريد ي. إذا لم يتم ه، يتم تخدام محلي عنوان البريد تروني
`customer_id`	عدد صحيح	لا	`null`	لعميل ي. عند ينشئ نشاط مرتبط للتتبع
`device_info`	كائن	لا	`{}`	لومات الجهاز حليلات توافق

ف	الافتراضي	مطلوب	النوع	المعلمة
توافق eSIM `supp` `may-` `suppo` `unkn` ستخدم حليلات	`"unknown"`	لا	سلسلة	`device_info.compatibility_status`
وكيل تخدم. نقاطها ئيًا من رؤوس لطلب لم يتم قديمها	رأس الطلب	لا	سلسلة	`device_info.user_agent`

تحديد المعدل

تقوم نقطة النهاية بتنفيذ تحديد مزدوج للمعدل لمنع الإساءة:

الوصف	النطاق	النافذة	الحد
حد معدل عام يتم تطبيقه عبر الزخرفة	لكل عنوان IP	1 دقيقة	10 طلبات
يمنع الطلبات المفرطة لـ eSIM لنفس البريد الإلكتروني	لكل عنوان بريد إلكتروني	24 ساعة	3 طلبات

عند تجاوز حدود المعدل، تعيد نقطة النهاية HTTP 429 مع `error_type: "RateLimitExceeded"`.

تنسيق الاستجابة

استجابة النجاح (HTTP 200):

```
{
  "result": "Success",
  "message": "eSIM request received. You will receive your eSIM QR code via email shortly.",
  "esim_id": 8472,
  "iccid": "89442000000001234567",
  "email_sent": true
}
```

الحقل	النوع	الوصف
`result`	سلسلة	دائمًا `"Success"` لاستجابات HTTP 200
`message`	سلسلة	رسالة تأكيد مقروءة للبشر
`esim_id`	عدد صحيح	معرف المخزون لـ eSIM المعين
`iccid`	سلسلة	ICCID ـ eSIM المعين
`email_sent`	منطقي	ما إذا كانت رسالة البريد الإلكتروني للتسليم قد أُرسلت بنجاح

استجابات الخطأ

رمز HTTP	نوع الخطأ	الوصف
400	-	عنوان بريد إلكتروني غير صالح أو مفقود
429	`RateLimitExceeded`	تجاوز حد المعدل لكل بريد إلكتروني أو لكل عنوان IP
503	`NoInventory`	لا توجد eSIMs متاحة في المخزون
500	-	خطأ في الخادم الداخلي

تنسيق استجابة الخطأ:

```
{
  "result": "Failed",
  "reason": "Error description",
  "error_type": "ErrorType"
}
```

تسليم البريد الإلكتروني

ترسل نقطة النهاية بريدًا إلكترونيًا لتسليم eSIM عبر Mailjet يحتوي على:

- ICCID
- رمز QR (إذا كان متاحًا في `itemtext2`)
- رمز التفعيل/سلسلة LPA (إذا كانت متاحة في `itemtext3` أو `management_url`)
- تعليمات تفعيل خطوة بخطوة

قم بتكوين قالب Mailjet في `crm_config.yaml`:

```
mailjet:
  api_crmCommunicationEsimDelivery:
    from_email: "support@aimobile.ac"
    from_name: "Example Mobile"
    template_id: 1234567
    subject: "Your eSIM is Ready"
```

المعلمة	النوع	مطلوب	الوصف
`from_email`	سلسلة	نعم	عنوان البريد الإلكتروني للمرسل
`from_name`	سلسلة	نعم	اسم العرض للمرسل
`template_id`	عدد صحيح	لا	معرف قالب Mailjet. إذا لم يتم تقديمه، يتم إنشاء بريد إلكتروني HTML افتراضي.
`subject`	سلسلة	نعم	سطر موضوع البريد الإلكتروني

إذا لم يتم تكوين أي قالب، يقوم النظام بإنشاء بريد إلكتروني HTML مع علامة تجارية يحمل:

- علامة تجارية وشعار الشركة
- رمز QR لتفعيل eSIM (إذا كان متاحًا)
- تعليمات رموز التفعيل اليدوية
- إرشادات إعداد محددة للجهاز لنظامي Android و iOS

تسجيل النشاط

يخلق كل طلب eSIM مع سجل نشاط:

- نوع النشاط: `esim_request`
- ارتباط معرف العميل (إذا تم تقديمه)
- حالة توافق الجهاز
- معلومات وكيل المستخدم

هذا يمكّن تتبع طلبات eSIM وتحليلات التحويل.

4. هيكل بيانات eSIM في HSS

تُميز eSIMs في HSS AuC (مركز المصادقة) بعلامة `esim: true`:

```
{
  "esim": true,
  "lpa": "LPA:1$smdp.example.com$ACTIVATION-CODE-ABC123XYZ",
  "iccid": "89612345678901234 57",
  "imsi": "310120123456790",
  "ki": "00112233445566778899AABBCCDDEEFF",
  "opc": "FFEEDDCCBBAA99887766554433221100",
  "pin1": "1234",
  "pin2": "5678",
  "puk1": "12345678",
  "puk2": "87654321",
  "batch_name": "eSIM_Batch_2024_Q1",
  "sim_vendor": "Thales"
}
```

عملية استيراد eSIM

عادةً ما يتم استيراد eSIMs بكميات كبيرة من مزود eSIM:

برنامج الاستيراد: `/OmniCRM-API/Provisioners/customer_product_creation/import_keys_esim.py`

تتضمن عملية الاستيراد:

1. قراءة بيانات eSIM من ملفات Excel المقدمة من البائع
2. تحميل رموز تفعيل LPA من ملفات CSV
3. تحديث HSS ببيانات اعتماد eSIM
4. إنشاء عناصر المخزون المقابلة في CRM
5. ربط ملفات تعريف eSIM بنظام المخزون

يضمن ذلك توفر ملفات تعريف eSIM بشكل صحيح تتبع المخزون على الحفاظ مع للتخصيص.

تكامل المخزون

يعتبر نظام المخزون مركزيًا لتوفير SIM، حيث يتتبع كل من الموارد الفعلية والافتراضية اللازمة لخدمات الهاتف المحمول.

قالب مخزون بطاقة SIM

يحدد قالب مخزون بطاقة SIM النموذجي:

```
{
  "item": "SIM Card",
  "itemtext1_label": "ICCID",
  "itemtext2_label": "IMSI",
  "itemtext3_label": "SIM Type",
  "wholesale_cost": 2.50,
  "retail_cost": 10.00
}
```

ملاحظة: تُخزن بيانات اعتماد المصادقة (Ki، OPC، PIN، PUK) في HSS AuC، وليس في مخزون CRM.

للحصول على معلومات مفصلة حول إنشاء وإدارة قوالب المخزون، بما في ذلك كيفية تعريف الحقول المخصصة (itemtext1-20) وربط القوالب بالمنتجات، انظر إدارة المخزون - قوالب القسم المخزون.

مخزون أرقام الهواتف المحمولة

تتم إدارة أرقام الهواتف كعناصر مخزون منفصلة:

```
{
  "inventory_id": 4001,
  "item": "Mobile Number",
  "itemtext1": "+61412345678",
  "itemtext2": "Melbourne",
  "itemtext3": "Mobile",
  "item_location": "Australia - VIC",
  "item_state": "New",
  "wholesale_cost": 1.00,
  "retail_cost": 0.00
}
```

حالات المخزون لبطاقات SIM

تتقدم بطاقات SIM عبر عدة حالات مخزون:

- **جديد** - SIM غير مستخدم، متاح للتخصيص
- **مخصص** - نشط حاليًا مع عميل
- **مستخدم** - تم تخصيصه سابقًا، تم إرجاعه إلى المخزون (يمكن إعادة استخدامه)
- **استخدام داخلي** - للاختبار أو استخدام الموظفين
- **تالف** - غير وظيفي، يحتاج إلى استبدال
- **مفقود** - لا يمكن تحديد موقعه
- **مسروق** - تم الإبلاغ عنه كمسروق

تكامل المنتج

تحدد المنتجات المخزون المطلوب عبر `inventory_items_list`:

```
{
  "product_id": 1,
  "product_slug": "Mobile-SIM",
  "product_name": "Mobile SIM Only",
  "provisioning_play": "play_psim_only",
  "inventory_items_list": "['SIM Card', 'Mobile Number']"
}
```

عند توفير هذا المنتج:

1. تعرض واجهة المستخدم مختار المخزون مع قائمتين منسدلتين
2. يختار المستخدم بطاقة SIM متاحة (أو eSIM فعلية)
3. يختار المستخدم رقم موبايل متاح
4. يتم تمرير معرفات المخزون إلى دفتر تشغيل Ansible
5. يسترجع دفتر التشغيل تفاصيل المخزون الكاملة عبر API
6. يتقدم التوفير بالموارد المحددة

للحصول على شرح كامل حول كيفية دفع المنتجات للتوفير، وكيفية تعريف متطلبات المخزون، وكيفية تمرير المتغيرات إلى دفاتر التشغيل، انظر نظام التوفير - كيف تدفع المنتجات للتوفير.

تدفق تخصيص المخزون

يتعامل دفتر تشغيل Ansible مع تخصيص المخزون.

للحصول على مزيد من التفاصيل حول كيفية مصادقة دفاتر التشغيل مع واجهة برمجة تطبيقات CRM (رموز Bearer، مفاتيح API، رموز التحديث)، انظر نظام التوفير - المصادقة والتفويض.

```
- name: Get SIM Card details from inventory
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_sim_card }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: sim_inventory_response

- name: Extract SIM identifiers
  set_fact:
    iccid: "{{ sim_inventory_response.json.itemtext1 }}"
    imsi: "{{ sim_inventory_response.json.itemtext2 }}"

- name: Get AuC ID from HSS (reference to authentication
credentials)
  uri:
    url: "{{ item }}/auc/imsi/{{ imsi }}"
    method: GET
  loop: "{{ hss_peers }}"
  register: auc_response

- name: Extract AuC ID
  set_fact:
    auc_id: "{{ auc_response.results[0].json.auc_id }}"

# ... provision to HSS/IMS/OCS ...

- name: Assign SIM to customer
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_sim_card }}"
    method: PATCH
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      customer_id: "{{ customer_id }}"
      service_id: "{{ service_id }}"
      item_state: "Assigned"
```

هذا يضمن:

- يتم وضع علامة على SIM كمخصصة للعميل
- يتم ربط معرف الخدمة للتتبع
- تعكس حالة المخزون الوضع الحالي
- يمكن للأنظمة الأخرى الاستعلام عن المخزون للعثور على SIM الخاص بالعميل

SIM التسجيل الذاتي مع توفير

يدعم OmniCRM تسجيل العملاء الذاتي حيث يمكن للعملاء تفعيل بطاقات SIM أو eSIM الخاصة بهم دون تدخل الموظفين.

تدفق التسجيل الذاتي

1. يبدأ العميل التسجيل

يزور العميل صفحة التسجيل الذاتي ويبدأ إنشاء الحساب:

- يقدم معلومات شخصية
- يختار خطة خدمة (منتج)
- يدخل تفاصيل الدفع

2. التحقق من رقم SIM

يمكن للعميل تقديم رقم SIM (اختياري) الخاص به:

```
GET /auth/self-signup/validate-itemtext1/{sim_number}
```

تتحقق هذه نقطة النهاية مما يلي:

- وجود SIM في المخزون
- أن SIM (غير مخصصة بالفعل) متاحة
- أن SIM ("جديد" أو "مستخدم") في الحالة الصحيحة

3. تخصيص SIM تلقائي (إذا لم يتم تقديم SIM)

إذا لم يكن لدى العميل SIM، يقوم النظام بتخصيص واحدة تلقائيًا:

```
def _populate_base_plan_inventory(sim_number, plan):
    if sim_number:
        # تحقق من وجوده - قدم العميل SIM
        sim_inventory = get_inventory_by_itemtext1(session,
sim_number, None)
        if not sim_inventory:
            # تحقق مما إذا كان هناك خدمة غير نشطة بهذا الرقم
            service = get_inactive_service_by_sim(sim_number)
            if service:
                sim_inventory =
get_sim_for_service(service.service_id)
    else:
        # تخصيص SIM متاحة من المخزون تلقائيًا
        sim_inventory =
get_first_available_inventory_by_item_name(
            session,
            "SIM Card",
            filters={"item_state": ["New"], "customer_id": [None]}
        )

    # الحصول على ICCID و MSISDN
    iccid = sim_inventory.itemtext1

    # تخصيص رقم الهاتف تلقائيًا
    number_inventory = get_first_available_inventory_by_item_name(
        session,
        "Mobile Number"
    )
    msisdn = number_inventory.itemtext1

    return {
        "SIM Card": sim_inventory.inventory_id,
        "Mobile Number": number_inventory.inventory_id
    }
```

4. إنشاء الحساب والتوفير

بمجرد الموافقة على الدفع:

```
def _post_signup_tasks(customer_id, plan, payment_method):
    # 1. حفظ وسيلة الدفع
    save_stripe_payment_method(customer_id, payment_method)

    # 2. شحن بطاقة العميل
    charge_customer_card(customer_id, plan.retail_setup_cost)

    # 3. تسجيل دخول المستخدم (الحصول على رموز JWT)
    tokens = login_customer(customer_id)

    # 4. ملء المخزون (SIM ورقم الهاتف)
    inventory_vars = _populate_base_plan_inventory(sim_number,
plan)

    # 5. إنشاء وظيفة التوفير
    provision_vars = {
        "product_id": plan.product_id,
        "customer_id": customer_id,
        **inventory_vars  # تضمين معرفات بطاقة SIM ورقم الهاتف
    }

    # اختياري: ربط المنتجات الإضافية
    if plan.addon_chain:
        provision_vars["addon_chain"] = plan.addon_chain

    provision_id = create_provisioning_job(provision_vars)

    # 6. إرجاع النجاح مع الرموز
    return {
        "access_token": tokens.access_token,
        "refresh_token": tokens.refresh_token,
        "customer_id": customer_id,
        "provision_id": provision_id
    }
```

5. يتلقى العميل تفاصيل التفعيل

بعد اكتمال التوفير:

- **الفعلية SIM**: بريد ترحيبي مع رقم الهاتف وتعليمات التفعيل
- **eSIM**: يتضمن البريد الإلكتروني رمز QR لتنزيل ملف تعريف eSIM، أو يمكن للعميل عرض رمز QR في بوابة العميل

6. أذونات خاصة للتسجيل الذاتي

مطالبة خاصة JWT أثناء التسجيل، يتضمن رمز:

```
{
  "sub": "customer_123",
  "signup_process": true,
  "permissions": ["VIEW_OWN_PROVISION", "UPDATE_OWN_INVENTORY"]
}
```

هذا يسمح بـ:

- يمكن للعميل قراءة/تحديث المخزون أثناء التسجيل
- مقيد لحساب العميل الخاص بهم فقط
- التي تخص عملاء آخرين SIMs يمنع تخصيص

أمان التسجيل الذاتي

فحوصات التحقق:

- في المخزون SIM يجب أن توجد
- مخصصة لعميل آخر SIM يجب ألا تكون
- في حالة مقبولة ("جديد"، "مستخدم") SIM يجب أن تكون
- لحسابه الخاص فقط SIMs يمكن للعميل تخصيص
- يجب أن يكون الدفع مصرحًا به قبل التوفير

حماية المخزون:

- تضمن المعاملات في قاعدة البيانات تخصيصًا ذريًا
- يتم منع محاولات التسجيل المتزامنة عبر أقفال قاعدة البيانات
- يتم تسجيل تغييرات حالة المخزون للتدقيق

توفير قائم على Ansible

يتم تنسيق جميع توفير SIM من خلال دفاتر تشغيل Ansible، مما يوفر عملية متسقة وقابلة للتدقيق وقابلة للتكرار.

ونمط الكتلة/الإنقاذ، وكيفية تفاعل Ansible للحصول على وثائق شاملة حول هيكل دفتر تشغيل دفاتر التشغيل مع المنتجات، وأفضل الممارسات، انظر دليل مفصل Ansible: دفاتر تشغيل.

دفتر تشغيل توفير SIM الرئيسي

الملف: `/OmniCRM-API/Provisioners/plays/play_psim_only.yaml`

هذا هو دفتر التشغيل الرئيسي لتوفير SIM المحمول (أو eSIM سواء الفعلي). يتعامل مع:

- خدمات الصوت والبيانات المحمولة
- خدمات البيانات فقط
- الصوتيين فقط IMS مشتركي

هيكل دفتر التشغيل (1,561 سطر):

للحصول على شرح مفصل لرؤوس دفتر التشغيل، وتوجيهات hosts/gather_facts/become، والهيكل العام، انظر هيكل دفتر التشغيل وتشريحه - Ansible دفاتر تشغيل.

```yaml
- name: OmniCore Service Provisioning 2024
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Main provisioning block
      block:
        # --- PRE-PROVISIONING (Lines 240-396) ---
        - name: Get SIM information from inventory
          # Retrieves IMSI, ICCID, etc.

        - name: Validate IMSI length
          # Must be exactly 15 digits

        - name: Check if IMSI exists in HSS
          # Prevents duplicate provisioning

        - name: Get AuC ID for IMSI
          # Retrieves auc_id reference from HSS

        # --- HSS PROVISIONING (Lines 398-534) ---
        - name: Create/Update HSS Subscriber
          # Provisions to all HSS peers

        - name: Create IMS Subscriber
          # Enables voice calling

        # --- OCS SETUP (Lines 536-837) ---
        - name: Create ENUM entry
          # E.164 number mapping

        - name: Create FilterS rule
          # Account identification

        - name: Create AttributeS profile
          # IMSI/MSISDN mapping

        - name: Create ResourceS profile
          # Concurrent session limits

        - name: Create OCS account
          # Billing account creation
```

```
    # --- SERVICE & INVENTORY (Lines 865-946) ---
    - name: Create service record
      # In CRM database

    - name: Assign SIM to service
      # Update inventory

    - name: Send welcome SMS
      # Customer notification

  rescue:
    # --- CLEANUP/DEPROVISIONING (Lines 975-1562) ---
    - name: Return inventory to pool
      # Reset customer_id, service_id to null

    - name: Delete OCS account
      # Remove billing account

    - name: Remove HSS subscribers
      # Or set to dormant state

    - name: Determine success/failure
      assert:
        that:
          - action == "deprovision"
```

خطوات التوفير الرئيسية

1. التحقق قبل التوفير

```
- name: Get SIM information from CRM inventory
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_sim_card }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: api_response_sim

- name: Extract and validate IMSI
  set_fact:
    imsi: "{{ api_response_sim.json.itemtext2 }}"

- name: Validate IMSI format
  assert:
    that:
      - imsi | length == 15
      - imsi is match('^[0-9]+$')
    fail_msg: "IMSI must be exactly 15 digits"
```

2. استرجاع AuC ID من HSS

```
- name: Get AuC ID for IMSI from HSS
  uri:
    url: "{{ item }}/auc/imsi/{{ imsi }}"
    method: GET
  loop: "{{ crm_config.hss.hss_peers }}"
  register: auc_lookup

- name: Extract auc_id from first HSS response
  set_fact:
    auc_id: "{{ auc_lookup.results[0].json.auc_id }}"
```

3. توفير مشترك HSS

```yaml
- name: Create subscriber in each HSS peer
  uri:
    url: "{{ item }}/subscriber/"
    method: PUT
    body_format: json
    body:
      enabled: true
      roaming_enabled: true
      auc_id: "{{ auc_id }}"
      msisdn: "{{ phone_number }}"
      imsi: "{{ imsi }}"
      ue_ambr_dl: 9999999  # Download speed limit
      ue_ambr_ul: 9999999  # Upload speed limit
      apn_list: "1,2,3"
      default_apn: 1
    status_code: [200, 201]
  loop: "{{ crm_config.hss.hss_peers }}"
  register: hss_responses
```

4. توفير مشترك IMS (صوت)

```
- name: Create IMS subscriber for voice services
  uri:
    url: "{{ item }}/ims_subscriber/"
    method: PUT
    body_format: json
    body:
      imsi: "{{ imsi }}"
      msisdn: "{{ phone_number }}"
      msisdn_list: "{{ phone_number }}"
      ifc_path: "default_ifc.xml"
      sh_profile: |
        <?xml version="1.0"?>
        <IMSSubscription>
          <PrivateID>{{ imsi }}@ims.mnc{{ mnc }}.mcc{{ mcc
}}.3gppnetwork.org</PrivateID>
          <ServiceProfile>
            <PublicIdentity>
              <Identity>sip:{{ phone_number }}@ims.mnc{{ mnc
}}.mcc{{ mcc }}.3gppnetwork.org</Identity>
            </PublicIdentity>
          </ServiceProfile>
        </IMSSubscription>
    status_code: [200, 201]
  loop: "{{ crm_config.hss.hss_peers }}"
```

5. تكوين فوترة OCS

```yaml
# إنشاء فلاتر الحساب (تحديد المشترك بواسطة IMSI/MSISDN)
- name: Create FilterS for IMSI identification
  uri:
    url: "{{ crm_config.ocs.ocsApi }}/jsonrpc"
    method: POST
    body_format: json
    body:
      method: "ApierV1.SetFilter"
      params:
        - ID: "FILTER_IMSI_{{ service_uuid }}"
          Type: "*string"
          Element: "~*req.OriginHost"
          Values: ["{{ imsi }}"]

# إنشاء ملف تعريف السمة (تعيين IMSI الحساب إلى)
- name: Create AttributeS profile
  uri:
    url: "{{ crm_config.ocs.ocsApi }}/jsonrpc"
    method: POST
    body_format: json
    body:
      method: "ApierV1.SetAttributeProfile"
      params:
        - ID: "ATTR_ACCOUNT_{{ service_uuid }}"
          FilterIDs: ["FILTER_IMSI_{{ service_uuid }}"]
          Attributes:
            - Path: "*req.Account"
              Value: "{{ service_uuid }}"

# إنشاء حساب OCS
- name: Create OCS account
  uri:
    url: "{{ crm_config.ocs.ocsApi }}/jsonrpc"
    method: POST
    body_format: json
    body:
      method: "ApierV2.SetAccount"
      params:
        - Tenant: "{{ crm_config.ocs.ocsTenant }}"
          Account: "{{ service_uuid }}"
          ActionPlanIds: []
          ExtraOptions:
            AllowNegative: false
```

```
            Disabled: false

# إضافة رصيد أولي
- name: Add initial monetary balance
  uri:
    url: "{{ crm_config.ocs.ocsApi }}/jsonrpc"
    method: POST
    body_format: json
    body:
      method: "ApierV1.AddBalance"
      params:
        - Tenant: "{{ crm_config.ocs.ocsTenant }}"
          Account: "{{ service_uuid }}"
          BalanceType: "*monetary"
          Balance:
            Value: 0
            ExpiryTime: "+8760h"  # 1 year
```

6. إنشاء الخدمة

```
- name: Create service record in CRM
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/service/"
    method: PUT
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      customer_id: "{{ customer_id }}"
      product_id: "{{ product_id }}"
      service_name: "Mobile Service - {{ phone_number }}"
      service_uuid: "{{ service_uuid }}"
      service_status: "Active"
      retail_cost: "{{ monthly_cost }}"
    status_code: 200
  register: service_creation_response

- name: Extract service_id
  set_fact:
    service_id: "{{ service_creation_response.json.service_id }}"
```

7. تخصيص المخزون

```
- name: Assign SIM card to customer and service
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_sim_card }}"
    method: PATCH
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      customer_id: "{{ customer_id }}"
      service_id: "{{ service_id }}"
      item_state: "Assigned"
      sold_date: "{{ ansible_date_time.iso8601 }}"

- name: Assign phone number to customer and service
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_phone_number }}"
    method: PATCH
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      customer_id: "{{ customer_id }}"
      service_id: "{{ service_id }}"
      item_state: "Assigned"
      sold_date: "{{ ansible_date_time.iso8601 }}"
```

إلغاء التخصيص والعودة

يتعامل نفس دفتر التشغيل مع كل من إلغاء التخصيص الفاشل والتنظيف المتعمد باستخدا كتلة `rescue` الخاصة بـ Ansible.

للحصول على شرح تفصيلي لنمط الكتلة/الإنقاذ ولماذا يعد هذا من أفضل الممارسات، انظر نظام التوفير - العودة والتنظيف: نمط أفضل ممارسة.

```yaml
rescue:
  # يتم تشغيل هذا القسم عندما:
  # 1. يفشل أي مهمة في الكتلة (العودة التلقائية)
  # 2. action == "deprovision" (تنظيف متعمد)

  - name: Get inventory items linked to service
    uri:
      url: "{{ crm_config.crm.base_url
}}/crm/inventory/customer_id/{{ customer_id }}"
      method: GET
      headers:
        Authorization: "Bearer {{ access_token }}"
    register: inventory_items
    ignore_errors: true

  - name: Return inventory to available pool
    uri:
      url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ item.inventory_id }}"
      method: PATCH
      headers:
        Authorization: "Bearer {{ access_token }}"
      body_format: json
      body:
        customer_id: null
        service_id: null
        item_state: "Used"
    loop: "{{ inventory_items.json.data }}"
    when: inventory_items.json.data is defined
    ignore_errors: true

  - name: Delete OCS account
    uri:
      url: "{{ crm_config.ocs.ocsApi }}/jsonrpc"
      method: POST
      body_format: json
      body:
        method: "ApierV1.RemoveAccount"
        params:
          - Tenant: "{{ crm_config.ocs.ocsTenant }}"
            Account: "{{ service_uuid }}"
    ignore_errors: true
```

```
  - name: Delete or disable HSS subscriber
    uri:
      url: "{{ item.key }}/subscriber/{{ item.value.subscriber_id
}}"
      method: DELETE
    loop: "{{ hss_subscriber_data | dict2items }}"
    when: deprovision_subscriber | default(false) | bool
    ignore_errors: true

  # تؤكد النتيجة النهائية النجاح أو الفشل
  - name: Determine if this was intentional deprovision or failed
provision
    assert:
      that:
        - action == "deprovision"
      fail_msg: "Provisioning failed and rollback completed"
      success_msg: "Deprovisioning completed successfully"
```

النقاط الرئيسية:

- `ignore_errors: true` على جميع مهام التنظيف يضمن إجراء جميع محاولات التنظيف
- إذا كانت `action == "deprovision"`، فإن التأكيد يمر (الحالة 0 = النجاح)
- إذا لم يتم تعيين `action`، فإن التأكيد يفشل (الحالة 2 = فشل التوفير)
- يضمن ذلك عمليات ذرية: إما أن يتم ا��توفير بالكامل أو يتم تنظيفه بالكامل

دفاتر تشغيل SIM الأخرى

دفتر تشغيل التطوير المحلي: `play_local_mobile_sim.yaml` - إصدار تطوير يتخطى الاعتماد على HSS/OCS الخارجية

تبديل MSISDN: `play_swap_msisdn.yaml` - تبديل أرقام الهواتف بين الخدمات الحالية

دفاتر تشغيل الشحن:

- `play_topup_monetary.yaml` - إضافة رصيد مالي
- `play_topup_no_charge.yaml` - إضافة بيانات/دقائق مجانية
- `play_topup_dongle.yaml` - شحن بيانات IoT/للأجهزة

تكامل HSS و IMS

يتكامل OmniCRM مع خادم المشترك المنزلي الخارجي (HSS) لإدارة المشتركين والمصادقة.

وظائف HSS

الملف: `/OmniCRM-API/Provisioners/hss.py`

الوظائف الرئيسية:

```
def ProvisionMobileSIM(json_data, hss_urls):
    """
    Provisions a mobile SIM to HSS and IMS

    Args:
        json_data: Dict with IMSI, MSISDN, authentication
credentials
        hss_urls: List of HSS peer URLs

    Returns:
        Dict with subscriber IDs from each HSS peer
    """
    # Create HSS subscriber (for data)
    # Create IMS subscriber (for voice)
    # Return subscriber IDs for tracking

def get_subscriber_info_msisdn(msisdn, hss_urls):
    """
    Retrieves subscriber details by phone number

    Args:
        msisdn: Phone number to lookup
        hss_urls: List of HSS peer URLs

    Returns:
        Subscriber information from HSS
    """
```

دعم HSS المتعدد

يدعم OmniCRM التوفير إلى عدة نظراء HSS من أجل التكرار:

```yaml
# crm_config.yaml
hss:
  hss_peers:
    - "http://10.12.64.140:8080"
    - "http://10.12.64.141:8080"
  apn_list: "1,2,3"
  default_apn: 1
```

يحدث التوفير عبر جميع مثيلات HSS المكونة لضمان:

- التكرار الجغرافي
- توزيع الحمل
- القدرة على الفشل
- بيانات المشترك المتزامنة

مركز المصادقة (AuC)

يخزن AuC بيانات اعتماد المصادقة للمشتركين. أثناء التوفير، يتم استرجاع فقط **auc_id**:

احصل على AuC ID لـ IMSI:

```
GET /auc/imsi/{imsi}
```

الاستجابة:

```
{
  "auc_id": 12345,
  "imsi": "310120123456789",
  "iccid": "89612345678901234 56",
  "esim": false,
  "lpa": null
}
```

تحتوي على رمز التفعيل للعرض على `lpa` و `true` هي `esim` تكون علامة eSIMs، بالنسبة لـ العملاء.

ملاحظة: تظل بيانات الاعتماد الفعلية (Ki، OPC، PIN، PUK) في HSS ولا تُعرض أبدًا عبر API. يحتاج نظام التوفير فقط إلى `auc_id` للإشارة إلى هذه البيانات عند إنشاء المشتركين.

تكامل OCS

يتعامل نظام الفوترة عبر الإنترنت (OCS) مع الشحن والتقييم في الوقت الفعلي لخدمات الهاتف المحمول. يستخدم OmniCRM CGRateS كمنصة OCS.

تكوين OCS

الملف: `/OmniCRM-API/Provisioners/ocs.py`

التكوين:

```
# crm_config.yaml
ocs:
  cgrates: "10.64.12.160:2080"
  ocsTenant: "mnc001.mcc001.3gppnetwork.org"
  ocsApi: "http://10.64.12.160:2080"
```

مكونات OCS لخدمات SIM

1. الفلاتر - تحديد المشترك

تحدد الفلاتر المشترك من الطلبات الشبكية:

```
{
  "ID": "FILTER_IMSI_Service_abc123",
  "Type": "*string",
  "Element": "~*req.OriginHost",
  "Values": ["310120123456789"]
}
```

2. السمات - ربط الحساب

تربط السمات IMSI بحساب الفوترة:

```
{
  "ID": "ATTR_ACCOUNT_Service_abc123",
  "FilterIDs": ["FILTER_IMSI_Service_abc123"],
  "Attributes": [
    {
      "Path": "*req.Account",
      "Value": "Service_abc123"
    },
    {
      "Path": "*req.Bandwidth",
      "Value": "100000000"  // 100 Mbps
    }
  ]
}
```

3. الموارد - حدود الجلسات

تتحكم الموارد في حدود الجلسات المتزامنة:

```
{
  "ID": "RES_SESSIONS_Service_abc123",
  "FilterIDs": ["FILTER_IMSI_Service_abc123"],
  "Limit": 5,  // Max 5 concurrent sessions
  "UsageTTL": "-1"
}
```

4. الإحصائيات - تتبع الاستخدام

تتبع قوائم الإحصائيات الاستخدام للتحليلات:

```
{
  "ID": "STATS_Service_abc123",
  "FilterIDs": ["FILTER_IMSI_Service_abc123"],
  "Metrics": [
    "*sum#~*req.Usage",
    "*tcc"  // Total call count
  ]
}
```

5. خطط العمل - الرسوم المتكررة

تتعامل خطط العمل مع العمليات المتكررة مثل الرسوم الشهرية:

```
{
  "Id": "ActionPlan_Service_abc123_Monthly",
  "ActionsId": "Action_Add_Monthly_Data",
  "Timing": {
    "MonthDays": [1],  // 1st of each month
    "Time": "00:00:00Z"
  }
}
```

وظائف واجهة برمجة التطبيقات OCS

الوظائف الرئيسية من ocs.py:

```
async def async_get_balance(account, server, tenant):
    """Get current account balance"""

async def async_topup_dongle(tenant, account, server, days):
    """Add time-based data allowance"""

def Get_Account_Status(account, server, tenant):
    """Get comprehensive account information"""

def CGRateS_API_Call(method, params, server, tenant):
    """Generic CGRateS JSON-RPC API call"""
```

تسوية مخزون SIM

يتضمن OmniCRM أدوات لضمان التناسق بين مخزون CRM و HSS.

برنامج التسوية

الملف: `/OmniCRM-API/Provisioners/hss_reconcile.py`

الغرض:

- تحديد بطاقات SIM في HSS (يتيمة) ولكن ليست في المخزون
- العثور على بطاقات SIM في المخزون التي لم يتم توفيرها في HSS المخصصة
- التحقق من تنسيقات MSISDN
- إنشاء تقارير HTML

الفحوصات التي يتم إجراؤها:

1. **بطاقات SIM المباعة في HSS:**
    - استعلام عن جميع بطاقات SIM المخصصة من المخزون
    - تحقق مما إذا كان كل IMSI موجودًا في HSS
    - الإبلاغ عن بطاقات SIM المخصصة للعملاء ولكن ليست في HSS
2. **بطاقات SIM الموفرة في المخزون:**
    - استعلام عن جميع المشتركين من HSS
    - تحقق مما إذا كان كل IMSI موجودًا في المخزون
    - الإبلاغ عن إدخالات HSS اليتيمة
3. **التحقق من MSISDN:**
    - التحقق من تنسيقات أرقام الهواتف
    - التحقق من الامتثال لرقم الهاتف الأسترالي
    - الإبلاغ عن الأرقام غير الصالحة

تشغيل التسوية:

```
python hss_reconcile.py
```

الإخراج:

- تقرير HTML مع نتائج ملونة
- قوائم بالمطابقات للتحقيق
- توصيات للتنظيف

أفضل الممارسات:

- تشغيل التسوية أسبوعيًا
- التحقيق في التباينات على الفور
- استخدام التقارير لتحديد فشل التوفير
- تنظيف الإدخالات اليتيمة لتحرير الموارد

استكشاف مشاكل توفير SIM

المشكلات الشائعة والحلول

المشكلة: فشل توفير SIM مع "IMSI موجود بالفعل في HSS"

- **السبب:** تم توفير IMSI بالفعل من محاولة سابقة
- **الحل:** تحقق من HSS لمشترك موجود، إما:
  - حذف المشترك الموجود إذا كان يتيمًا
  - تحديث المشترك الموجود بدلاً من إنشاء جديد
  - تشغيل التسوية لتحديد التعارض

المشكلة: عدم توليد رمز QR لـ eSIM

- **السبب:** لم يتم ملء حقل `management_url` برمز LPA
- **الحل:** تأكد من أن برنامج استيراد eSIM يحدد `management_url` بشكل صحيح
- **التنسيق:** يجب أن يكون `LPA:1$server$activation-code`

المشكلة: يظهر عنصر المخزون كـ "متاح" ولكنه مخصص

- **السبب:** فشل التوفير قبل تخصيص المخزون

- **الحل:** تحقق من أحداث التوفير لمعرفة سبب الفشل
- **الإصلاح:** إما إكمال التوفير أو إعادة المخزون إلى المجموعة

المشكلة: لا يمكن للعميل إجراء مكالمات (البيانات تعمل)

- **السبب:** لم يتم توفير مشترك IMS
- **الحل:** تحقق مما إذا كانت خطوة توفير IMS قد اكتملت
- **الإصلاح:** توفير مشترك IMS يدويًا أو إعادة تشغيل دفتر التشغيل

المشكلة: فشل المصادقة (الاتصال SIM لا يمكن لـ)

- **السبب:** عدم تطابق Ki/OPC بين بطاقة SIM و HSS
- **الحل:** تحقق من أن بيانات اعتماد SIM الفعلية تتطابق مع بطاقة AuC
- **الإصلاح:** تحديث HSS AuC بالبيانات الصحيحة

المشكلة: فشل التسجيل الذاتي في تخصيص SIM

- **السبب:** لا توجد SIMs متاحة في المخزون
- **الحل:** تحقق من المخزون عن SIMs بحالة "جديد" و customer_id NULL
- **الإصلاح:** استيراد دفعة SIM جديدة أو وضع علامات على SIMs المرتجعة كـ "مستخدمة""

نصائح للتصحيح

1. تحقق من أحداث التوفير:

```
GET /crm/provision/provision_id/{id}
```

ابحث عن المهام الفاشلة (الحالة 2) واستعرض رسائل الخطأ.

2. تحقق من مشترك HSS:

```
GET {hss_url}/subscriber/msisdn/{phone_number}
```

تحقق مما إذا كان المشترك موجودًا وله تكوين صحيح.

3. تحقق من حساب OCS:

```
{
  "method": "ApierV2.GetAccount",
  "params": [
    {
      "Tenant": "mnc001.mcc001.3gppnetwork.org",
      "Account": "Service_abc123"
    }
  ]
}
```

4. استعلام عن حالة المخ��ون:

```
GET /crm/inventory/?filters={"itemtext2":["310120123456789"]}
```

ابحث حسب IMSI للعثور على عنصر المخزون والتحقق من حالته.

5. مراجعة سجلات Ansible:

أو تفاصيل أحداث التوفير للحصول على مخرجات دفتر التشغيل `/var/log/ansible/` تحقق من بالكامل.

أفضل الممارسات

إدارة مخزون SIM

1. استخدم برامج الاستيراد لدفعات SIM الكبيرة **الاستيراد بالجملة:**
2. تشغيل التسوية الأسبوعية بين HSS والمخزون **التسوية المنتظمة:**
3. الحفاظ على دقة حالات المخزون (جديد، مخصص، مستخدم) **إدارة الحالة:**
4. سجل معلومات بائع SIM واسم الدفعة **تتبع البائع:**
5. الحفاظ على تكاليف الجملة/التجزئة بدقة **تتبع التكلفة:**

الأمان

1. تظل بيانات اعتماد المصادقة (Ki، OPC) في HSS فقط، ولا توجد في **فصل البيانات:** CRM

2. **التحكم في الوصول:** تحديد من يمكنه تخصيص/تعديل مخزون SIM
3. **تسجيل التدقيق:** تتبع جميع تخصيصات SIM والتغييرات
4. **إدارة PIN/PUK:** تُخزن رموز PIN/PUK في HSS، وتُقدم للعملاء عبر قنوات آمنة
5. **عملية المفقود/المسروق:** تعليق الخدمة على الفور في HSS ووضع علامة على حالة المخزون

التوفير

1. **التحقق:** تحقق دائمًا من تنسيق IMSI وفريد من نوعه
2. **العودة:** استخدم الكتلة/الإنقاذ لتنظيف تلقائي عند الفشل
3. **الإشعارات:** إرسال تعليمات تفعيل واضحة للعملاء
4. **الاختبار:** اختبار التوفير في التطوير قبل الإنتاج
5. **المراقبة:** تتبع معدلات نجاح التوفير وأسباب الفشل

محددة eSIM

1. **تنسيق LPA:** تحقق من تنسيق رمز LPA قبل التخزين
2. **اختبار QR:** اختبار رموز QR على أجهزة متعددة
3. **تعليمات التفعيل:** توفير أدلة خطوة بخطوة واضحة
4. **إدارة الملف الشخصي:** تتبع تنزيلات وتثبيتات ملفات تعريف eSIM
5. **رموز النسخ الاحتياطي:** تقديم رموز QR اليدوية في حالة فشل مسح LPA

الوثائق ذات الصلة

- نظام التوفير - تفاصيل سير العمل الكامل للتوفير
- دفاتر تشغيل Ansible - هيكل دفتر التشغيل وأفضل الممارسات
- إدارة المخزون - نظرة عامة على نظام المخزون
- المنتجات والخدمات - تكوين المنتج
- رعاية العملاء - دعم العملاء في مشاكل SIM
- الشحن وإعادة الشحن - إضافة رصيد إلى خدمات الهاتف المحمول

سجل نشاط العميل

يسجل سجل النشاط كل تغيير تم إجراؤه على عميل أو جهة اتصال أو موقع، أو خدمة، أو معلمات مالية مثل المعاملات، والفواتير، وطرق الدفع.

يسمح لنا ذلك بتتبع التغييرات التي تم إجراؤها على النظام، ومن الذي قام بها، ومتى تم ذلك. السجل مفيد لمراجعة التغييرات وتتبع المشكلات. على سبيل المثال، إذا قال عميل إنه لم يتلقَ فاتورة، يمكننا التحقق من سجل النشاط لمعرفة ما إذا كان النظام قد أرسلها. إذا تمت إزالة جهة اتصال، يمكننا أن نرى من الذي أزالها ومتى.

سجل النشاط هو قائمة زمنية بالتغييرات. تظهر أحدث التغييرات في الأعلى، وتظهر التغييرات القديمة في الأسفل. يمكنك تصفية القائمة من علامات التبويب.

لا يمكنك حذف سجلات سجل النشاط. يمكنك تصفيتها، ويمكنك عرض التفاصيل لمعرفة ما تم تغييره ومن قام بذلك.

إضافة خدمة

الخدمات هي الأشياء التي نقوم بفرض رسوم عليها على العميل. يمكن أن تكون خدمات الإنترنت المنزلية، أو الخدمات المحمولة، أو حتى خدمات مجردة مثل تأجير شبكة فرعية أو توفير الكهرباء المدفوعة لرف.

الخدمة هي مجرد حالة من المنتج لعميل معين. يقوم النظام باختيارها من كتالوج المنتجات وتوفيرها للعميل.

إذا لم تكن قد أنشأت عميلًا بالفعل، تحتاج إلى القيام بذلك أولاً، لأن الخدمات مرتبطة بالعملاء. تحتاج أيضًا إلى تحديد طريقة دفع للعميل، لأن الخدمات تولد رسومًا يجب دفعها.

يمكن للعملاء توفير خدماتهم الخاصة (إذا سمحنا لهم بذلك). يمكن أيضًا لموظفي خدمة العملاء توفير الخدمات للعميل.

تحدد تعريفات المنتجات قواعد حول من يمكنه شراء منتج. على سبيل المثال، يمكن أن تسمح قاعدة معينة فقط للعملاء التجاريين بشراء منتج تجاري. يمكن أن تسمح قاعدة أيضًا فقط للعملاء الذين لديهم خدمة محمولة بشراء إضافة محمولة.

يمكن أن تحتوي الخدمات أيضًا على مكونات استخدام، مثل استخدام البيانات، ودقائق المكالمات، أو رسوم الاستخدام الأخرى. يمكن أن تحتوي الخدمة على رسوم متعددة، مثل الرسوم الشهرية، أو الرسوم لمرة واحدة، أو رسوم الاستخدام. نحن نعرض هذه من زر "الاستخدام".

تدعم العديد من الخدمات الإضافات. على سبيل المثال، تحتوي الخدمة المحمولة على جميع الشحنات المتاحة من قائمة الإضافات. قد تحتوي خدمة الإنترنت المنزلية على عنوان IP ثابت أو بيانات إضافية متاحة كإضافة. يحدد كتالوج المنتجات كل هذا.

سير العمل لتعيين الخطط

تتيح ميزة تعيين الخطط للموظفين **توفير الخدمات للعملاء** من خلال اختيار المنتجات من الكتالوج وبدء عملية التوفير. هذه هي الطريقة الرئيسية لإنشاء خدمات جديدة عندما لا يقوم العملاء بالتوفير الذاتي.

نظرة عامة

يتضمن تعيين خطة ما يلي:

1. اختيار عميل
2. اختيار منتج من الكتالوج
3. تكوين متطلبات المخزون (المعدات، إلخ، بطاقات SIM)
4. تعيين معلمات الخدمة (التجديد التلقائي، الحقول المخصصة)
5. بدء التوفير
6. مراقبة تقدم التوفير

يتم استخدام هذا سير العمل لجميع أنواع الخدمات: الهاتف المحمول، الإنترنت، IPTV، وVoIP.

الوصول إلى تعيين الخطة

من كتالوج المنتجات:

من صفحة العميل:

من صفحة الإضافات:

تفتح واجهة تعيين الخطة في نافذة منبثقة أو صفحة مخصصة.

سير العمل خطوة بخطوة

الخطوة 1: تصفح كتالوج المنتجات

يعرض كتالوج المنتجات المنتجات المتاحة مجمعة حسب الفئة ونوع العميل.

تبديلات العرض:

الفئات: • خطط الخدمة • الإضافات

يؤدي التبديل بين الفلاتر الفردية والتجارية إلى تصفية المنتجات لعرض تلك المتاحة فقط لنوع العميل المحدد.

بطاقات المنتجات:

تُعرض المنتجات في نافذة دوارة أو شبكة:

انقر على **"تعيين للعميل"** للمتابعة.

الخطوة 2: اختيار العميل

إذا لم تكن بالفعل على صفحة العميل، يطلب منك النظام اختيار عميل.

بحث عن العملاء:

> ابحث عن العملاء: [جون ▼]
>
> العملاء المتطابقون: • جون دو (ID: 456) • مؤسسات ون سميث (ID: 123) • جونسون (ID: 789)

اكتب للبحث عن:

- اسم العميل
- ID العميل
- عنوان البريد الإلكتروني
- رقم الهاتف

اختر العميل من القائمة المنسدلة.

> **⚠ WARNING**
> تأكد من أنك قد اخترت العميل الصحيح قبل المتابعة. إذا قمت بتعيين خطة للعميل الخطأ، فسيتطلب الأمر تدخلًا يدويًا لتصحيح ذلك.

الخطوة 3: تكوين المخزون (إذا لزم الأمر)

إذا كان المنتج يتطلب عناصر مخزون (محددة في `inventory_items_list`)، تظهر مختارات المخزون.

مثال مختار المخزون:

> عناصر المخزون المطلوبة:
>
> بطاقة SIM *
>
> بطاقات SIM المتاحة: • SIM-00123 - ICCID: 8944...0001 (جديدة) • SIM-00124 - ICCID: 8944...0002 (جديدة) • SIM-00125 - ICCID: 8944...0003 (جديدة)
>
> رقم الهاتف المحمول *
>
> الأرقام المتاحة: • +44 7700 900123 (محجوزة) • +44 7700 900124 (متاحة) • +44 7700 900125 (متاحة)

قواعد اختيار المخزون:

- يشير النجمة (*) إلى حقل مطلوب
- تظهر فقط عناصر المخزون المتاحة (الحالة: "في المخزون" أو "جديدة")
- يتم تحميل القوائم المنسدلة ديناميكيًا بناءً على أسماء قوالب المخزون
- بمجرد تحديدها، يتم حجز العناصر مؤقتًا

ماذا يحدث:

- يتم تمرير عناصر المخزون المحددة كمتغيرات إلى كتاب التوفير

- أثناء التوفير، يتم تعيين العناصر للخدمة والعميل
- "تتغير حالة العنصر من "في المخزون" إلى "مخصص

الخطوة 4: تكوين التجديد التلقائي (اختياري)

بالنسبة للخدمات المتكررة، قد يُطلب منك تعيين التجديد التلقائي:

> هل ترغب في تفعيل التجديد التلقائي لهذه الخدمة؟
>
> عند التفعيل، ستتجدد هذه الخدمة تلقائيًا في نهاية كل فترة فواتير وتُحاسب طريقة الدفع الافتراضية للعميل.
>
> [نعم] [لا]

سلوك التجديد التلقائي:

- **نعم**: تتجدد الخدمة تلقائيًا، ويتم تحصيل رسوم من العميل شهريًا
- **لا**: تنتهي الخدمة في نهاية فترة العقد، ويتطلب التجديد يدويًا

أفضل الممارسات: افتراضيًا إلى "نعم" للخدمات الاستهلاكية، و"لا" للخدمات لمرة واحدة أو عندما يطلب العميل التحكم اليدوي.

الخطوة 5: مراجعة وتأكيد

تظهر شاشة المراجعة جميع الاختيارات قبل التوفير:

> المنتج: خطة الهاتف المحمول - 20 جيجابايت (ID: 123) العميل: جون سميث
>
> رقم الهاتف المحمول: +44 7700 900123 • المخزون: بطاقة SIM: SIM-00123 (ICCID: 8944...0001)
>
> التسعير: • رسوم الإعداد: £0.00 • التكلفة الشهرية: £15.00
>
> التجديد التلقائي: نعم
>
> [تأكيد وتوفير] [إلغاء]

انقر على **"تأكيد وتوفير"** لبدء عملية التوفير.

الخطوة 6: تقدم التوفير

تظهر نافذة التوفير تقدمًا في الوقت الفعلي:

> ✓ تعيين رقم SIM (ICCID: 8944...0001) ✓ تعيين بطاقة التحقق من حساب العميل ✓ تعيين رقم الهاتف المحمول (+44 7700 900123) ⟳ تكوين OCS (قيد التقدم...) إنشاء ساب سجل الخدمة إرسال بريد الترحيب
>
> التقدم: 3 من 6 مهام مكتملة

تستعلم النافذة عن واجهة برمجة التطبيقات للتوفير كل 0.2 ثانية للحصول على تحديثات الحالة.

مؤشرات التقدم:

- ☑ مكتمل بنجاح
- ⟳ قيد التشغيل حاليًا
- معلق (لم يبدأ)
- ✗ فشل (إذا حدثت أخطاء)

الخطوة 7: الإكمال

نجاح:

> تم توفير الخدمة بنجاح لجون سميث
>
> الخدمة: 789 اسم الخدمة: الهاتف المحمول - +44 7700 900123 الحالة: نشط ID
>
> [إغلاق] [عرض الخدمة]

انقر على **"عرض الخدمة"** لفتح صفحة تفاصيل الخدمة.

فشل:

إذا فشل التوفير:

> خطأ: غير قادر على الاتصال بـ OCS
>
> تم إنشاء سجل الخدمة ولكن لم يكتمل التوفير. يرجى مراجعة الخطأ وإعادة المحاولة.
>
> التوفير: ID 456
>
> [إغلاق] [عادة المحاولة] [عرض السجلات]

- **عرض السجلات**: يفتح تفاصيل التوفير مع رسائل الخطأ
- **إعادة المحاولة**: يحاول التوفير مرة أخرى
- **إغلاق**: يخرج من النافذة (يظل سجل الخدمة ولكن غير مفعل)

حالات خاصة

إضافة إضافات إلى خدمة موجودة

عند تعيين إضافة (فئة: "إضافة") لعميل لديه خدمة بالفعل:

1. **الكشف التلقائي عن الخدمة:**
   - يجد النظام خدمات العميل الحالية
   - يتم تصفيتها حسب نوع الخدمة (الإضافة المحمولة تظهر فقط للخدمات المحمولة)
   - إذا كان لدى العميل خدمات متعددة متطابقة، يُطلب منه اختيار أي منها
2. **لا يتم إنشاء خدمة جديدة:**
   - يتم توفير الإضافة ضد service_id الموجود
   - يستخدم حساب OCS للخدمة الحالية
   - يتم تعيين المخزون (إن وجد) للخدمة الحالية
3. **كتاب التوفير:**
   - كتاب مختلف عن الخدمات المستقلة
   - عادةً ما يضيف رصيدًا، ميزات، أو معدات إلى الحساب الحالي

مثال:

> • خدمات الهاتف المحمول الحالية: • الهاتف المحمول - +44 7700 900123 (ID: 789)
> • الهاتف المحمول - +44 7700 900456 (ID: 790)
>
> أي خدمة يجب أن تتلقى هذه الإضافة؟ [الهاتف المحمول - +44 7700 900123 ▼]
>
> [متابعة] [إلغاء]

التوفير للعملاء التجاريين

قد يكون لدى العملاء التجاريين متطلبات إضافية:

- **اختيار الموقع** - اختر أي موقع تجاري يتلقى الخدمة
- **تعيين جهة الاتصال** - تعيين جهات الاتصال للفوترة/التقنية
- **الحقول المخصصة** - أرقام الحسابات، مراكز التكلفة، أرقام أوامر الشراء

مثال على سير العمل التجاري:

اختر موقع التثبيت: [مكتب لندن - 123 الشارع الرئيسي ▼]

جهة الاتصال للفوترة: [جين دو - <jane@acme.com> ▼]

جهة الاتصال الفنية: [بوب سميث - <bob@acme.com> ▼]

رقم أمر الشراء: [PO-2025-001234_____]

[متابعة] [إلغاء]

تعيين الخدمة بالجملة

لتعيين ��فس الخطة لعدة عملاء (على سبيل المثال، عمليات النقل الجماعي):

1. استخدم استيراد CSV (إذا كان متاحًا)
2. أو قم بالتعيين بشكل فردي باستخدام إعدادات القالب
3. يجب أن يكون المخزون متاحًا بكميات كبيرة
4. راقب قائمة التوفير لتجنب التحميل الزائد

سير العمل الشائعة

سير العمل 1: خدمة الهاتف المحمول الجديدة

1. يدخل العميل المتجر ويريد خدمة الهاتف المحمول
2. يفتح الموظف **المنتجات** ← **الخطط**
3. يتبدل إلى نوع العميل **الفردي**
4. يختار "**الهاتف المحمول - خطة 20 جيجابايت**"
5. ينقر على "**تعيين للعميل**"
6. يبحث عن العميل برقم الهاتف: "1234567890+"
7. يختار **جون سميث** من النتائج
8. يختار بطاقة SIM من مخزون المختار
9. يختار رقم الهاتف المحمول المتاح
10. يفعّل **التجديد التلقائي**
11. يؤكد ويقوم بالتوفير
12. يراقب التقدم حتى الاكتمال

13. يسلم بطاقة SIM للعميل مع حزمة الترحيب

سير العمل 2: إضافة الإنترنت لعميل موجود

1. انتقل إلى صفحة العميل: **العملاء ← جون سميث**
2. انقر على علامة التبويب **الخدمات**
3. انقر على زر **"إضافة خدمة"**
4. تصفح خطط الإنترنت
5. اختر **"الألياف - 100 ميجابت في الثانية"**
6. اختر موقع التثبيت (إذا كان تجاريًا/عدة مواقع)
7. اختر مودم CPE من مخزون
8. حدد تاريخ التثبيت (إذا لزم الأمر)
9. قم بتوفير الخدمة
10. أنشئ تذكرة التثبيت

سير العمل 3: تعيين إضافة زيادة البيانات

1. يتصل العميل: "أحتاج المزيد من البيانات"
2. يبحث الموظف عن العميل في البحث العالمي
3. يفتح علامة تبويب خدمات العميل
4. ينقر على "الإضافات" بجوار خدمة الهاتف المحمول
5. يختار "زيادة بيانات 5 جيجابايت"
6. يتم تفويض الدفع
7. يتم توفير الإضافة على الفور
8. يتلقى العميل زيادة البيانات الفورية

استكشاف الأخطاء وإصلاحها

"لا توجد منتجات مت��حة"

- **السبب:** تستبعد الفلاتر جميع المنتجات
- **الحل:**
  - قم بتبديل نوع العميل (فردي مقابل تجاري)
  - تحقق من أن كتالوج المنتجات يحتوي على منتجات مفعلة

- تحقق من أن المنتجات تتطابق مع أهلية العميل

"لا يوجد مخزون متاح"

- **السبب:** عناصر المخزون غير متوفرة أو جميعها مخصصة
- **الحل:**
  - أضف المزيد من عناصر المخزون إلى النظام
  - تحقق من حالة العنصر (يجب أن تكون "جديدة" أو "في المخزون")
  - تحقق من أن أسماء قوالب المخزون تتطابق مع متطلبات المنتج

"العميل غير موجود"

- **السبب:** العميل غير موجود أو مصطلح البحث غير صحيح
- **الحل:**
  - أنشئ العميل أولاً
  - جرب مصطلحات بحث مختلفة (ID، هاتف، بريد إلكتروني)
  - تحقق من الأخطاء المطبعية

التوفير يتعثر

- **السبب:** خطأ في الكتاب أو النظام الخارجي غير متاح
- **الحل:**
  - انتظر انتهاء المهلة (عادةً 5 دقائق)
  - تحقق من سجلات التوفير للخطأ المحدد
  - تحقق من أن أنظمة OCS والشبكة متصلة
  - أعد محاولة التوفير بعد إصلاح المشكلة

التوفير ينجح لكن الخدمة لا تعمل

- **السبب:** تم إنشاء حساب OCS لكن الشبكة لم يتم تحديثها
- **الحل:**
  - تحقق من أن OCS لديه حساب
  - تحقق من تنشيط بطاقة SIM في HLR/HSS
  - تحقق من توفير الشبكة (RADIUS، DPI، إلخ)
  - راجع مهام الكتاب للخطوات المفقودة

أفضل الممارسات

قبل التعيين:

- تحقق من أن العميل لديه طريقة دفع صالحة في الملف
- تأكد من أهلية العميل للمنتج (سكنية مقابل تجارية)
- تأكد من توفر المخزون المطلوب
- راجع شروط المنتج والتسعير مع العميل

أثناء التعيين:

- تحقق مرة أخرى من اختيار العميل قبل التأكيد
- اختر عناصر المخزون الصحيحة (تحقق من أرقام السلاسل)
- فعل التجديد التلقائي للراحة (ما لم يعترض العميل)
- راقب تقدم التوفير حتى الاكتمال

بعد التعيين:

- تحقق من ظهور الخدمة في قائمة خدمات العميل
- تحقق من أن حالة الخدمة "نشطة"
- تأكد من تعيين المخزون بشكل صحيح
- أرسل بريد الترحيب أو التعليمات للعميل
- اختبر الخدمة إذا أمكن (قم بإجراء مكالمة اختبار، تحقق من البيانات)

بالنسبة للإضافات:

- تأكد من أن الإضافة متوافقة مع الخدمة الحالية
- اشرح الفوترة (لمرة واحدة مقابل متكررة)
- تحقق من الدفع قبل التوفير
- تحقق من تحديث الرصيد على الفور بعد توفير الإضافة

الوثائق ذات الصلة

- `csa_add_service` - نظرة عامة على الخدمات
- `concepts_products_and_services` - مفاهيم كتالوج المنتجات

- `administration_inventory` - إدارة عناصر المخزون
- `concepts_provisioning` - تفاصيل نظام التوفير
- `guide_product_lifecycle` - دورة حياة المنتج الكاملة بما في ذلك التوفير

سجلات تفاصيل المكالمات (CDRs)

تظهر سجلات تفاصيل المكالمات **المكالمات الفردية، رسائل SMS، وجلسات البيانات** لخدمة معينة. حيث تُظهر استخدام الخدمة مقدار الرصيد المتبقي، تُظهر سجلات المكالمات الأحداث التي استخدمته: من تم الاتصال به، ومتى، ومدة المكالمة، وما تكلفته.

فتح سجلات المكالمات

توجد سجلات المكالمات في نافذة **استخدام الخدمة**، بجوار عرض الرصيد.

من خدمة:

1. انقر على زر **الرصيد** (أو **الاستخدام**) بجوار الخدمة
2. اختر علامة التبويب **سجلات المكالمات (CDRs)**

يمكنك أيضًا فتحها من زر **عرض CDRs** في شاشة تحرير الخدمة. بالنسبة للعملاء، افتحها من **سجلات المكالمات** في صفحة استخدام بوابة الخدمة الذاتية.

قراءة القائمة

تُدرج السجلات مع **الأحدث أولاً**. كل صف يظهر:

العمود	ما يظهره
وقت الإعداد	متى بدأت المكالمة أو الرسالة أو الجلسة
الوجهة	الرقم المتصل به، أو مستلم SMS، أو وجهة البيانات
الفئة	ما إذا كانت **مكالمة**، **رسالة** **SMS**، أو حدث **بيانات**
الاستخدام	مدة المكالمة (`HH:MM:SS`)، كمية البيانات (MB/GB)، أو عدد رسائل SMS
التكلفة	ما تم فرضه مقابل الحدث

استخدم **السابق** و **التالي** في الأسفل للتنقل عبر السجلات القديمة.

انقر على أي صف لفتح التفاصيل الكاملة لذلك السجل، بما في ذلك وقت الرد والمعلومات الإضافية المسجلة ضده.

العثور على سجلات محددة

استخدم شريط الفلتر في الأعلى لتضييق القائمة، ثم انقر على **تطبيق**:

- **أزرار سريعة (1 ساعة / 24 ساعة / 7 أيام / 30 يوم)**: تظهر السجلات من الساعة الأخيرة، أو اليوم، أو الأسبوع، أو الشهر.
- **من / إلى**: اختر تاريخ ووقت البداية والنهاية الخاصين بك.
- **الوجهة تبدأ بـ**: اكتب الأرقام الأولى من الرقم لعرض المكالمات المطابقة فقط (على سبيل المثال، `44` لأرقام المملكة المتحدة).
- **الفئة**: عرض فقط **المكالمات**، **رسائل** **SMS**، أو **البيانات** (أو **الكل**).

إعادة تعيين clears the filters and shows the most recent records again.

> **لا ترى أي شيء؟** تفتح العرض على السجلات الأحدث، ولكن إذا اخترت نطاق تاريخ، فقد لا تكون الخدمة قد شهدت أي نشاط خلال تلك الفترة. جرب **إعادة تعيين**، أو وسع تواريخ **من / إلى**.

تنزيل السجلات

لتنزيل السجلات المعروضة حاليًا (بعد تطبيق الفلاتر) كملف جدول بيانات. **CSV** انقر على **تصدير**. هذا مفيد للمشاركة مع عميل أو الاحتفاظ بسجل لاستفسار الفواتير.

من يمكنه رؤية سجلات المكالمات

- **الموظفون** يحتاجون إلى إذن سجلات المكالمات. إذا لم ترَ علامة التبويب **سجلات المكالمات**، اطلب من مسؤولك تمكينها لحسابك.
- **العملاء** يمكنهم رؤية السجلات لخدماتهم **فقط** من خلال بوابة الخدمة الذاتية.

سجلات المكالمات هي **للعرض فقط**. لا يمكن تعديلها أو حذفها من هنا.

تعديل خدمة

يمكن تعديل الخدمات من قبل العميل النهائي عبر بوابة العناية الذاتية، أو من قبل مسؤول عبر بوابة الإدارة.

بمجرد توفير الخدمة، يمكنك تعديل معلماتها، إضافة تحسينات، أو تغيير الإعدادات.

تحرير معلمات الخدمة

يمكن تعديل معلمات الخدمة الأساسية من خلال النقر على زر **تحرير** في صفحة تفاصيل الخدمة.

الحقول القابلة للتحرير:

- اسم الخدمة
- حالة الخدمة (نشطة، غير نشطة، معلقة)
- ملاحظات الخدمة

- إعدادات التجديد التلقائي
- العناصر المرتبطة بالمخزون
- حقول مخصصة

يتم حفظ التغييرات على الفور وتظهر في بوابة العناية الذاتية لل��ميل.

إضافة ملحقات الخدمة

تعمل الملحقات على تعزيز الخدمات الحالية بميزات إضافية، أو حزم بيانات، أو أجهزة. تشمل أنواع الملحقات الشائعة:

- **إعادة شحن البيانات** - حزمة بيانات إضافية (مثل، "5" GBمثل، "تعزيز بيانات")
- **ترقيات الميزات** - قدرات إضافية (مثل، "المكالمات الدولية")
- **تأجير الأجهزة** - إضافات المعدات (مثل، "مودم" WiFi 6")
- **خدمات متميزة** - ميزات محسّنة (مثل، "عنوان IP ثابت")

الوصول إلى كتالوج الملحقات

من صفحة تفاصيل الخدمة:

1. انتقل إلى الخدمة التي ترغب في تعزيزها
2. انقر على زر **"إضافة ملحق"** أو **"تصفح الملحقات"**
3. يفتح كتالوج الملحقات، مصفّى لعرض الملحقات المتوافقة فقط

التصفية التلقائية:

يقوم النظام بتصفية الملحقات تلقائيًا بناءً على:

- **نوع الخدمة** - يظهر فقط الملحقات التي تتطابق مع نوع الخدمة (محمول، إنترنت، VoIP، إلخ.)
- **نوع العميل** - تصفية حسب العميل السكني مقابل العميل التجاري
- **توافق الخدمة** - يتحقق مما إذا كانت الخدمة تلبي متطلبات الملحق
- **التوفر** - يظهر فقط المنتجات المفعلة

على سبيل المثال، إذا كنت تعرض خدمة موبايل سكنية، سترى فقط الملحقات المحمولة المخصصة للعملاء السكنيين.

عرض الملحقات

تظهر الملحقات في دوار تفاعلي يعرض:

عرض بطاقة المنتج:

المعلومات المعروضة:

- أيقونة المنتج
- اسم المنتج
- قائمة الميزات (نقاط)
- تكلفة الإعداد
- التكلفة الشهرية/الدورية
- رابط الشروط والأحكام
- زر "إضافة إلى الخدمة"

توفير ملحق

الخطوة 1: اختيار الملحق

انقر على بطاقة الملحق المرغوبة، ثم انقر "**إضافة إلى الخدمة**"

الخطوة 2: اختيار المخزون (إذا لزم الأمر)

إذا كان الملحق يتطلب مخزونًا ماديًا (مثل، تأجير الأجهزة)، ي��هر محدد المخزون:

> أجهزة المودم المتاحة: • مودم-12345 - TP-Link AX1800 (جديد) • مودم-12346 - TP-Link AX1800 (جديد) • مودم-12347 - Netgear RAX40 (جديد)

اختر العنصر المحدد من المخزون لتعيينه لهذه الخدمة.

الخطوة 3: تكوين التجديد التلقائي (اختياري)

بالنسبة للملحقات المتكررة، قد يُطلب منك:

> هل ترغب في تمكين التجديد التلقائي لهذا الملحق؟
>
> [نعم] [لا]

- **نعم** - يتم تجديد الملحق تلقائيًا في كل فترة فواتير
- **لا** - شراء لمرة واحدة، يجب على العميل تجديده يدويًا

الخطوة 4: تأكيد وتوفير

"تأكيد" راجع تفاصيل الملحق وانقر"

تظهر نافذة التوفير مع عرض التقدم في الوقت الفعلي:

> ✓ التحقق من الدفع ✓ تعيين المخزون ⟳ تكوين الخدمة (قيد التقدم...) إنشاء المعاملة إرسال بريد التأكيد

بمجرد الانتهاء، يظهر الملحق في قائمة الملحقات الخاصة بالخدمة.

التوفير خلف الكواليس:

عند إضافة ملحق، يقوم النظام بـ:

1. التحقق من إمكانية شراء العميل للملحق
2. تشغيل كتاب التشغيل الخاص بالملحق (`provisioning_play`)
3. تعيين أي عناصر مخزون مطلوبة للخدمة
4. إنشاء سجل المعاملة للفوترة
5. تحديث تكوين الخدمة (OCS، إلخ.أنظمة الشبكة،)
6. إرسال إشعار تأكيد للعميل

للحصول على تفاصيل فنية حول توفير الملحقات، راجع دليل دورة حياة المنتج الكامل - إضافة الملحقات.

عرض الملحقات النشطة

تظهر الملحقات النشطة في صفحة تفاصيل الخدمة في قسم **"الملحقات"**:

عرض المثال:

> 5GB تعزيز بيانات
> أضيفت: 10 يناير 2025 تنتهي: 17 يناير 2025 الحالة: نشطة التكلفة: £5.00
>
> WiFi 6 تأجير مودم
> أضيفت: 01 ديسمبر 2024 المخزون: مودم-12345 التجديد التلقائي: نعم شهريًا: £10.00

إزالة الملحقات

لإزالة ملحق من خدمة:

1. انتقل إلى صفحة تفاصيل الخدمة
2. حدد الملحق في قس**"الملحقات النشطة"** ��
3. انقر على **"إزالة"** أو **رمز سلة المهملات** بجوار الملحق
4. أكد الإزالة

ماذا يحدث:

- **ملحقات الأجهزة** - يتم تشغيل كتاب التشغيل لإلغاء التوفير، ويتم وضع علامة على المخزون للعودة
- **الملحقات الافتراضية** - تتم إزالة الفوائد على الفور
- **الملحقات المتجددة تلقائيًا** - يتم إلغاء التجديدات المستقبلية
- **تم إنشاء معاملة** - يتم إضافة أي مبلغ مقسوم إذا كان ذلك ممكنًا

> **⚠WARNING**
>
> إعادة المعدات. سيقوم النظام (المودم، CPE، إلخ.) عادةً ما تتطلب إزالة ملحقات الأجهزة بوضع علامة على المخزون للعودة وقد يرسل تعليمات العودة إلى العميل.

سيناريوهات الملحقات الشائعة

السيناريو 1: نفاد بيانات العميل

1. يتصل العميل بالدعم: "لقد استخدمت حزمة البيانات الشهرية الخاصة بي"
2. يتنقل الموظف إلى خدمة الموبايل الخاصة بالعميل
3. ينقر على "إضافة ملحق"
4. يختار "5GB تعزيز بيانات"
5. يتم التوفير على الفور
6. يتلقى العميل تعزيز بيانات فوري

السيناريو 2: تحتاج الأعمال إلى IP ثابت

1. يطلب العميل التجاري عنوان IP ثابت لـ VPN
2. يفتح الموظف خدمة الإنترنت الخاصة بالعميل
3. يتصفح الملحقات، ويختار "عنوان IP ثابت"
4. يقوم النظام بتوفير IP (المخزون) من المجموعة المتاحة
5. يقوم بتكوين التوجيه في معدات الشبكة
6. تتلقى الأعمال تفاصيل تكوين IP

السيناريو 3: تأجير معدات لخدمة الألياف

1. يسجل العميل للحصول على إنترنت الألياف
2. أثناء التوفير، يختار العميل تأجير مودم
3. يضيف الموظف ملحق "مودم WiFi 6"
4. يختار مودم متاح من المخزون
5. يتم شحن المودم إلى العميل
6. يتم إضافة رسوم متكررة بقيمة £10 شهريًا إلى الحساب

استكشاف الأخطاء وإصلاحها

"لا توجد ملحقات متوافقة متاحة"

- **السبب:** لا توجد ملحقات تتطابق مع نوع الخدمة أو نوع العميل

- **الحل:** تحقق من كتالوج المنتجات يحتوي على ملحقات تتطابق مع إعدادات `service_type` و `residential`/`business`

فشل توفير الملحق

- **السبب:** خطأ في كتاب التشغيل الخاص بالتوفير أو عدم توفر المخزون
- **الحل:**
  - تحقق من سجلات التوفير للخطأ المحدد
  - تحقق من أن عناصر المخزون متاحة (إذا لزم الأمر)
  - راجع سجلات كتاب التشغيل في نظام التوفير

محدد المخزون لا يظهر أي عناصر

- **السبب:** لا توجد عناصر مخزون متاحة من النوع المطلوب
- **الحل:**
  - أضف عناصر المخزون إلى النظام
  - تحقق من أن العناصر الموجودة ليست جميعها مخصصة أو تالفة
  - تحقق من أن اسم قالب المخزون يتطابق مع `inventory_items_list` تمامًا

التجديد التلقائي لا يعمل

- **السبب:** لم يتم تعيين علامة التجديد التلقائي أو انتهت صلاحية طريقة الدفع
- **الحل:**
  - تحقق من أن الخدمة تحتوي على `auto_renew: true`
  - ت��ق من أن العميل لديه طريقة دفع صالحة في السجل
  - راجع الوظائف المجدولة في نظام الفوترة

الوثائق ذات الصلة

- `guide_product_lifecycle` - تدفق توفير الملحقات الكامل
- `csa_add_service` - إنشاء خدمات جديدة
- `concepts_products_and_services` - مفاهيم المنتجات والخدمات
- `administration_inventory` - إدارة المخزون للملحقات الأجهزة

إدارة الخدمة

واجهة إدارة الخدمة توفر أدوات شاملة لعرض وتصفية وإدارة خدمات العملاء عبر مؤسستك.

المستندات ذات الصلة: إضافة خدمات، استخدام الخدمة، تعديل الخدمات، المنتجات والخدمات.

الوصول إلى قائمة الخدمات

انتقل إلى:

أو مباشرة:

هذا يعرض جميع الخدمات مع قدرات التصفية والترتيب والإدارة الجماعية.

نظرة عامة على قائمة الخدمات

تعرض قائمة الخدمات الخدمات في جدول مع:

الأعمدة:

- **معرف الخدمة** - معرف فريد
- **اسم الخدمة** - اسم وصفي (على سبيل المثال، "محمول - +44 7700 900123")

- **اسم العميل** - العميل المرتبط (رابط قابل للنقر)
- **نوع الخدمة** - محمول، VoIP، إنترنت، IPTV
- **الحالة** - نشط، غير نشط، معلق
- **تاريخ البدء** - متى بدأت الخدمة
- **تاريخ الانتهاء** - متى تنتهي/تنتهي الخدمة
- **التكلفة الشهرية** - السعر التجزئة
- **الإجراءات** - عرض، تعديل، إضافات، استخدام، حذف

تصفية الخدمات

علامات الحالة

تصفية سريعة للخدمات حسب الحالة باستخدام شريط علامات التبويب:

- **جميع النتائج** - يظهر جميع الخدمات بغض النظر عن الحالة
- **نشط** - الخدمات النشطة حالياً (عرض افتراضي)
- **غير نشط** - الخدمات الملغاة أو المنتهية
- **معلق** - الخدمات المعطلة مؤقتاً (عدم الدفع، الاحتيال، إلخ)

النقر على علامة تبويب يحدث القائمة على الفور. يتم تمييز علامة التبويب النشطة.

فلتر نوع الخدمة

تصفية حسب نوع الخدمة:

اختر نوعًا أو أكثر لعرض الخدمات المطابقة فقط.

حالات الاستخدام:

- **محمول** - عرض جميع خدمات بطاقة SIM المحمولة
- **إنترنت** - عرض الألياف، DSL، واللاسلكي الثابت
- **IPTV** - عرض خدمات الاشتراك في التلفزيون
- **VoIP** - قائمة خدمات الصوت عبر الإنترنت

فلتر العميل

البحث والتصفية حسب العميل:

ابدأ بكتابة اسم العميل لرؤية النتائج المطابقة:

اختر عميلًا لعرض خدماته فقط.

حالة الاستخدام: عرض جميع الخدمات لعميل معين بسرعة.

البحث

بحث عالمي عبر جميع حقول الخدمة:

يبحث في:

- اسم الخدمة
- معرف الخدمة
- اسم العميل
- أرقام الهواتف (لخدمات المحمول)
- UUID الخدمة

مثال: ابحث عن "0770" للعثور على جميع الخدمات التي تحتوي أرقام هواتفها على "0770".

ترتيب الخدمات

انقر على أي رأس عمود لترتيب حسب هذا الحقل:

الأعمدة القابلة للفرز:

- معرف الخدمة (افتراضي: الأحدث أولاً)
- اسم الخدمة (أبجدي)
- اسم العميل (أبجدي)
- نوع الخدمة (أبجدي)
- تاريخ البدء (زمني)
- تاريخ الانتهاء (زمني)
- التكلفة الشه��ية (رقمي)

اتجاه الفرز:

- انقر مرة واحدة: تصاعدي (A-Z، الأدنى-الأعلى، الأقدم-الأحدث)
- انقر مرتين: تنازلي (Z-A، الأعلى-الأدنى، الأحدث-الأقدم)
- يظهر عمود الفرز النشط مؤشر ▲ أو ▼

قائمة الفرز المنسدلة:

تقدم القائمة المنسدلة في الزاوية العليا اليمنى إعدادات فرز سريعة:

عرض تفاصيل الخدمة

انقر على أي اسم خدمة لفتح عرض تفاصيل الخدمة.

علامات تفاصيل الخدمة:

1. **نظرة عامة** - ملخص الخدمة، الحالة، التواريخ، التسعير
2. **المخزون** - المعدات المعينة (بطاقات SIM، إلخ، المودمات.)
3. **المعاملات** - الرسوم، الائتمانات، المدفوعات
4. **الاستخدام**/SMS - إحصائيات استخدام البيانات/الصوت
5. **الإضافات** - الإضافات النشطة والمتاحة
6. **سجل النشاط** - تاريخ التغييرات

إجراءات سريعة

من قائمة الخدمة، انقر على قائمة الإجراءات (⋮) للعمليات السريعة:

- **عرض** - يفتح تفاصيل الخدمة
- **تعديل** - تعديل معلمات الخدمة
- **إضافات** - تصفح وإضافة تحسينات الخدمة
- **استخدام** - عرض الاستخدام الحالي والرصيد
- **حذف** - إلغاء/إزالة الخدمة

العمليات الجماعية

اختر خدمات متعددة باستخدام مربعات الاختيار لأداء إجراءات جماعية.

اختيار الخدمات

اختيار فردي:

انقر على مربع الاختيار بجوار كل خدمة تريد اختيارها.

اختيار الكل:

انقر على مربع الاختيار في رأس الجدول لاختيار جميع الخدمات المرئية.

```
☑ الخدمة 1 ☑ الخدمة 2 ☑ الخدمة 3
```

عداد الاختيار:

تظهر الواجهة عدد الخدمات المختارة:

الإجراءات الجماعية

بمجرد اختيار الخدمات، تظهر أزرار الإجراءات الجماعية:

حذف خدمات متعددة:

1. اختر الخدمات المراد حذفها
2. انقر على زر **"حذف المحدد"**
3. أكد الحذف في النافذة المنبثقة

> **⚠WARNING**
> الحذف الجماعي دائم وسيفعل:
>
> - إلغاء جميع ال��دمات المختارة
> - وضع المخزون كغير معين
> - إيقاف جميع الرسوم المتكررة
> - إنشاء إدخالات سجل النشاط

حالات الاستخدام:

- تنظيف خدمات الاختبار
- إلغاء الخدمات لحسابات العملاء المغلقة
- إزالة الإدخالات المكررة أو الخاطئة

أفضل ممارسة: استخدم الفلاتر لتقليل القائمة قبل العمليات الجماعية لتجنب الحذف العرضي.

إدارة حالة الخدمة

أنواع الحالة

يمكن أن تكون للخدمات ثلاث حالات:

نشط

- الخدمة تعمل
- تطبق الرسوم
- يمكن للعميل استخدام الخدمة
- تظهر بشارة خضراء

غير نشط

- تم إلغاء الخدمة أو انتهت
- لا تطبق الرسوم
- لا يمكن للعميل استخدام الخدمة
- تظهر بشارة رمادية
- المخزون محدد للعودة/إعادة التصنيع

معلق

- الخدمة معطلة مؤقتاً
- قد يتم فرض رسوم أو لا (قابل للتكوين)
- لا يمكن للعميل استخدام الخدمة
- تظهر بشارة برتقالية/صفراء
- الأسباب ال��ائعة: عدم الدفع، التحقيق في الاحتيال، طلب العميل

تغيير حالة الخدمة

لتغيير حالة الخدمة:

1. افتح تفاصيل الخدمة
2. انقر على زر **"تعديل"**
3. غير قائمة **"حالة الخدمة"**
4. انقر على **"حفظ"**

ماذا يحدث:

- **نشط → غير نشط:** يشغل خطة إلغاء التخصيص (إذا كانت مكونة)
- **نشط → معلق:** يعطل الخدمة في OCS/السجل بالشبكة ولكن يحتفظ
- **معلق → نشط:** يعيد تفعيل الخدمة، يستأنف الفوترة
- **غير نشط → نشط:** قد يشغل إعادة التخصيص (استخدم الحذر)

عرض استخدام الخدمة

انقر على **"استخدام"** في قائمة الإجراءات لفتح نافذة الاستخدام.

المعلومات المعروضة:

- **الرصيد** - الائتمان المتبقي أو القيمة المدفوعة مسبقاً
- **استخدام البيانات** - المستخدم مقابل المخصص (شريط التقدم)
- **استخدام الصوت** - الدقائق المستخدمة مقابل المخصص في الخطة

- **استخدام SMS** - الرسائل المرسلة مقابل المخصص
- **تاريخ الانتهاء** - متى تنتهي الرصيد/الخطة الحالية

الإجراءات:

- **إعادة شحن** - إضافة رصيد أو بيانات (لخدمات الدفع المسبق)
- **عرض التفاصيل** - رؤية تفاصيل الاستخدام
- **تصدير** - تنزيل تقرير الاستخدام (إذا كان متاحاً)

التجديدات التلقائية القادمة

عرض التجديدات التلقائية القادمة يوفر واجهة مركزية لمراقبة وإدارة جميع تجديدات الخدمات المجدولة عبر مؤسستك.

الوصول إلى التجديدات التلقائية

انتقل إلى:

أو مباشرة:

هذا يعرض جميع الخدمات مع التجديد التلقائي المجدول، مرتبة حسب تاريخ التجديد التالي.

نظرة عامة على التجديدات التلقائية

تعرض قائمة التجديدات التلقائية التجديدات المجدولة في جدول مع:

الأعمدة:

- **العميل** - اسم العميل (رابط قابل للنقر إلى نظرة عامة على العميل)
- **الخدمة** - اسم الخدمة (رابط قابل للنقر إلى نظرة عامة على العميل)
- **المنتج** - اسم المنتج/الخطة التي يتم تجديدها
- **ال��كلفة** - تكلفة التجديد (من سعر التجزئة للمنتج)
- **يتجدد في** - تاريخ ووقت التجديد التالي بتنسيق مقروء بشري
- **الحالة** - حالة الخدمة (نشطة، معلق، إلخ.)
- **الإجراءات** - تجديد الآن أو إزالة التجديد التلقائي

مثال عرض:

كيف يعمل التجديد التلقائي

يتم جدولة التجديد التلقائي في CGRateS (نظام الفوترة). النظام عند إنشاء خدمة أو تعديلها:

1. ينشئ خطة إجراء في CGRateS مع تاريخ التجديد - **جدولة الإجراء**
2. يتحقق مما إذا كان لدى العميل رصيد كافٍ قبل التجديد - **مراقبة الرصيد**
3. في التاريخ المجدول، يجدد الخدمة تلقائياً - **تنفيذ التجديد**
4. ينشئ معاملات، يحدث تواريخ الخدمة، ويسجل النشاط - **تحديث السجلات**

مصدر البيانات:

يستعلم CGRateS مباشرة باستخدام عرض التجديدات التلقائية القادمة `ApierV1.GetScheduledActions` API ويغني البيانات بمعلومات العميل والخدمة من قاعدة بيانات CRM.

تجديد الآن

لتفعيل التجديد يدويًا قبل التاريخ المجدول:

1. انقر على زر **(تجديد الآن)** للخدمة

2. راجع تفاصيل التجديد في نافذة التأكيد:

> المنتج: إنترنت الخدمة: FixedWireless_75628fa5 العميل: Acme Corp
> منزلي ميغا التكلفة: $89.99
>
> التجديد المجدول التالي: 11 نوفمبر 2025 10:45 صباحًا

> [تأكيد التجديد] [إلغاء]

3. انقر على **"تأكيد التجديد"** للمعالجة الفورية
4. يتم إنشاء وتنفيذ وظيفة تخصيص
5. راقب التقدم في نافذة حالة التخصيص

ماذا يحدث:

- يتم إعادة شحن الخدمة بمخصصات المنتج
- يتم تحديث الرصيد في CGRateS
- يتم إنشاء سجل المعاملة
- يتم تمديد تاريخ انتهاء الخدمة
- يتم إنشاء إدخال سجل النشاط

حالات الاستخدام:

- طلب العميل تجديد مبكر
- حل مشكلات الخدمة عن طريق إعادة تعيين المخصصات
- اختبار عمليات التجديد
- يريد العميل استخدام الخدمة قبل التجديد المجدول

إزالة التجديد التلقائي

لإلغاء التجديد التلقائي لخدمة:

1. انقر على زر **(إزالة التجديد التلقائي)** للخدمة
2. أكد الحذف في النافذة:

> سيؤدي ذلك إلى إزالة التجديد التلقائي المجدول لـ:
>
> العميل: Acme Corp الخدمة: FixedWireless_75628fa5 التجديد التالي: 11 نوفمبر 2025 10:45 صباحًا
>
> لن يتم تجديد الخدمة تلقائيًا. ستحتاج إلى تجديدها يدويًا أو ستنتهي الخدمة في تاريخ الانتهاء.

[إزالة التجديد التلقائي] [إلغاء]

3. انقر على **"إزالة التجديد التلقائي"** للتأكيد
4. تتم إزالة خطة الإجراء من CGRateS
5. ستنتهي الخدمة بشكل طبيعي ما لم يتم تجديدها يدويًا

ماذا يحدث:

- يتم حذف خطة الإجراء من CGRateS
- تظل حالة الخدمة دون تغيير
- ستنتهي الخدمة في تاريخ انتهائها الحالي
- يجب على العميل تجديد الخدمة يدويًا أو ستتوقف الخدمة

حالات الاستخدام:

- العميل يلغي الخدمة
- الانتقال إلى عملية التجديد اليدوي
- تغيير خطة الخدمة
- طلب العميل إيقاف الرسوم التلقائية

> ⚠ **WARNING**
> إزالة التجديد التلقائي تعني أن الخدمة ستنتهي ما لم يتم تجديدها يدويًا. لن يتلقى العملاء استمرار الخدمة تلقائيًا.

فهم العرض

عمود العميل:

- يظهر اسم العميل كرابط قابل للنقر
- روابط إلى صفحة نظرة عامة على العميل
- يظهر "N/A" (خدمة يتيمة) إذا كانت الخدمة غير مرتبطة بعميل

عمود الخدمة:

- يظهر اسم الخدمة/UUID

- روابط إلى صفحة نظرة عامة على العميل
- الخدمة إذا لم يتم تعيين اسم UUID يظهر
- إذا لم يمكن العثور على الخدمة في قاعدة البيانات "N/A" يظهر

عمود المنتج:

- اسم المنتج من خطة إجراء CGRateS
- مستخرج من بيانات التعريف الخاصة بالإجراء المجدول
- يظهر دائمًا اسم المنتج حتى إذا لم يتم العثور على الخدمة

عمود التكلفة:

- تكلفة التجزئة من جدول المنتج في CRM
- المبلغ الذي سيتم فرضه عند التجديد
- إذا لم يتم العثور على المنتج في قاعدة البيانات "N/A" يظهر

عمود يتجدد في:

- التاريخ والوقت الدقيق للتجديد التالي
- الوقت النسبي المقروء بشريًا (على سبيل المثال، "بعد 3 ساعات"، "بعد 2 يوم")
- يستخدم منطقة زمنية خادم (مدرك للمنطقة الزمنية CGRateS)

عمود الحالة:

- الحالة الحالية للخدمة من قاعدة بيانات CRM
- **نشط** (أخضر) - الخدمة تعمل
- **معلق** (برتقالي) - الخدمة معطلة مؤقتًا
- **غير معروف** (رمادي) - الخدمة غير موجودة في قاعدة البيانات أو الحالة غير محددة

استكشاف الأخطاء وإصلاحها

الخدمات تظهر "N/A" للعميل/الخدمة

- **السبب:** UUID الخدمة في CGRateS لا يتطابق مع service_uuid في قاعدة بيانات CRM
- **الإصلاح:**
  - تحقق من وجود الخ��مة في قاعدة البيانات

    - تحقق من أن تنسيق service_uuid يتطابق: `ServiceType_UUID` (على سبيل المثال، `FixedWireless_75628fa5`)
    - ولكن لا تزال مجدولة في CRM قد تكون الخدمة قد حذفت من CGRateS

التكلفة تظهر "N/A"

- **السبب:** معرف المنتج في CRM غير موجود في جدول منتجات CGRateS
- **الإصلاح:**
    - تحقق من وجود المنتج في قاعدة البيانات
    - تحقق من أن product_id في خطة الإجراء يتطابق مع منتج في CRM
    - قد يكون المنتج قد حذفت

لا يظهر التاريخ "يتجدد في"

- **السبب:** NextRunTime غير موجود في استجابة CGRateS
- **الإصلاح:**
    - تحقق من تكوين خطة إجراء CGRateS
    - تحقق من أن ActionTiming مكون بشكل صحيح
    - تحقق من سجلات CGRateS للأخطاء

فشل "تجديد الآن"

- **السبب:** أخطاء تخصيص متنوعة
- **استكشاف الأخطاء:**
    - تحقق من نافذة حالة التخصيص للحصول على تفاصيل الأخطاء
    - تحقق من أن خطط Ansible مكونة بشكل صحيح
    - تحقق من اتصال OCS
    - راجع سجلات التخصيص

تمت إزالة التجديد التلقائي ولكن لا يزال يظهر

- **السبب:** تأخير في التخزين المؤقت أو مشكلة مزامنة CGRateS
- **الإصلاح:**
    - انقر على زر **"تحديث"** لإعادة تحميل البيانات
    - تحقق من أنه تمت إزالة خطة الإجراء فعليًا في CGRateS
    - تحقق من اتصال API CGRateS

الصفحات

تُعرض الخدمات في صفحات لأداء أفضل:

> ◄ السابق 1 [2] 3 4 5 التالي ►
>
> العناصر لكل صفحة: [10 ▼]
>
> - 10
> - 25
> - 50
> - 100

التحكم:

- **السابق/التالي** - التنقل بين الصفحات
- **أرقام الصفحات** - الانتقال إلى صفحة معينة
- **العناصر لكل صفحة** - ضبط عدد الخدمات المعروضة في كل صفحة

نصيحة الأداء: استخدم الفلاتر لتقليل النتائج الإجمالية بدلاً من زيادة العناصر لكل صفحة.

شارات الخدمة ومؤشرات الحالة

تساعد المؤشرات المرئية في تحديد حالات الخدمة بسرعة:

شارات الحالة:

مؤشر التجديد التلقائي:

تظهر الخدمات التي تم تمكين التجديد التلقائي لها:

تنتهي قريبًا:

تظهر الخدمات التي تنتهي خلال 7 أيام:

متأخرة:

تظهر الخدمات التي لديها رصيد مستحق:

تكامل CGRateS (متقدم)

يمكن للمسؤولين إدارة CGRateS (محرك الفوترة والتقييم)، بالنسبة للخدمات المدمجة مع التكوينات المتقدمة مباشرة من عرض الخدمة. يشمل ذلك السمات، الفلاتر، وعرض الجلسات النشطة.

> **ملاحظة**
> تتطلب ميزات تكامل CGRateS إذن **cgrates_api_access**. فقط المسؤولون لديهم حق الوصول إلى هذه الميزات بشكل افتراضي. راجع `rbac` لتكوين الأذونات.

التخصيص التلقائي مقابل الإدارة اليدوية

في التشغيل العادي، يتم تخصيص سمات CGRateS والفلاتر تلقائيًا بواسطة Ansible خلال سير عمل تخصيص الخدمة الأولي. عند إنشاء خدمة جديدة، تقوم خطط التخصيص:

- إنشاء حساب الخدمة في CGRateS
- تكوين السمات (IMSI، MSISDN، إ��خ، معرفات الحساب، ملفات تعريف السرعة.)
- إعداد الفلاتر لضمان التقييم الصحيح
- تطبيق خطط التقييم المناسبة

ومع ذلك، هناك حالات قد تحتاج فيها إلى تعديل هذه التكوينات بعد تخصيص الخدمة:

- **تغييرات ملف تعريف السرعة** - يقوم العميل بترقية/خفض عرض النطاق الترددي (MaxBitrateDL/UL)
- **تعديلات السياسة** - تغيير سياسات QoS أو قواعد تشكيل الحركة (PcefPolicyName)
- **تغييرات رقم الهاتف** - تحديث MSISDN أو معرفات أخرى
- **استكشاف الأخطاء** - إصلاح التكوينات الخاطئة أو اختبار إعدادات مختلفة
- **تكوينات خاصة** - تطبيق سمات مخصصة ليست جزءًا من التخصيص القياسي

تسمح واجهة الإدارة اليدوية للمسؤولين بإجراء هذه التغييرات مباشرة دون إعادة تشغيل سير عمل التخصيص بالكامل. هذا مفيد بشكل خاص لـ:

- **تغييرات سريعة** - تعديل سمة واحدة دون الانتظار لـ التخصيص
- **الاختبار** - تجربة تكوينات مختلفة
- **دعم العملاء** - حل المشكلات بسرع�� أثناء مكالمات الدعم

- **تكوينات مخصصة** - تطبيق إعدادات خاصة بالخدمة غير المغطاة بواسطة القوالب

> **⚠WARNING**
> سير عمل التخصيص القياسي. تأكد من أنك CGRateS تجاوز التغييرات اليدوية على تكوين تفهم تأثير تغييراتك، حيث يمكن أن تؤثر التكوينات الخاطئة على الفوترة ووظائف الخدمة. يتم تسجيل جميع التغييرات في تغذية نشاط العميل لأغراض التدقيق.

الوصول إلى ميزات CGRateS

تظهر ثلاثة أقسام قابلة للطي في أسفل CGRateS، عند عرض أو تعديل خدمة تم تخصيصها في نموذج الخدمة:

- **سمات CGRateS** - تكوين السمات الخاصة بالخدمة
- **فلاتر CGRateS** - تعريف قواعد التصفية للخدمة
- **الجلسات النشطة** - عرض الجلسات النشطة في الوقت الحقيقي

كل قسم مطوي بشكل افتراضي للحفاظ على واجهة نظيفة. انقر على رأس القسم لتوسيعه وعرض/تعديل التكوين.

تظهر الأقسام المطوية شارات العد التي تشير إلى عدد السمات، الفلاتر، أو الجلسات النشطة الموجودة للخدمة.

سمات CGRateS

تسمح السمات لك بتعريف حقول مخصصة وتحويلات تُطبق على أحداث التقييم لهذه الخدمة المحددة.

تنسيق معرف السمة: `ATTR_ACCOUNT_{service_uuid}`

مثال: بالنسبة للخدمة ذات UUID `Mobile_SIM_c2880638`، فإن معرف ملف السمة هو `ATTR_ACCOUNT_Mobile_SIM_c2880638`

إدارة السمات:

1. افتح عرض تعديل الخدمة
2. قم بتوسيع قسم **"CGRateS"** **سمات"**
3. انقر على زر **"تعديل السمات"**
4. أضف/عدل/احذف السمات حسب الحاجة
5. انقر على **"حفظ السمات"**

حقول السمة:

- **المسار** - الحقل الذي سيتم تعديله (على سبيل المثال، `*req.Account`، `*req.IMSI`)

- **النوع** - كيفية تعيين القيمة:
  - `*constant` - تعيين قيمة ثابتة
  - `*variable` - التقاط القيمة من حقول الحدث باستخدام RSRParser
  - `*composed` - إلحاق القيمة ��دلاً من الكتابة فوقها
  - `*usage_difference` - حساب المدة بين حقلين
  - `*sum` - جمع قيم متعددة
  - `*value_exponent` - حساب أس الحقل
- **القواعد** - القيمة(القيم) التي سيتم تطبيقها (يمكن أن تحتوي على قواعد متعددة لكل سمة)

مثال على تكوين السمة:

```
{
  "Path": "*req.Account",
  "Type": "*constant",
  "Value": [{"Rules": "Mobile_SIM_474a380a"}]
}
```

حالات الاستخدام الشائعة:

- تعيين معرف الحساب للتقييم
- ربط IMSI/MSISDN بالخدمة
- تكوين حدود عرض النطاق الترددي (MaxBitrateDL/UL)
- تعيين أسماء السياسات (PcefPolicyName)
- تحويل أو إثراء أحداث التقييم

تسجيل النشاط:

يتم تسجيل جميع تعديلات السمات في تغذية نشاط العميل مع تفاصيل كاملة عن التغييرات التي تم إجراؤها.

فلاتر CGRateS

تحدد الفلاتر قواعد المطابقة التي تحدد متى يجب تطبيق تكوين هذه الخدمة أثناء التقييم.

تنسيق معرف الفلتر: `FLTR_ACCOUNT_{service_uuid}`

فإن معرف الفلتر هو ، `Mobile_SIM_c2880638` UUID **مثال:** بالنسبة ل��خدمة ذات `FLTR_ACCOUNT_Mobile_SIM_c2880638`

إدارة الفلاتر:

1. افتح عرض تعديل الخدمة
2. قم بتوسيع قسم **CGRateS** **فلاتر**"
3. انقر على زر **تعديل الفلاتر**"
4. أضف/عدل/احذف قواعد الفلتر
5. انقر على **حفظ الفلاتر**"

حقول قاعدة الفلتر:

- **العنصر** - الحقل الذي سيتم المطابقة ضده (على سبيل المثال، `~*req.Account`، `~*req.Destination`)
- **النوع** - نوع المطابقة:
  - `*string` - مطابقة نصية دقيقة
  - `*prefix` - يبدأ بالقيمة المحددة
  - `*suffix` - ينتهي بالقيمة المحددة
  - `*empty` - الحقل فارغ
  - `*exists` - الحقل موجود
  - `*notexists` - الحقل غير موجود
  - `*timings` - مطابقة أنماط الوقت/التاريخ
  - `*destinations` - مطابقة أنماط الوجهة
  - `*rsr` - مطابقة حقل RSR
  - `*gt` / `*gte` / `*lt` / `*lte` - مقارنات عددية
- **القيم** - القيمة(القيم) التي سيتم المطابقة معها (يمكن أن تحتوي على قيم متعددة لكل قاعدة)

مثال على تكوين ا��فلتر:

```
{
  "Element": "~*req.Account",
  "Type": "*string",
  "Values": ["Mobile_SIM_474a380a"]
}
```

حالات الاستخدام الشائعة:

- التأكد من أن السمات تنطبق فقط على حساب محدد
- تصفية حسب الوجهة (محلي مقابل دولي)
- تصفية زمنية (ذروة مقابل غير ذروة)
- تصفية حسب نوع الخدمة أو الفئة

تسجيل النشاط:

يتم تسجيل جميع تعديلات الفلاتر في تغذية نشاط العميل.

الجلسات النشطة

عرض الجلسات النشطة في الوقت الحقيقي لهذه الخدمة. يظهر هذا المكالمات الجارية، جلسات البيانات، أو أحداث قابلة للفوترة الأخرى التي تحدث حاليًا.

عرض الجلسات النشطة:

1. افتح عرض تعديل الخدمة
2. قم بتوسيع قسم **"الجلسات النشطة"**
3. عرض قائمة الجلسات النشطة
4. انقر على **"عرض التفاصيل"** في أي جلسة لرؤية بيانات الجلسة الكاملة
5. انقر على **"تحديث"** لإعادة تحميل قائمة الجلسات

معلومات الجلسة المعروضة:

- **وقت الإعداد** - متى بدأت الجلسة
- **الاستخدام** - مدة الجلسة الحالية (بالثواني)
- **الوجهة** - الرقم المتصل أو الوجهة

نافذة تفاصيل الجلسة:

النقر على "عرض التفاصيل" يفتح نافذة تظهر:

- **معلومات أساسية:**
  - CGRID (معرف الجلسة)
  - الحساب (UUID الخدمة)
  - وقت الإعداد
  - الاستخدام/المدة الحالية
  - الوجهة
  - الفئة
- **بيانات الجلسة الكاملة:**
  - تمثيل JSON كامل للجلسة
  - جميع حقول جلسة CGRateS
  - حالة الجلسة في الوقت الحقيقي
  - عارض JSON قابل للتمرير للتفتيش

حالات الاستخدام:

- مراقبة المكالمات النشطة أو جلسات البيانات
- استكشاف مشكلات الفوترة

- التحقق من أن الجلسة يتم تقييمها بشكل صحيح
- التحقق من سمات الجلسة والقيم
- تدقيق استخدام الخدمة النشطة

معدل التحديث:

يتم جلب الجلسات عند الطلب عند توسيع القسم. انقر على "تحديث" للحصول على أحدث بيانات الجلسة.

> **ⓘ ملاحظة**
> فقط الجلسات التي تتطابق مع حساب هذه الخدمة (UUID الخدمة) هي المعروضة. يتم تطبيق الفلتر `*string:~*req.Account:{service_uuid}` بشكل تلقائي.

وكيل API CGRateS

تستخدم جميع عمليات CGRateS (السمات، الفلاتر، الجلسات) نقطة نهاية وكيل OmniCRM API:

نقطة النهاية: `POST /crm/ocs/proxy`

الحقول المطلوبة:

- `method` - طريقة API CGRateS (على سبيل المثال، `APIerSv1.GetAttributeProfile`)
- `params` - مصفوفة من المعلمات للطريقة
- `customer_id` - معرف العميل (لتسجيل النشاط)
- `service_id` - معرف الخدمة (لتسجيل النشاط)

الحقول الاختيارية:

- `tenant` - مستأجر CGRateS (يتم تعيينه افتراضيًا من قيمة التكوين)

مثال على الطلب:

```
{
  "method": "APIerSv1.GetAttributeProfile",
  "params": [{"ID": "ATTR_ACCOUNT_Mobile_SIM_c2880638"}],
  "customer_id": 123,
  "service_id": 456
}
```

تكوين المستأجر:

يتم تعيين المستأجر تلقائيًا من ملف تكوين OmniCRM (`crm_config.yaml`) تحت `ocs.ocsTenant`. يضمن ذلك أن جميع عمليات CGRateS تستخدم المستأجر الصحيح دون ترميز القيم في الواجهة الأمامية.

متطلبات الإذن:

يتطلب إذن `cgrates_api_access`. يتم منح هذا الإذن لدور `admin` بشكل افتراضي.

تسجيل النشاط:

يتم تسجيل جميع عمليات API CGRateS غير GET تلقائيًا في تغذية نشاط العميل، بما في ذلك:

- طريقة API التي تم استدعاؤها
- المستأجر المستخدم
- المعلمات الكاملة المرسلة
- معرف الخدمة التي تم تنفيذ العملية عليها
- المستخدم الذي نفذ العملية
- الطابع الزمني

هذا ينشئ مسار تدقيق كامل لجميع تغييرات تكوين CGRateS.

استكشاف أخطاء تكامل CGRateS

"إذن مرفوض" عند الوصول إلى ميزات CGRateS

- **السبب:** يفتقر المستخدم إلى إذن `cgrates_api_access`
- **الإصلاح:** منح الإذن لدور المستخدم (عادةً ميزة خاصة بالمسؤولين فقط)

السم��ت أو الفلاتر لا يتم تحميلها

- أو الملف الشخصي غير موجود CGRateS **السبب:** مشكلة في الاتصال بـ
- **الإصلاح:**
  - في التكوين CGRateS تحقق من اتصال خادم
  - تحقق من أن تكوين المستأجر صحيح
  - API تحقق من وحدة التحكم في المتصفح للحصول على أخطاء
  - قد لا يكون الملف الشخصي موجودًا بعد (سيظهر نموذج فارغ)

التغييرات لا تحفظ

- API CGRateS **السبب:** خطأ في التحقق أو خطأ في
- **الإصلاح:**
  - تحقق من الحقول المطلوبة (المسار، النوع، العنصر)
  - صحيح JSON تحقق من أن تنسيق
  - تحقق من سجل النشاط للحصول على تفاصيل الأخطاء
  - CGRateS راجع سجلات

لا تظهر جلسات نشطة

- **السبب:** لا توجد جلسات نشطة حاليًا لهذه الخدمة
- **الإصلاح:**
  - هذا طبيعي إذا لم تكن الخدمة قيد الاستخدام
  - حاول التحديث بعد بدء جلسة (مكالمة، بيانات، إلخ.)
  - CGRateS الخدمة يتطابق مع الحساب في UUID تحقق من أن

تفاصيل الجلسة لا تتحدث في الوقت الحقيقي

- **السبب:** يتم جلب بيانات الجلسة عند الطلب، وليس مباشرة
- **الإصلاح:** انقر على زر "تحديث" للحصول على أحدث بيانات الجلسة

CGRateS سجل النشاط لا يظهر تغييرات

- فقط (لا يتم تسجيل القراءات GET) **السبب:** يتم تسجيل العمليات غير
- **الإصلاح:** هذا عن قصد - فقط الكتابات/التعديلات تخلق إدخالات نشاط

سير العمل الشائعة

سير العمل 1: العثور على خدمات العميل

1. (اختياري) انقر على **فلتر نوع الخدمة**
2. انقر على **فلتر العميل**
3. اكتب اسم العميل
4. اختر العميل من القائمة المنسدلة
5. راجع خدمات العميل

سير العمل 2: تحديد الخدمات التي ستنتهي قريبًا

1. انقر على علامة **"نشط"**
2. رتب حسب **"تاريخ الانتهاء"** (تصاعدي)
3. تظهر الخدمات التي ستنتهي قريبًا أولاً
4. اتصل بالعملاء للتجديد

سير العمل 3: تنظيف خدمات الاختبار

1. ابحث عن "اختبار" في مربع البحث
2. راجع النتائج للتأكد من أنها بيانات اختبار
3. اختر جميع خدمات الاختبار
4. انقر على **"حذف المحدد"**
5. أكد الحذف

سير العمل 4: تعليق عميل غير دافع

1. انتقل إلى حساب العميل
2. عرض علامة الخدمات
3. اختر جميع الخدمات النشطة
4. غير الحالة إلى "معلق"
5. احفظ التغييرات

سير العمل 5: عرض استخدام الخدمة المحمولة

1. تصفية حسب **نوع الخدمة: محمول**
2. انقر على اسم الخدمة لفتح التفاصيل
3. انقر على علامة **"استخدام"**
4. راجع استهلاك البيانات/الصوت/SMS
5. حدد المستخدمين الكثيفين أو التجاوزات

استكشاف الأخطاء

الخدمات لا تظهر في القائمة

- **السبب:** فلتر الحالة يخفي النتائج
- **الإصلاح:** انقر على علامة "جميع النتائج" لعرض جميع الحالات

لا يمكن العثور على الخدمة من خلال البحث

- **السبب:** مصطلح البحث لا يتطابق مع البيانات المخزنة
- **الإصلاح:**
  - اول البحث الجزئي (على سبيل المثال، "0770" بدلاً من الرقم الكامل)��
  - استخدم فلتر العميل بدلاً من ذلك
  - تحقق من الأخطاء المطبعية

زر الحذف الجماعي معطل

- **السبب:** لم يتم اختيار أي خدمات
- **الإصلاح:** تحقق من المربعات بجوار الخدمات التي تريد حذفها

الترتيب لا يعمل

- **السبب:** العمود غير قابل للفرز
- **الإصلاح:** فقط الأعمدة ذات أيقونات ▲▼ قابلة للفرز

الصفحة تحميلها ببطء

- **السبب:** عدد كبير جدًا من الخدمات للعرض
- **الإصلاح:**

    - تطبيق الفلاتر لتقليل مجموعة النتائج
    - تقليل العناصر لكل صفحة
    - استخدام البحث لتضييق النتائج

المستندات ذات الصلة

- `csa_add_service` - إضافة خدمات جديدة
- `csa_modify` - تعديل الخدمات وإضافة الإضافات
- `concepts_products_and_services` - مفاهيم المنتجات والخدمات
- `basics_customers` - إدارة العملاء

تتبع استخدام الخدمة والرصيد

يوفر نظام استخدام الخدمة **مراقبة في الوقت الحقيقي** لاستهلاك العملاء للبيانات، الصوت، الرسائل القصيرة، والأرصدة النقدية. تتكامل هذه الميزة مع نظام الشحن عبر الإنترنت (OCS) لعرض الاستخدام الحالي، والامتيازات المتبقية، ومعلومات انتهاء صلاحية الرصيد لكل من العملاء والموظفين (عبر بوابة العناية الذاتية).

نظرة عامة

يمكن تتبع الاستخدام:

- **عرض الرصيد في الوقت الحقيقي** - عرض الاستخدام الحالي والامتيازات المتبقية
- **أنواع رصيد متعددة** - تتبع بيانات، صوت، رسائل قصيرة، وأرصدة نقدية في وقت واحد
- **مراقبة انتهاء الصلاحية** - معرفة متى تنتهي صلاحية الأرصدة
- **تحليل الرصيد** - عرض مفصل لأجزاء الرصيد الفردية
- **تحديث تلقائي** - تحديث الاستخدام كل 3 ثوانٍ تلقائيًا

الوصول إلى استخدام الخدمة

من قائمة الخدمات:

1. انتقل إلى **الخدمات ← قائمة الخدمات**
2. انقر على **قائمة الإجراءات (⋮)** بجوار خدمة
3. اختر **"الاستخدام"**

من تفاصيل الخدمة:

1. افتح صفحة تفاصيل الخدمة
2. انقر على علامة التبويب **"الاستخدام"**

من صفحة العميل:

1. افتح نظرة عامة على العميل

2. انتقل إلى علامة التبويب **الخدمات**
3. انقر على **"عرض الاستخدام"** بجوار أي خدمة

يفتح نموذج الاستخدام أو الصفحة ويظهر بيانات الاستهلاك في الوقت الحقيقي. لفحص المكالمات الفردية، الرسائل القصيرة، وجلسات البيانات وراء هذه الأرصدة، انتقل إلى علامة التبويب **سجلات المكالمات (CDRs)**. انظر سجلات تفاصيل المكالمات.

عرض الاستخدام

تظهر واجهة الاستخدام بطاقات ملخص وأشرطة تقدم مفصلة لكل نوع من أنواع الرصيد.

بطاقات الملخص

تعرض الصف العلوي بطاقات عرض سريعة لكل نوع من أنواع الرصيد:

معلومات البطاقة:

- **نوع الرصيد** - أيقونة وملصق (بيانات، صوت، رسائل قصيرة، نقدية)
- **المبلغ المتبقي** - الرصيد الحالي بالوحدات المناسبة
- **وقت انتهاء الصلاحية** - الأيام/الساعات حتى انتهاء صلاحية الرصيد
- **زر المزيد من المعلومات** - انقر للتوسع في التحليل المفصل

أشرطة التقدم

أسفل البطاقات، تظهر أشرطة التقدم الاستهلاك بصريًا مع الأجزاء المملوءة التي تشير إلى الرصيد المتبقي.

ميزات شريط التقدم:

- **مؤشر بصري** - الجزء المملوء يظهر الرصيد المتبقي
- **النسبة المئوية** - النسبة المئوية العددية من الرصيد المتبقي
- **القيم المطلقة** - تظهر المستخدمة مقابل الإجمالي (على سبيل المثال، "12.5 GB / 20GB")
- **ترميز الألوان**:
  - أخضر: >50% متبقية
  - أصفر: 20-50% متبقية
  - أحمر: <20% متبقية
- **قابل للنقر** - انقر للتوسع في التحليل المفصل

أنواع الرصيد

رصيد البيانات

يتتبع استهلاك بيانات الإنترنت.

الوحدات: جيجابايت (GB) أو ميجابايت (MB)

تنسيق العرض:

> التقدم: 12.5 GB / 20GB (62%)

السيناريوهات الشائعة:

- **خطط بيانات الهاتف المحمول** - 5 من الامتيازات الشهرية GB، 10GB، 20GB
- **اللاسلكي الثابت** - غير محدود أو محدد بمبالغ عالية (500GB، 1TB)
- **إعادة الشحن** - بيانات إضافية تم شراؤها في منتصف الدورة
- **خدمات الدونجل** - بيانات مدفوعة مسبقًا لأجهزة النقاط الساخنة

أجزاء متعددة:

غالبًا ما تحتوي الخدمات على أرصدة بيانات متعددة:

- الامتياز الشهري (تنتهي شهريًا)
- بيانات إضافية (تنتهي بعد فترة الحملة)
- بيانات إعادة الشحن (انتهاء صلاحية أقصر، تُستهلك أولاً)

رصيد الصوت

يتتبع دقائق المكالمات الهاتفية.

الوحدات: دقائق (min)

تنسيق العرض:

> التقدم: 125 min / 500 min (25%)

تتبع مدة المكالمات:

- المكالمات الواردة (إذا كانت مدفوعة)
- المكالمات الصادرة
- المكالمات الدولية (دلو منفصل إذا كان ذلك مناسبًا)
- الأرقام المميزة

الحساب:

داخليًا، ويتم تحويلها إلى دقائق osecondsيتم حساب استخدام الصوت حسب مدة المكالمة بالنان للعرض.

رصيد الرسائل القصيرة

يتتبع استخدام الرسائل النصية.

الوحدات: رسائل (msgs)

تنسيق العرض:

> التقدم: 45 / 250 (18%)

أنواع الرسائل:

- SMS القياسية (160 حرفًا)
- SMS الطويلة (عدة مقاطع)
- MMS (إذا تم تتبعها بشكل منفصل)

رصيد النقد

يتتبع الائتمان المدفوع مسبقًا أو رصيد الحساب.

الوحدات: عملة (£، $، €، إلخ.)

تنسيق العرض:

> التقدم: £15.50 / £20.00 (77%)

الاستخدام:

- تستخدم الحسابات المدفوعة مسبقًا الرصيد النقدي لدفع تكاليف الاستخدام
- يتم خصم الائتمان مع استخدام العميل للخدمات
- يمكن إعادة شحنها عبر الدفع أو القسائم
- قد تنتهي صلاحيتها إذا لم يتم استخدامها خلال فترة الصلاحية

تحليل الرصيد المفصل

انقر على **"المزيد من المعلومات"** على أي بطاقة أو انقر على شريط التقدم لتوسيع التحليل المفصل.

عرض موسع:

> 🔵 الامتياز الشهري 20GB
> تنتهي: 25 يناير 2025 (15 يومًا) الوزن: 10 GB المستخدم: 7.5 GB المتبقي: 12.5
>
> 🔵 بيانات إضافية 5GB
> تنتهي: 31 يناير 2025 (21 يومًا) الوزن: 20 GB المستخدم: 0 GB المتبقي: 5.0
>
> 🔵 بيانات إعادة الشحن 3GB
> تنتهي: 18 يناير 2025 (منتهية) الوزن: 30 GB المستخدم: 3.0 GB المتبقي: 0
>
> 17.5 GB إجمالي المتبقي:

حقول دلو الرصيد:

- **الاسم** - معرف لدلو الر��يد/**ID**
- **المتبقي** - المبلغ المتبقي في هذا الدلو المحدد
- **المستخدم** - المبلغ المستهلك من هذا الدلو
- **تاريخ انتهاء الصلاحية** - متى تنتهي صلاحية هذا الرصيد
- **الوزن** - ترتيب الأولوية (يتم استهلاك الوزن الأعلى أولاً)

نظام الوزن

تمتلك الأرصدة قيمة **وزن** تحدد ترتيب الاستهلاك:

- **وزن أعلى = يُستهلك أولاً**
- **وزن أقل = يُستهلك آخرًا**

أمثلة على الأوزان:

- بيانات إعادة الشحن: الوزن 30 (يتم استهلاكها أولاً، انتهاء صلاحية أقصر)
- بيانات إضافية: الوزن 20 (يتم استهلاكها ثانيًا)
- الامتياز الشهري: الوزن 10 (يتم استهلاكها آخرًا، انتهاء صلاحية أطول)

يضمن ذلك استخدام الأرصدة التي تنتهي صلاحيتها قبل الأرصدة التي تدوم لفترة أطول.

للحصول على تفاصيل كاملة حول قواعد استهلاك الرصيد، واستراتيجيات الوزن، وكيفية تحديد أولويات الأرصدة بواسطة CGRateS، انظر إجراءات CGRateS وسلوكيات إعادة الشحن.

التحديثات في الوقت الحقيقي

تتجدد بيانات الاستخدام تلقائيًا كل **3 ثوانٍ** عبر الاستطلاع.

ما يتم تحديثه:

- المبالغ الحالية للرصيد
- أشرطة تقدم الاستخدام
- مؤقتات انتهاء الصلاحية
- تفاصيل دلو فردية

تجربة المستخدم:

- لا حاجة لإعادة تحميل الصفحة
- تحديثات سلسة دون وميض
- تحميل تغطية أثناء التحديث
- شارة الحالة تظهر حالة الخدمة الحالية

حالات الاستخدام:

- مراقبة استخدام العميل أثناء المكالمة
- مشاهدة انخفاض الرصيد في الوقت الحقيقي مع استخدام العميل للخدمة
- التحقق من إعادة الشحن فور الشراء

الاستخدام في أنواع الخدمات المختلفة

خدمات الهاتف المحمول

تعرض جميع أنواع الرصيد الأربعة:

- بيانات (GB)

- صوت (دقائق)
- رسائل قصيرة (رسائل)
- نقدية (عملة)

مثال:

> البيانات: 12.5GB المتبقية الصوت: 125 min المتبقية الرسائل القصيرة: 45 msgs المتبقية
> النقدية: £15.50 المتبقية

اللاسلكي الثابت / الإنترنت

عادةً ما يظهر فقط:

- بيانات (GB أو TB)
- نقدية (إذا كانت مدفوعة مسبقًا)

مثال:

> البيانات: 450GB / 500GB (ائتمان مدفوع مسبقًا) £45.00 المتبقية النقدية:

خدمات النقاط الساخنة / الدونجل

يظهر تتبع البيانات المحددة للدونجل:

- بيانات (مستهلكة مقابل مدفوعة مسبقًا)
- نقدية (رصيد مدفوع مسبقًا)

وضع العرض:

يخفي المكون الصوت والرسائل القصيرة، ويظهر فقط البيانات ذات `dongle=true` عندما يكون الصلة والأرصدة النقدية.

استكشاف الأخطاء وإصلاحها

الاستخدام يظهر كـ 0 / 0

- **السبب:** الخدمة غير متكاملة مع CGRateS أو OCS
- **الإصلاح:**

- تحقق من أن الخدمة مُعَدّة في OCS
- تحقق من اتصال واجهة برمجة التطبيقات OCS
- مراجعة خريطة UUID الخدمة

الاستخدام لا يتحدث

- **السبب:** توقف الاستطلاع أو OCS غير متاح
- **الإصلاح:**
  - تحديث الصفحة
  - تحقق من وحدة التحكم في المتصفح للأخطاء
  - تحقق من أن واجهة برمجة التطبيقات OCS متصلة

الأرصدة تظهر مبالغ غير صحيحة

- **السبب:** عدم تطابق بيانات أو مشكلة في التخزين المؤقت OCS
- **الإصلاح:**
  - فرض تحديث رصيد OCS
  - تحقق من المعاملات المعلقة
  - تحقق من تكوين OCS

تاريخ انتهاء الصلاحية مفقود

- **السبب:** الرصيد ليس له تاريخ انتهاء محدد
- **الإصلاح:**
  - بعض الأرصدة مُعَدّة لتبقى صالحة إلى الأبد (صلاحية غير محدودة)
  - تحقق من تكوين الرصيد في OCS

الأرصدة المتعددة مربكة

- **السبب:** إضافة إعادة شحن متعددة أو بيانات إضافية
- **الإصلاح:**
  - استخدم عرض التحليل المفصل لرؤية جميع الدلاء
  - فرز حسب الوزن لرؤية ترتيب الاستهلاك
  - مراجعة تواريخ انتهاء الصلاحية الفردية

التكامل مع OCS/CGRateS

تأتي بياناتا من **OCS (نظام الشحن عبر الإنترنت)**، عادةً CGRateS. للحصول على الاستخدام من معلومات شاملة حول كيفية عمل الأرصدة، بما في ذلك أنواع الأرصدة، وقواعد الاستهلاك، وسلوكيات التمديد، وتكوينات الإجراءات، انظر إجراءات وسلوكيات إعادة الشحن CGRateS.

تدفق البيانات:

1. يفتح المستخدم عرض الاستخدام
2. يستدعي OmniCRM `GET /crm/service/{service_id}`
3. تستعلم واجهة برمجة التطبيقات OCS عبر UUID الخدمة
4. تعيد OCS خريطة الرصيد:

```
{
  "BalanceMap": {
    "*data": [
      {
        "ID": "monthly_data_20GB",
        "Value": 13421772800,
        "ExpiryTime": "2025-01-25T23:59:59Z",
        "Weight": 10
      }
    ],
    "*voice": [
      {
        "ID": "monthly_voice_500min",
        "Value": 7500000000000,
        "ExpiryTime": "2025-01-25T23:59:59Z",
        "Weight": 10
      }
    ],
    "*sms": [
      {
        "ID": "monthly_sms_250",
        "Value": 250,
        "ExpiryTime": "2025-01-25T23:59:59Z",
        "Weight": 10
      }
    ],
    "*monetary": [
      {
        "ID": "prepaid_credit",
        "Value": 1550,
        "ExpiryTime": "2025-02-25T23:59:59Z",
        "Weight": 10
      }
    ]
  }
}
```


5. تقوم واجهة المستخدم بتحويل القيم إلى وحدات العرض (بايت → GB، → نانوثانية → دقائق)
6. يتم عرض أشرطة التقدم والبطاقات
7. يستمر الاستطلاع كل 3 ثوانٍ

تعيين أنواع رصيد OCS:

تقوم OCS بإرجاع بيانات الرصيد مع بادئات الأنواع التي تتطابق مع عرض واجهة المستخدم:

- بطاقة **البيانات** (استخدام الإنترنت) → `*data`
- بطاقة **الصوت** (دقائق المكالمات) → `*voice`
- بطاقة **الرسائل القصيرة** (الرسائل النصية) → `*sms`
- بطاقة **النقد** (ائتمان مدفوع مسبقًا) → `*monetary`

يمكن أن تحتوي كل نوع من أنواع الرصيد على عدة دلاء (على سبيل المثال، الامتياز الشهري + البيانات الإضافية + بيانات إعادة الشحن)، جميعها معروضة في عرض التحليل المفصل.

تحويلات قيمة الرصيد:

- **البيانات:** بايت → GB (قسم على 1024^3)
- **الصوت:** نانوثانية → دقائق (قسم على 60×10^9)
- **الرسائل القصيرة:** عدد (لا تحويل)
- **النقد:** سنتات → عملة (قسم على 100)

التجديد التلقائي وخطط الإجراءات

تحتوي الخدمات التي تم تمكين التجديد التلقائي لها على **خطط إجراءات** مجدولة في OCS. للحصول على معلومات مفصلة حول كيفية عمل الإجراءات، وتلاعب الرصيد، وتكوين خطط الإجراءات، انظر إجراءات وسلوكيات إعادة الشحن CGRateS.

ما هي خطط الإجراءات؟

خطط الإجراءات هي مهام مجدولة في CGRateS يتم تنفيذها تلقائيًا في أوقات محددة:

- إضافة رصيد إلى حساب (إعادة شحن تلقائية)
- تجديد الامتيازات الشهرية
- تطبيق رسوم متكررة
- انتهاء صلاحية الأرصدة القديمة

كيف يعمل التجديد التلقائي:

1. **توفير الخدمة:**

- عند إنشاء الخدمة مع `auto_renew = true`
- تقوم خطة توفير الخدمة بإنشاء خطة إجراء في OCS
- تم تكوين خطة الإجرا�ء للتنفيذ شهريًا (أو حسب دورة الفوترة)

2. **تكوين خطة الإجراء:**

   تحتوي خطة الإجراء على:

   - **معرف الحساب** - UUID الخدمة
   - **الإجراءات** - ما يجب القيام به (إضافة بيانات، صوت، رسائل قصيرة، رصيد نقدي)
   - **الجدول الزمني** - متى يتم التنفيذ (على سبيل المثال، شهريًا في اليوم الأول في 00:00 UTC)
   - **المبلغ** - مقدار الرصيد الذي يجب إضافته

3. **التنفيذ التلقائي:**

   - تنفذ OCS خطة الإجراء في الوقت المحدد
   - تضيف رصيدًا إلى الحساب (على سبيل المثال، 20 GB بيانات، 500 دقيقة صوت)
   - تحدد تاريخ انتهاء صلاحية الرصيد الجديد (على سبيل المثال، 30 يومًا)
   - يتم تحميل العميل عبر طريقة الدفع المسجلة

4. **عرض خطط الإجراءات:**

   - انتقل OCS إلى تفاصيل الخدمة في عرض
   - تُدرج خطط الإجراءات مع وقت التنفيذ التالي
   - تعرض: اسم الخطة، تاريخ التشغيل التالي، تفاصيل الإجراء

مثال على خ��ة إجراء:

```
{
  "ActionPlanId":
"ProductID_MonthlyPlan__ProductName_20GB_Mobile__ActionPlan_Monthly_F
  "NextExecTime": "2025-02-01T00:00:00+00:00",
  "ActionName_hr": "تجديد شهري",
  "PlanName": "20GB Mobile",
  "ActionFrequency_hr": "كل خطة شهرية",
  "custom_NextExecTime_hr": "في 22 يومًا"
}
```

إدارة التجديد التلقائي:

- **تمكين** - يتم تعيينه أثناء إنشاء الخدمة أو تعديلها
- **تعطيل** - إزالة خطة الإجراء من OCS (تحتفظ الخدمة بالرصيد الحالي ولكن لن تتجدد تلقائيًا)
- **تعديل** - تغيير مبلغ التجديد أو التكرار عبر تعديل الخدمة

التجديد اليدوي:

إذا تم تعطيل التجديد التلقائي، يجب على العميل يدويًا:

- إعادة الشحن قبل انتهاء صلاحية الرصيد
- أو يتم تعليق الخدمة عندما ينفد الرصيد

عرض في واجهة المستخدم:

تظهر علامة تبويب الخدمات حالة التجديد التلقائي:

> التجديد التالي: 1 فبراير 2025 (في 22 يومًا) مبلغ التجديد: £15.00

أفضل الممارسات

للم��ظفين الدعم:

- تحقق من الاستخدام قبل الرد على مكالمات "لماذا خدمتي بطيئة؟"
- تحقق من الرصيد بعد إعادة الشحن لتأكيد النجاح
- استخدم التحليل المفصل لتحديد الدلاء المنتهية
- راقب العملاء ذوي الاستخدام العالي لتجنب التجاوزات

للعملاء (العناية الذاتية):

- تحقق من الاستخدام بانتظام لتجنب نفاد الرصيد
- أعد الشحن قبل انتهاء صلاحية الرصيد
- افهم نظام الوزن لمعرفة أي رصيد يتم استهلاكه أولاً
- اتصل بالدعم إذا بدا أن الاستخدام غير صحيح

للمسؤولين:

- تكوين انتهاء صلاحية الرصيد المناسب
- تعيين قيم الوزن لإعطاء الأولوية للأرصدة المنتهية
- مراقبة اتصال OCS للتقارير الدقيقة
- مراجعة تكوينات الرصيد لتتناسب مع عروض المنتجات

الوثائق ذات الصلة

- إجراءات CGRateS وسلوكيات إعادة الشحن - دليل كامل لأنواع الأرصدة، وقواعد الاستهلاك، والأوزان، وسلوكيات التمديد، وتكوينات الإجراءات
- وجهات CGRateS - تكوين الوجهات للقيود الجغرافية و PLMN
- `features_topup_recharge` - نظام إعادة الشحن لإضافة الرصيد
- `csa_service_management` - إدارة الخدمات
- `csa_modify` - إضافة إضافات لزيادة الامتيازات
- `concepts_products_and_services` - تكوين المنتج

خدمة العملاء

انتحال الهوية للمساعدة وحل المشكلات

يسمح انتحال الهوية للموظفين المصرح لهم بتسجيل الدخول مؤقتًا كـ مستخدم آخر. يمكن للموظفين حل المشكلات، والتحقق من التكوينات، أو رؤية بالضبط ما يختبره المستخدم. هذه الميزة ضرورية لدعم العملاء. تتطلب الأذونات المناسبة وتتم مراقبتها بالكامل.

عند انتحال هوية عميل، يمكنك الوصول إلى بوابة الرعاية الذاتية تمامًا كما يراها العميل. هذا يوفر تشخيصًا ودعمًا دقيقًا.

لتكوين الأذونات، العملاء لإدارة العملاء، بوابة الرعاية الذاتية لميزات بوابة RBAC: انظر أيضًا العملاء.

الهدف

يوفر انتحال الهوية:

1. **حل المشكلات**: رؤية بالضبط ما يراه العميل لتشخيص المشكلات
2. **التحقق**: تأكيد أن تكوينات الخدمة والأذونات تعمل بشكل صحيح
3. **التدريب**: توضيح الميزات من منظور العميل
4. **الدعم**: مساعدة العملاء في التنقل في النظام دون مشاركة الشاشة
5. **سجل التدقيق**: يقوم النظام بتسجيل جميع جلسات انتحال الهوية لأغراض الأمان والامتثال

الأذونات المطلوبة

للانتحال الهوية، يجب أن يكون لديك واحدة من الأذونات التالية:

- `can_impersonate`: إذن انتحال مخصص لموظفي الدعم
- `admin`: وصول إداري كامل (يشمل حقوق الانتحال)

لا يمكن للمستخدمين الذين ليس لديهم هذه الأذونات الوصول إلى ميزة الانتحال.

كيفية انتحال هوية مستخدم

عبر واجهة الويب:

1. **انتقل إلى العميل**: ابحث عن العميل في نظام إدارة علاقات العملاء
2. **اختر جهات الاتصال**: عرض قائمة جهات الاتصال الخاصة بالعميل
3. **انقر على "تسجيل الدخول كمستخدم"**: يظهر الزر بجوا كل جهة اتصال لديها حساب مستخدم
4. **تأكيد الانتحال**: قد يطلب النظام تأكيدًا
5. **تبدأ الجلسة**: أنت الآن مسجل الدخول كـ ذلك المستخدم

عبر API:

بدء جلسة انتحال:

نقطة النهاية: `POST /auth/impersonate`

الأذن المطلوبة: `can_impersonate` أو `admin`

الطلب:

```
{
  "user_id": 42
}
```

الاستجابة:

```
{
  "success": true,
  "impersonating_user_id": 1,
  "target_user_id": 42,
  "impersonation_start": "2025-01-04T15:30:00Z",
  "access_token": "new_token_for_impersonated_user",
  "refresh_token": "new_refresh_token"
}
```

التوكنات المعادة هي لجلسة المستخدم الذي تم انتحاله.

ماذا يحدث أثناء انتحال الهوية

عند انتحال هوية مستخدم:

- **السياق الكامل**: ترى النظام تمامًا كما يراه المستخدم المستهدف
  - لوحة التحكم والتنقل الخاصة بهم
  - بيانات العميل الخاصة بهم (إذا كان مستخدمًا للعميل)
  - أذوناتهم وقيود الوصول
  - خدماتهم، فواتيرهم، واستخدامهم
- **تتبع الجلسة**: يتتبع النظام كلا الهوية:
  - `impersonating_user_id`: معرف المستخدم الحقيقي الخاص بك
  - `target_user_id`: المستخدم الذي تقوم بانتحاله
  - `impersonation_start`: متى بدأ الانتحال
- **مؤشر**: تعرض واجهة المستخدم لافتة تظهر أنك تقوم بانتحال شخص ما:
  - "أنت مسجل الدخول حاليًا كـ [اسم المستخدم]"
  - "انقر هنا لإيقاف الانتحال"
- **تسجيل التدقيق**: يقوم النظام بتسجيل جميع الإجراءات مع كلا معرفي المستخدم:
  - تظهر الإجراءات كما لو كانت قد تم تنفيذها بواسطة المستخدم المستهدف
  - تسجل سجلات التدقيق من كان يقوم بتنفيذها فعليًا (المستخدم المنتحل)
  - يتم الحفاظ على سجل تدقيق كامل في جدول `ImpersonationLog`

إيقاف الانتحال

عبر واجهة الويب:

1. **انقر على اللافتة**: انقر على لافتة الانتحال في أعلى الصفحة
2. **أو انتقل**: اذهب إلى قائمة المست��دم واختر "إيقاف الانتحال"
3. **التأكيد**: تنتهي الجلسة وتعود إلى حسابك الخاص

عبر API:

نقطة النهاية: `POST /auth/stop_impersonation`

الطلب: لا حاجة لجسم (طلب مصادق عليه)

الاستجابة:

```
{
  "success": true,
  "impersonation_end": "2025-01-04T15:45:00Z",
  "duration_seconds": 900,
  "access_token": "your_original_token",
  "refresh_token": "your_original_refresh_token"
}
```

يتم استعادة جلستك الأصلية.

تسجيل تدقيق الانتحال

تُسجل جميع جلسات الانتحال في جدول `ImpersonationLog` مع:

- **impersonating_user_id**: عضو الفريق الذي قام بانتحال الهوية
- **target_user_id**: العميل أو المستخدم الذي تم انتحاله
- **impersonation_start**: الطابع الزمني للبداية
- **impersonation_end**: الطابع الزمني للنهاية (عندما تم إيقاف الجلسة)
- **impersonation_duration**: المدة بالثواني

يوفر هذا مسؤولية كاملة عن جميع جلسات الانتحال ويسمح بـ:

- **تدقيق الأمان**: مراجعة من انتحل من ومتى
- **تقارير الامتثال**: إثبات الاستخدام الصحيح للوصول المرتفع
- **التحقيق**: تتبع الإجراءات أثناء جلسات الانتحال
- **المراقبة**: تحديد أنماط الانتحال غير العادية

عرض سجلات الانتحال:

يمكن للمسؤولين استعلام سجلات الانتحال عبر API:

`GET /auth/impersonation_logs?user_id={user_id}&start_date={date}&end_date={date}`

تصفية حسب:

- المستخدم المنتحل (من قام بانتحال الهوية)
- المستخدم المستهدف (من تم انتحاله)
- نطاق التاريخ
- المدة

أفضل الممارسات

1. **تقليل المدة**: انتحل الهوية فقط طالما كان ذلك ضروريًا لحل المشكلة
2. **توثيق الهدف**: لاحظ لماذا كانت هناك حاجة للانتحال في سجل نشاط العميل
3. **إبلاغ العملاء**: دع العملاء يعرفون أنك قد تحتاج إلى عرض حسابهم (سياسة الخصوصية)
4. **التحقق من الهوية**: تأكد من هوية المستخدم قبل انتحال حسابهم
5. **مراجعة السجلات**: قم بتدقيق سجلات الانتحال بانتظام بحثًا عن أنماط غير عادية
6. **تقييد الأذونات**: امنح `can_impersonate` فقط لموظفي الدعم الذين يحتاجون إليها
7. **التدريب**: تأكد من فهم الموظفين للمسؤولية وآثار التدقيق

اعتبارات الأمان

- **الوصول الكامل**: يمنح الانتحال وصولًا كاملاً إلى حساب المستخدم المستهدف
- **لا حاجة لكلمة مرور**: يتجاوز الانتحال عملية المصادقة (يعتمد فقط على الأذونات)

- **الإجراءات المسجلة**: تنسب سجلات التطبيق جميع الإجراءات أثناء الانتحال إلى المستخدم المستهدف (لكن سجلات التدقيق تظهر الفاعل الحقيقي)
- **عزل الجلسة**: ينشئ الانتحال جلسة جديدة. لا يؤثر على جلسات المستخدم المستهدف النشطة
- **حدود الوقت**: يجب أن تكون جلسات الانتحال محدودة زمنياً (قابلة للتكوين)
- **تجاوز MFA**: يتجاوز الانتحال متطلبات 2FA (يستخدم مصادقة المنتحل)

القيود

- **لا يمكن انتحال هوية المسؤولين**: اعتمادًا على التكوين، قد لا تتمكن من انتحال هوية مسؤولين آخرين
- **نفس الأذونات**: تحصل على أذونات المستخدم المستهدف، وليس اتحاد أذونات كلا المستخدمين
- **حدود الجلسة**: جلسة انتحال واحدة فقط لكل عضو في الفريق في وقت واحد
- **متطلبات التدقيق**: لا يمكنك تعطيل أو إخفاء تسجيل الانتحال

حل المشكلات الشائعة

المشكلة: لا يظهر زر "تسجيل الدخول كمستخدم"

- الحل: تحقق من أن لديك إذن `admin` أو `can_impersonate`
- الحل: تأكد من أن جهة الاتصال لديها حساب مستخدم مرتبط

المشكلة: فشل الانتحال مع خطأ في الإذن

- الحل: تحقق مما إذا كان المستخدم المستهدف مسؤولاً (قد يكون مقيدًا)
- الحل: تحقق من أن إذن الانتحال الخاص بك نشط

المشكلة: لا يمكن إيقاف الانتحال

- الحل: استخدم نقطة النهاية API لإيقاف الانتحال: `POST /auth/stop_impersonation`
- الحل: امسح ملفات تعريف الارتباط في المتصفح وسجل الدخول مرة أخرى باستخدام بيانات اعتمادك

المشكلة: الإجراءات لا تسجل بشكل صحيح

- الحل: تحقق من أن جلسة الانتحال نشطة (تحقق من اللافتة)
- الحل: راجع سجلات التدقيق - يتم تسجيل الإجراءات مع كلا معرفي المستخدم

تعريف المنتجات وتكوين CGRateS

تشرح هذه الدليل كيفية تعريف المنتجات في OmniCRM مع تكوين الفوترة المرتبط بها في CGRateS.

عادةً ما يتم تعريف المنتجات وإجراءات CGRateS الخاصة بها معًا في نصوص بايثون التي تقوم بتكوين كل من CRM و OCS (نظام الشحن عبر الإنترنت).

نظرة عامة

يتطلب تعريف منتج كامل تكوينًا في **نظامين**:

1. **CGRateS/OCS** - تعريف سلوك الفوترة (الإجراءات، الأسعار، الوجهات)
2. **OmniCRM** - تعريف بيانات المنتج (السعر، الاسم، الميزات)

تم ربط هذه الأنظمة بشكل غير محكم عن عمد، مما يسمح لك بالتفاعل مع CGRateS عبر إجراءات Ansible التي تقوم بتعريفها.

مهم: قدرات CGRateS

CGRateS هو محرك تقييم قوي ومرن للغاية. تعرض هذه الدليل **نمط تكوين شائع واحد** لمنتجات OmniCRM، لكن CGRateS يمكنه القيام ب**لكثير من الأمور الأخرى**:

- **سيناريوهات تقييم متقدمة**: أسعار حسب الوقت، فئات العملاء، توجيه بناءً على جودة المكالمات، فوترة مفاجئة
- **منطق تجوال معقد**/SMS الصادرة مقابل الواردة عند التجوال: أسعار مختلفة لمكالمات التجوال، تسعير محدد بالشبكة
- **دعم البروتوكولات القديمة**: CAMEL 2 لتجوال G/3G (SMS الصادر والوارد، USSD)، Diameter لـ LTE/5G
- **شحن متطور**: شحن قائم على الحجز، التحكم في الائتمان، تراجع متعدد المستويات
- **توجيه ديناميكي**: توجيه بناءً على التكلفة، اختيار المورد (أقل تكلفة توجيه) LCR،

هذا هو دليل CRM فقط - يركز على أنماط تعريف المنتجات البسيطة التي تعمل بشكل جيد مع OmniCRM. للحصول على تكوينات CGRateS المتقدمة، راجع وثائق CGRateS أو راجع إجراءات CGRateS للحصول على طرق التوازن واستراتيجيات التقييم وسلوكيات الشحن.

سير عمل تعريف ال��نتج الكامل

الخطوة 1: المصادقة على واجهة برمجة تطبيقات CRM

```python
import requests
import json

crm_url = 'https://crm.example.com/'
session = requests.Session()

headers = {
    "Content-Type": "application/json"
}

# الحصول على رمز المصادقة
response = session.post(crm_url + '/crm/auth/login', json={
    "email": "admin@example.com",
    "password": "your_password"
}, headers=headers)

assert response.status_code == 200
headers['Authorization'] = 'Bearer ' + response.json()['token']
print("تمت المصادقة على CRM بنجاح")
```

الخطوة 2: تعريف قوالب المخزون (اختياري)

تعرف قوالب المخزون أنواع العناصر المادية (أجهزة التوجيه، إلخ، بطاقات SIM) التي يمكن تعيينها أثناء التوفير.

```
inventory_list_new = []

# تعريف قالب مخزون بطاقة SIM
inventory_list_new.append({
    "item": "SIM Card",
    "itemtext1_label": "ICCID",
    "itemtext2_label": "IMSI",
    "wholesale_cost": 0.2,
    "retail_cost": 1,
    "allow_dropdown_staff": True,
    "allow_dropdown_customer": True,
    "icon": "fa-solid fa-sim-card"
})

# تعريف قالب مخزون رقم الهاتف المحمول
inventory_list_new.append({
    "item": "Mobile Number",
    "itemtext1_label": "E.164 Mobile Number",
    "itemtext2_label": "Type",
    "wholesale_cost": 0.2,
    "retail_cost": 3,
    "icon": "fa-solid fa-phone"
})

# تحقق مما إذا كان قالب المخزون موجودًا، قم بإنشائه أو تحديثه
for inventory_template in inventory_list_new:
    # الحصول على القوالب الموجودة
    inventory_list_existing = session.get(
        crm_url + '/crm/inventory/item_template/',
        headers=headers
    )

    # تحقق مما إذا كان القالب موجودًا بالفعل
    if inventory_template['item'] in [x['item'] for x in
inventory_list_existing.json()]:
        # تحديث القالب الموجود
        inventory_template_id = [
            x['inventory_template_id']
            for x in inventory_list_existing.json()
            if x['item'] == inventory_template['item']
        ][0]

        response = session.patch(
```

```
            crm_url + '/crm/inventory/item_template/' +
str(inventory_template_id),
            json=inventory_template,
            headers=headers
        )
        assert response.status_code == 200
        print(f"تم تحديث قالب المخزون:
{inventory_template['item']}")
    else:
        # إنشاء قالب جديد
        response = session.put(
            crm_url + '/crm/inventory/item_template',
            json=inventory_template,
            headers=headers
        )
        assert response.status_code == 200
        print(f"تم إنشاء قالب المخزون:
{inventory_template['item']}")
```

حقول قالب المخزون:

- **item** (مطلوب) - اسم القالب
- **itemtext1_label** - تسمية للحقل المخصص الأول
- **itemtext2_label** - تسمية للحقل المخصص الثاني
- **wholesale_cost** - تكلفتك لهذا العنصر
- **retail_cost** - التكلفة التي يواجهها العميل
- **allow_dropdown_staff** - عرض في قوائم الموظفين
- **allow_dropdown_customer** - عرض في قوائم العملاء
- **icon** - فئة أيقونة Font Awesome

الخطوة 3: الاتصال بـ CGRateS

```
import cgrateshttpapi
import time

OCS_Obj = cgrateshttpapi.CGRateS("ocs.example.com", "2080")

tenant = "your_tenant_name"
tpid = str(tenant) + "_" + str(int(time.time()))
```

الخطوة 4: تعريف الوجهات في CGRateS

تعرف أين يمكن استخدام الأرصدة. يجب تعريفها قبل إنشاء إجراءات تشير إليها **الوجهات**.

هناك نوعان:

1. **الوجهات الجغرافية**/SMS - بادئات الأرقام للمكالمات (مثل إلى أماكن محددة `Dest_AU_Mobile`, `Dest_International_UK`)
2. **وجهات PLMN** - رموز الشبكة لاستخدام البيانات والتجوال (مثل `Dest_PLMN_OnNet`, `Dest_PLMN_US_Verizon`)

قاعدة حرجة:

- **أرصدة المكالمات/SMS** → استخدم الوجهات الجغرافية (الرقم الذي يتم الاتصال به)
- **أرصدة البيانات** → استخدم وجهات PLMN (الشبكة التي يتصل بها العميل)

لأمثلة كاملة عن تعريف الوجهات بما في ذلك:

- الوجهات الجغرافية (محمول، خط ثابت، مجاني، دولي)
- وجهات PLMN (على الشبكة، تجوال، مناطق)
- قواعد تنسيق PLMN (`mccXXX.mncYYY`)
- متى يجب استخدام كل نهج

انظر: إجراءات CGRateS - تعريف الوجهات

الخطوة 5: تعريف إجراءات CGRateS

الإجراءات هي الآلية لإضافة الأرصدة إلى ��سابات العملاء. قبل أن تتمكن من ربط منتج CRM بإجراء، يجب تعريف الإجراء في CGRateS.

حرج: يتم تعريف الإجراءات أثناء التكوين الأولي للنظام (وليس أثناء التوفير). تحدد ما الأرصدة التي يجب إضافتها، وأوقات انتهاء الصلاحية، وسلوك التراجع، إلخ.

راجع إجراءات CGRateS للحصول على الوثائق الكاملة حول وسلوكيات الشحن:

- كيفية تعريف الإجراءات باستخدام بايثون (`OCS_Obj.SendData`)
- فهم `*topup` مقابل `*topup_reset` (سلوك التراجع)
- حسابات الوحدات (SMS بايتات للبيانات، نانوثانية للصوت، عدد لـ)
- مرجع الحقول (BalanceId، BalanceType، DestinationIDs، Units، ExpiryTime، Weight)
- أمثلة عمل كاملة (خطط متعددة الأرصدة، إضافات البيانات، إضافات الصوت)
- قواعد استهلاك الأرصدة والأولوية

مرجع سريع للتراجع:

- **`*topup`** - يضيف إلى الرصيد الحالي (يمكن التراجع). لـ: إضافات مسبقة الدفع، حزم البيانات
- **`*topup_reset`** - يستبدل الرصيد الحالي (لا يمكن التراجع). لـ: الخطط الشهرية مع مخصصات ثابتة

الخطوة 6: إنشاء المنتجات في CRM

الآن بعد أن تم تعريف الإجراءات في CGRateS، قم بإنشاء المنتجات المقابلة في CRM.

مثال: خدمة SIM مستقلة

```
product_list_new = []

product_list_new.append({
    "category": "standalone",
    "provisioning_play": "play_psim_only",
    "relies_on_list": "",
    "contract_days": 0,
    "retail_cost": 0,
    "retail_setup_cost": 0,
    "product_slug": "Mobile-SIM",
    "product_name": "Mobile SIM Only",
    "comment": "فعلية للاستخدام مع الهواتف المحمولة SIM بطاقة",
    "provisioning_json_vars": "{\"iccid\" : \"\", \"msisdn\" :
\"\"}",
    "terms": "يجب تفعيلها خلال 6 أشهر. يتم فقدان جميع الرصيد إذا لم
يتم استخدام الخدمة لمدة 12 شهرًا.",
    "service_type": "mobile",
    "residential": True,
    "business": True,
    "enabled": True,
    "inventory_items_list": "['SIM Card', 'Mobile Number']",
    "icon": "fa-solid fa-sim-card",
    "features_list": "['رقم هاتف أسترالي (61xxx)', 'أسرع السرعات
أفضل تغطية']",
    "wholesale_cost": 3,
    "wholesale_setup_cost": 1,
    "customer_can_purchase": True
})
```

مثال: إضافة خطة شهرية

```
product_list_new.append({
    "category": "addon",
    "provisioning_play": "play_topup_charge_then_action",
    "relies_on_list": "",
    "contract_days": 30,
    "retail_cost": 59.00,
    "retail_setup_cost": 0,
    "product_slug": "au-premium-plan-1",  # يرتبط بـ Action_au-premium-plan-1
    "product_name": "AU Premium Plan 1",
    "comment": "100GB دقيقة في أستراليا، 3000 بيانات في أستراليا، 3000 SMS 6 بيانات تجوال GBفي أستراليا",
    "provisioning_json_vars": "",
    "terms": "خطة ما بعد الدفع. تتجدد تلقائيًا كل 30 يومًا.",
    "service_type": "mobile",
    "residential": True,
    "business": True,
    "enabled": True,
    "icon": "fa-solid fa-mobile",
    "features_list": "['100GB دقيقة في 3000', 'بيانات في أستراليا', '3000 SMS 6', 'بيانات تجوال GBفي أستراليا']",
    "wholesale_cost": 45.00,
    "wholesale_setup_cost": 0,
    "auto_renew": "True",
    "customer_can_purchase": True
})
```

مثال: إضافة شحن لمرة واحدة

```
product_list_new.append({
    "category": "addon",
    "provisioning_play": "play_topup_charge_then_action",
    "relies_on_list": "",
    "contract_days": 0,
    "retail_cost": 30,
    "retail_setup_cost": 0,
    "product_slug": "au-data-addon-20gb",  # يرتبط بـ Action_au-data-addon-20gb
    "product_name": "AU Data Addon 20GB",
    "comment": "بيانات في أستراليا 20GB",
    "provisioning_json_vars": "",
    "terms": "شحن مسبق الدفع. رسوم فورية.",
    "service_type": "mobile",
    "residential": True,
    "business": True,
    "enabled": True,
    "icon": "fa-solid fa-mobile",
    "features_list": "['بيانات في أستراليا 20GB']",
    "wholesale_cost": 22,
    "wholesale_setup_cost": 0,
    "auto_renew": "False",
    "customer_can_purchase": True
})
```

وصف حقول المنتج:

- **category** (مطلوب) - فئة المنتج:
  - `standalone` - خدمة كاملة (تنشئ سجل خدمة جديد)
  - `addon` - مضافة إلى خدمة موجودة
  - `bundle` - حزمة من خدمات متعددة
- **provisioning_play** (مطلوب) - تاب لعب Ansible للتنفيذ (مثل `play_topup_charge_then_action`, `play_simple_service`)
- **relies_on_list** - قائمة مفصولة بفواصل من product_slugs التي يعتمد عليها هذا المنتج
- **contract_days** - مدة العقد بالأيام (0 = لا عقد، 30 = شهري)
- **retail_cost** (مطلوب) - التكلفة الشهرية/المتكررة التي يواجهها العميل
- **retail_setup_cost** - رسوم إعداد لمرة واحدة تُفرض على العميل

- **product_slug** (مطلوب) - **معرف فريد يرتبط بإجراء CGRateS** (التقليد: `Action_{product_slug}`)
- **product_name** (مطلوب) - اسم العرض
- **comment** - وصف قصير
- **provisioning_json_vars** - سلسلة JSON من المتغيرات المرسلة إلى كتاب اللعب
- **terms** - نص الشروط والأحكام
- **service_type** - تصنيف الخدمة (محمول، دونجل، ثابت، إلخ.)
- **residential** - متاح للعملاء السكنيين
- **business** - متاح للعملاء التجاريين
- **enabled** - المنتج نشط وقابل للشراء
- **inventory_items_list** - سلسلة قائمة بايثون من العناصر المطلوبة في المخزون
- **icon** - فئة أيقونة Font Awesome
- **features_list** - سلسلة قائمة بايثون من ميزات النقاط
- **wholesale_cost** - تكلفتك المتكررة
- **wholesale_setup_cost** - تكلفتك لمرة واحدة
- **auto_renew** - سلسلة "True" أو "False" للتجديد التلقائي
- **customer_can_purchase** - يمكن للعميل الشراء عبر بوابة الخدمة الذاتية

راجع أيضًا: دورة حياة المنتج

الخطوة 7: إنشاء أو تحديث المنتجات عبر واجهة برمجة

التطبيقات

```
# الحصول على المنتجات الموجودة
product_list_existing = session.get(crm_url + '/crm/product',
headers=headers)
existing_products = product_list_existing.json()

# معالجة كل منتج
for product in product_list_new:
    print(f"معالجة المنتج: {product['product_slug']}")

    # تحقق مما إذا كان المنتج موجودًا بالفعل
    if product['product_slug'] in [x['product_slug'] for x in
existing_products]:
        print(f"المنتج موجود بالفعل: {product['product_slug']}")

        # الحصول على product_id الموجود للمنتج
        product_id = [
            x['product_id']
            for x in existing_products
            if x['product_slug'] == product['product_slug']
        ][0]

        product['product_id'] = product_id

        # تحديث المنتج الموجود
        response = session.patch(
            crm_url + '/crm/product/' + str(product_id),
            json=product,
            headers=headers
        )
        print(f"الحالة: {response.status_code}")
        assert response.status_code == 200
        print(f"تم تحديث المنتج: {product['product_slug']}")
    else:
        # إنشاء منتج جديد
        print(f"إنشاء منتج جديد: {product['product_slug']}")
        response = session.put(
            crm_url + '/crm/product',
            json=product,
            headers=headers
        )
        print(f"الحالة: {response.status_code}")
```

```
        assert response.status_code == 200
        print(f"تم إنشاء المنتج: {product['product_slug']}")
```

ربط المنتجات بالإجراءات

الرابط الحرج بين منتجات CRM وإجراءات CGRateS هو **تقليد التسمية**:

```
# في CGRateS
ActionsId = "Action_au-premium-plan-1"

# في CRM
product_slug = "au-premium-plan-1"

# في كتاب اللعب (يتم إنشاؤه تلقائيًا)
cgr_action_name = "Action_" + product_slug
# النتيجة: "Action_au-premium-plan-1"
```

كيف تنفذ كتب اللعب الإجراءات

ملاحظة: استخدام `ExecuteAction` هو مجرد نمط واحد لإضافة الأرصدة. يمكنك أيضًا استدعاء `SetBalance` داخل كتاب اللعب. ومع CGRateS مباشرة أو واجهات برمجة التطبيقات الأخرى لـ `OmniCRM-API/Provisioners/plays/play_topup_charge_then_action.yaml` ذلك، يستخدم نمط ExecuteAction لأنه نهج شائع وقابل لإعادة الاستخدام يفصل منطق الرصيد (الإجراءات) عن منطق التوفير (كتب اللعب).

عندما يتم توفير منتج باستخدام هذا النمط، يقوم كتاب اللعب بإنشاء اسم الإجراء من product_slug:

```
# في play_topup_charge_then_action.yaml
- name: تعيين cgr_action_name fact
  set_fact:
    cgr_action_name: "Action_{{
api_response_product.json.product_slug }}"

# تنفيذ الإجراء
- name: تنفيذ إجراء CGRateS
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "APIerSv1.ExecuteAction",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "ActionsId": "{{ cgr_action_name }}"
        }]
      }
```

عند استخدام هذا النمط: CGRateS قبل تشغيل كتاب اللعب، يجب أن **يوجد الإجراء في** "الإجراء غير موجود" وإلا ستفشل التنفيذ مع خطأ".

راجع أيضًا: دليل كتب Ansible

مثال كامل: تعريف منتج جديد

توضح هذه القسم كيفية إنشاء منتج جديد كامل "حزمة بيانات 50GB":

1. تعريف الإجراء في CGRateS

راجع مثال إضافة بيانات بسيطة في الخطوة 5 أعلاه حول كيفية تعريف إجراء إضافة بيانات. بالنسبة لمثالنا 50GB، سيتبع الإجراء نفس النمط ولكن مع وحدات مختلفة:

- **ActionsId**: `"Action_50gb-data-pack"` (يتطابق مع product_slug أدناه)
- **BalanceType**: `"*data"`
- **DestinationIDs**: `"Dest_PLMN_OnNet"` (لاستخدام البيانات على الشبكة)

- **Units**: `50 * 1024 * 1024 * 1024` (50GB بالبايت)
- **Identifier**: `"*topup"` (يمكن التراجع)

2. إنشاء المنتج في CRM

```
product_50gb_data = {
    "category": "addon",
    "provisioning_play": "play_topup_charge_then_action",
    "relies_on_list": "",
    "contract_days": 0,
    "retail_cost": 25.00,
    "retail_setup_cost": 0,
    "product_slug": "50gb-data-pack",  # يجب أن يتطابق مع ActionsId بدون "Action_"
    "product_name": "50GB Data Pack",
    "comment": "50GB يومًا لمدة صالحة السرعة عالية بيانات من",
    "provisioning_json_vars": "",
    "terms": "تنتهي صلاحية البيانات بعد 30 يومًا. يتم ترحيل البيانات غير المستخدمة إذا تم شحنها مرة أخرى قبل انتهاء الصلاحية.",
    "service_type": "mobile",
    "residential": True,
    "business": True,
    "enabled": True,
    "icon": "fa-solid fa-database",
    "features_list": "['50GB 30 صلاحية ','بيانات عالية السرعة ','تمكين الترحيل ','يومًا']",
    "wholesale_cost": 20.00,
    "wholesale_setup_cost": 0,
    "auto_renew": "False",
    "customer_can_purchase": True
}

# الإنشاء في CRM
response = session.put(
    crm_url + '/crm/product',
    json=product_50gb_data,
    headers=headers
)
assert response.status_code == 200
print("50 :تم إنشاء المنتج gb-data-pack")
```

3. التحقق من الرابط

يمكن الآن توفير المنتج:

1. يقوم العميل بشراء "حزمة بيانات 50GB" (product_slug: `50gb-data-pack`)
2. ينفذ كتاب اللعب `play_topup_charge_then_action`
3. يقوم كتاب اللعب بإنشاء اسم الإجراء: `Action_50gb-data-pack`
4. يستدعي كتاب اللعب `APIerSv1.ExecuteAction` مع ActionsId `Action_50gb-data-pack`
5. يجد CGRateS الإجراء وينفذه
6. يتلقى العميل رصيد 50GB

راجع أيضًا: الشحن والمدفوعات من كتب اللعب

أنماط المنتجات الشائعة

تظهر هذه الأنماط تكوينات المنتجات النموذجية في CRM وكيف ترتبط بإجراءات CGRateS. للحصول على تعريفات الإجراءات المقابلة، راجع إجراءات وسلوكيات الشحن CGRateS.

النمط 1: خطة شهرية متعددة الأرصدة

حالة الاستخدام: خطة شهرية شاملة مع بيانات وصوت و SMS وتجوال

منتج CRM:

```
{
    "product_slug": "premium-monthly-plan",  # يرتبط بـ Action_premium-monthly-plan
    "product_name": "Premium Monthly Plan",
    "category": "addon",
    "retail_cost": 59.00,
    "auto_renew": "True",
    "contract_days": 30,
    "provisioning_play": "play_topup_charge_then_action",
    "features_list": "['100GB بيانات', '5000 دقيقة', '5000 SMS', 'التجوال مشمول']"
}
```

مع أنواع أرصدة متعددة. انظر المثال 1: خطة شهرية متعددة `*topup_reset` **الإجراء:** يستخدم الأرصدة (بايثون).

النمط 2: إضافة بيانات بسيطة (ترحيل)

حالة الاستخدام: شراء بيانات لمرة واحدة يتم ترحيلها إذا تم شحنها مرة أخرى

منتج CRM:

```
{
    "product_slug": "10gb-addon",  # يرتبط بـ Action_10gb-addon
    "product_name": "10GB Data Addon",
    "category": "addon",
    "retail_cost": 15.00,
    "auto_renew": "False",  # شراء لمرة واحدة
    "contract_days": 0,
    "provisioning_play": "play_topup_charge_then_action",
    "features_list": "['10GB يومًا 30 صلاحية' ,'بيانات مع ترحيل']"
}
```

الإجراء: يستخدم `*topup` لسلوك الترحيل. انظر إجراءات الترحيل - CGRateS.

النمط 3: خطة شهرية ثابتة (بدون ترحيل)

حالة الاستخدام: خطة شهرية تعيد تعيين نفس المبلغ تمامًا

منتج CRM:

```
{
    "product_slug": "fixed-50gb-monthly",  # يرتبط بـ Action_fixed-50gb-monthly
    "product_name": "Fixed 50GB Monthly",
    "category": "addon",
    "retail_cost": 35.00,
    "auto_renew": "True",
    "contract_days": 30,
    "provisioning_play": "play_topup_charge_then_action",
    "features_list": "['50GB بدون ' ,'تتجدد شهريًا ' ,'بيانات ترحيل']"
}
```

الإجراء: يستخدم `*topup_reset` إجراءات لمنع الترحيل. انظر سلوك إعادة - CGRateS الضبط.

متقدم: الشواحن وخطط التقييم

مهم: تغطي هذه القسمات التقييم الأساسي للدفع عند الاستخدام. يدعم CGRateS سيناريوهات تقييم معقدة للغاية بما في ذلك:

- **أسعار التجوال التفاضلية**: أسعار مختلفة لمكالمات الصادرة مقابل الواردة SMS/ عند التجوال
- **شحن محدد بالبروت��كول**: CAMEL 2 ـJG/3G (SMS الصادر والوارد، USSD)، Diameter 4 ـJG/5G
- **تسعير يعتمد على السياق**: أسعار تعتمد على الوقت، الموقع، شريحة العملاء، جودة الشبكة
- **تقييم متعدد الأبعاد**: يجمع بين نوع الشبكة، حالة التجوال، الوجهة، ووقت اليوم

تظهر الأمثلة أدناه تكوينات PAYG بسيطة ومبسطة مناسبة لمعظم نشرات OmniCRM.

للفوترة بالدفع عند الاستخدام (حيث تكلف الاستخدامات التي تتجاوز الأرصدة المتضمنة المال)، تحتاج إلى تعريف خطط تقييم تحدد المبالغ بالدولار التي يجب فرضها.

كيف يعمل التقييم

عندما يستخدم العميل خدمة (صوت، بيانات، SMS):

1. يتحقق CGRateS من الأرصدة الوحدوية (مثل GB للبيانات المضمنة)
2. إذا كان هناك رصيد وحدوي، يتم خصم الاستخدام منه
3. إذا لم يكن هناك رصيد وحدوي أو تم استنفاده، يتراجع CGRateS إلى الرصيد النقدي
4. **تحدد خطة التقييم** السعر (مثل 0.10 دولار لكل ميغابايت)
5. يتم خصم الت��لفة المحسوبة من رصيد العميل النقدي

مثال: إعداد رسوم البيانات بالدفع عند الاستخدام

```
# 1. تعريف سعر الوجهة (السعر)
OCS_Obj.SendData({
    "method": "ApierV2.SetTPDestinationRate",
    "params": [{
        "TPid": tpid,
        "ID": "DR_DATA_PAYG",
        "DestinationRates": [{
            "DestinationId": "Dest_PLMN_OnNet",
            "RateId": "RT_DATA_$0_10_PER_MB",
            "RoundingMethod": "*up",
            "RoundingDecimals": 4,
            "MaxCost": 0,
            "MaxCostStrategy": ""
        }]
    }]
})

# 2. تعريف السعر الفعلي
OCS_Obj.SendData({
    "method": "ApierV2.SetTPRate",
    "params": [{
        "TPid": tpid,
        "ID": "RT_DATA_$0_10_PER_MB",
        "RateSlots": [{
            "ConnectFee": 0,          # لا رسوم اتصال
            "Rate": 0.10,             # 0.10 دولار لكل وحدة
            "RateUnit": "1048576",    # 1 (1024 * 1024 بايت) ميغابايت
            "RateIncrement": "1048576", # الفوترة لكل ميغابايت
            "GroupIntervalStart": "0s"
        }]
    }]
})

# 3. تعريف خطة تقييم تستخدم هذا السعر
OCS_Obj.SendData({
    "method": "ApierV2.SetTPRatingPlan",
    "params": [{
        "TPid": tpid,
        "ID": "RP_PAYG_DATA",
        "RatingPlanBindings": [{
            "DestinationRatesId": "DR_DATA_PAYG",
```

```
            "TimingId": "*any",
            "Weight": 10
        }]
    }]
})

# 4. ربط خطة التقييم بملف تعريف التقييم (لحسابات محددة)
OCS_Obj.SendData({
    "method": "ApierV2.SetTPRatingProfile",
    "params": [{
        "TPid": tpid,
        "Tenant": tenant,
        "Category": "data",
        "Subject": "*any",  # أو معرف حساب محدد
        "RatingPlanActivations": [{
            "ActivationTime": "2024-01-01T00:00:00Z",
            "RatingPlanId": "RP_PAYG_DATA",
            "FallbackSubjects": ""
        }]
    }]
})

# 5. تحميل خطة التعريفة
OCS_Obj.SendData({
    "method": "APIerSv1.LoadTariffPlanFromStorDb",
    "params": [{
        "TPid": tpid,
        "DryRun": False,
        "Validate": True,
        "APIOpts": {}
    }]
})
```

ما يفعله هذا:

- عندما لا يكون لدى العميل رصيد بيانات (أو ينفد)، تكلف استخدام البيانات **0.10 دولار لكل ميغابايت**
- يتم خصم الرسوم من رصيدهم النقدي (`*monetary`)
- يتم احتساب الفواتير بزيادات 1 ميغابايت (تدوير للأعلى)

مثال: أسعار المكالمات الصوتية

```
# تعريف أسعار مختلفة لوجهات مختلفة
OCS_Obj.SendData({
    "method": "ApierV2.SetTPRate",
    "params": [{
        "TPid": tpid,
        "ID": "RT_VOICE_DOMESTIC_$0_30_PER_MIN",
        "RateSlots": [{
            "ConnectFee": 0.15,           # 0.15 دولار رسوم اتصال
            "Rate": 0.30,                 # 0.30 دولار لكل دقيقة
            "RateUnit": "60s",            # الفوترة لكل دقيقة
            "RateIncrement": "60s",       # التدوير إلى دقائق كاملة
            "GroupIntervalStart": "0s"
        }]
    }]
})

OCS_Obj.SendData({
    "method": "ApierV2.SetTPRate",
    "params": [{
        "TPid": tpid,
        "ID": "RT_VOICE_INTL_$1_50_PER_MIN",
        "RateSlots": [{
            "ConnectFee": 0.25,           # 0.25 دولار رسوم اتصال
            "Rate": 1.50,                 # 1.50 دولار لكل دقيقة
            "RateUnit": "60s",
            "RateIncrement": "1s",        # الفوترة لكل ثانية (أكثر دقة)
            "GroupIntervalStart": "0s"
        }]
    }]
})
```

ملفات تعريف الشحن (اختياري)

تطبق الشواحن التقييم على أنواع محددة من الأحداث. في معظم الحالات، تكون الشاحن الافتراضي كافية:

```
OCS_Obj.SendData({
    "method": "APIerSv1.SetChargerProfile",
    "params": [{
        "Tenant": tenant,
        "ID": "Charger_Default",
        "FilterIDs": [],
        "AttributeIDs": ["*constant:*req.RequestType:*none"],
        "Weight": 999
    }]
})
```

راجع أيضًا:

- وسلوكيات الشحن - طرق إدارة الأرصدة (وحدوية، نقدية، هجينة) إجراءات CGRateS وتكوين التقييم
- وجهات CGRateS - كيفية تعريف الوجهات الجغرافية وPLMN

أفضل الممارسات

1. استخدم تقاليد التسمية المتسقة

```
# جيد - علاقة واضحة
ActionsId = "Action_premium-plan-100gb"
product_slug = "premium-plan-100gb"

# سيء - لا علاقة واضحة
ActionsId = "Action_Plan_A"
product_slug = "premium_100"
```

2. تعريف ال��جراءات قبل المنتجات

قم دائمًا بإنشاء إجراءات CGRateS قبل إنشاء منتجات CRM. سيفشل كتاب اللعب إذا حاول تنفيذ إجراء غير موجود.

3. تضمين تسجيل CDR

قم دائمًا بإضافة إجراء `*cdrlog` لتتبع إضافات الأرصدة:

```
{
    "Identifier": "*cdrlog",
    "BalanceType": "*generic",
    "ExtraParameters": "
{\"Category\":\"^activation\",\"Destination\":\"اسم المنتج\"}",
    "Weight": 80
}
```

4. استخدم معرفات الأرصدة الوصفية

```
# جيد
"BalanceId": "Premium_Data_100GB"
"BalanceId": "AU_Voice_Domestic__" + str(units)

# سيء
"BalanceId": "data1"
"BalanceId": "balance"
```

5. وثق الوحدات بوضوح

```
# جيد - يظهر الحساب
Units = 100 * 1024 * 1024 * 1024  # 100GB بالبايت
Units = 3000 * 60 * 1000000000    # 3000 دقيقة بالنانوثانية

# سيء - رقم سحري
Units = 107374182400
```

6. استخدم الوزن بشكل استراتيجي

وزن أعلى = أولوية أعلى لكل من التنفيذ والاستهلاك:

```
# ترتيب التنفيذ (وزن أعلى = ينفذ أولاً)
{"Identifier": "*reset_account", "Weight": 700}
{"Identifier": "*topup_reset", "Weight": 90}
{"Identifier": "*cdrlog", "Weight": 80}

# استهلاك الرصيد (وزن أعلى = يُستهلك أولاً)
"BalanceWeight": 1200  # أرصدة مميزة/محلية
"BalanceWeight": 1100  # أرصدة تجوال
"BalanceWeight": 1000  # أرصدة دولية
```

7. اختبر بقيم صغيرة أولاً

عند تعريف منتجات جديدة، اختبر بقيم أرصدة صغيرة أولاً:

```
# اختبار
Units = 100 * 1024 * 1024  # 100MB للاختبار

# الإنتاج
Units = 100 * 1024 * 1024 * 1024  # 100GB
```

استكشاف الأخطاء وإصلاحها

خطأ: "الإجراء غير موجود"

الأعراض: يفشل كتاب اللعب مع "الإجراء غير موجود" أو "ActionsId غير موجود"

السبب: الإجراء غير معرف في CGRateS، أو عدم تطابق في التسمية

الحل:

1. استعلام: تحقق مما إذا كان الإجراء موجودًا عن الإجراء CGRateS
2. تأكد من: تحقق من التسمية `ActionsId = "Action_" + product_slug`
3. تعريف الإجراء إذا كان مفقودًا

خطأ: "الوجهة غير موجودة"

الأع��اض: يتم تنفيذ الإجراء ولكن لا يتم إنشاء الأرصدة

السبب: تشير DestinationIDs إلى وجهة غير موجودة

الحل:

1. أولاً CGRateS تعريف الوجهة في
2. استخدم `*any` للوجهات العالمية
3. تحقق من تهجئة معرفات الوجهات

تم إنشاء المنتج ولكن لم تتم إضافة أرصدة

الأعراض: يتم توفير المنتج بنجاح ولكن لا يوجد رصيد للعميل

السبب: الإجراء موجود ولكن ليس لديه عمليات إضافة رصيد

الحل:

1. تحقق مما إذا كان الإجراء يحتوي على عمليات `*topup` أو `*topup_reset`
2. تحقق من أن قيمة Units (ليست 0) صحيحة
3. تحقق من أن وقت انتهاء الصلاحية لم ينته بعد

الأرصدة لا تتجدد

الأعراض: استخدام الترحيل ولكن الرصيد غير المستخدم يختفي

السبب: استخدام `*topup_reset` بدلاً من `*topup`

الحل:

- تغيير `Identifier` من `*topup_reset` إلى `*topup`
- تأكد من أن BalanceId متسق عبر الشحنات

راجع أيضًا: إجراءات CGRateS وسلوكيات الشحن

ملخص

تظهر هذه الدليل نهجًا شائعًا واحدًا لتكوين المنتجات في OmniCRM مع CGRateS. تعمل الأنماط الموضحة هنا بشكل جيد لحالات الاستخدام النموذجية لمشغلي الشبكات الافتراضية المحمولة (MVNO) والتجوال، وحزم البيانات، مع خطط مسبقة الدفع/ما بعد الدفع.

تذكر: CGRateS قادر على سيناريوهات أكثر تعقيدًا بكثير مما تم تغطيته هنا. إذا كنت بحاجة إلى ميزات متقدمة مثل:

- تسعير MO/MT التفاضلي لمكالمات الصوت والتجوال
- شحن قائم على CAMEL 2 للأجهزة القديمة 2G/3G
- تغييرات في الأسعار حسب الوقت أو الموسم
- تسعير يعتمد على فئات العملاء (برونزي/فضي/ذهبي)
- شحن يعتمد على جودة الشبكة أو QoS
- توجيه أقل تكلفة مع عدة موردين

استشر وثائق CGRateS مباشرة واعمل على تكوين خطة التقييم بشكل مستقل عن تعريفات منتجات CRM...

الوثائق ذات الصلة

- وسلوكيات الشحن - ليل مفصل حول أنواع الإجراءات وسلوك إجراءات CGRateS الأرصدة
- دليل كتب Ansible - كيفية تنفيذ كتب اللعب للإجراءات
- الشحن والمدفوعات من كتب اللعب - معالجة المدفوعات أثناء التوفير
- دورة حياة المنتج - إدارة المنتجات خلال دورة حياتها
- المفاهيم: المنتجات والخدمات - أساسيات المنتج
- وثائق CGRateS - مرجع الرسمية CGRateS الكامل

حملات البريد الإلكتروني (EDM) والرسائل القصيرة

تتيح وحدة الحملات الجماعية في OmniCRM (EDM: البريد المباشر الإلكتروني) للمشغلين بناء شرائح الجمهور، وتصميم القوالب، وإرسال الحملات الجماعية للعملاء عبر **قناتين: البريد الإلكتروني (عبر SMTP) والرسائل القصيرة (عبر OmniMessage)**. تستهدف الحملة شريحة واحدة بقالب واحد على قناة مختارة، مع تخصيص لكل مستلم، ومعالجة إلغاء الاشتراك والتتبع. تغطي هذه الدليل كلا القناتين.

انظر أيضًا: Mailjet للتكامل مع Mailjet، Notification API لإشعارات النظام، Configuration and Customization للإعدادات العامة.

> **ملاحظة نطاق:** يصف هذا فقط مسار إرسال **الحملة الجماعية** (البريد الإلكتروني الترحيب، الفواتير، إعادة تعيين) والرسائل القصيرة). يتم تسليم البريد الإلكتروني المعاملاتي وليس Mailjet REST API بشكل منفصل عبر (تحديثات التذاكر، eSIM، كلمة المرور، تسليم له تأثير على الإعدادات أدناه.

نظرة عامة

تحتوي الوحدة على هذه اللبنات الأساسية:

المفهوم	الوصف
الشريحة	جمهور مستهدف، يتم تعريفه بواسطة معايير الفلترة (نوع الفوترة، نوع الاتصال، الحالة، الرصيد، ...) أو قائمة عناوين ثابتة.
القالب	محتوى قابل لإعادة الاستخدام لقناة: **نص تم تحميله بواسطة HTML** **قصيرة**، كلاهما مع تخصيص لكل **SMS المشغل** (البريد الإلكتروني) أو **رسالة** مستلم.
الحملة	إرسال: قالب واحد إلى شريحة واحدة على **قناة** مختارة، مع تتبع لكل مستلم.
القناة	أو **الرسائل القصيرة** (SMTP توجيه) كيفية تسليم الحملة: **البريد الإلكتروني** (OmniMessage).

القنوات

تحدد `القناة` الخاصة بالحملة ل شيء في الأسفل: وسيلة النقل، تنسيق القالب، وأي إلغاء اشتراك ينطبق:

القناة	يتم التسليم عبر	التخصيص	إلغاء الاشتراك	التفاصيل
البريد الإلكتروني	توجيه SMTP (Mailjet، Mailgun، SES، ...)	قوالب HTML Jinja2	تفضيلات فئة `email` + إلغاء الاشتراك + قائمة الحجب	تسليم البريد الإلكتروني
الرسائل القصيرة	OmniMessage	استبدال `var1`–`var4`	تفضيلات فئة `sms` + OmniMessage STOP	تسليم الرسائل القصيرة (OmniMessage)

قوالب البريد الإلكتروني والتخصيص

تخزن القوالب HTML الخام التي **تقوم بتحميلها أو لصقها** في محرر القالب. يستخدم التخصيص **صيغة Jinja2**، لذا يمكن أن تحتوي القوالب على متغيرات ومنطق بسيط:

- `{{ first_name }}`, `{{ last_name }}`, `{{ customer_name }}`, `{{ email }}`
- الشروط والحلقات، مثل `{% if first_name %}مرحبًا {{ first_name }}{% else %}مرحبًا{% endif %}`

أي حقل إضافي يتم توفيره بواسطة الشريحة لمستلم متاح أيضًا للقالب بالاسم. يتم عرض بريد كل مستلم بشكل فردي قبل الإرسال.

استخدام الحملات الجماعية

تغطي هذه الجولة بناء شريحة، وإنشاء حملة بريد إلكتروني أو SMS، وإرسالها وتتبعها، بالإضافة إلى ما يراه المستلمون. كل شيء موجود تحت قسم **التسويق** في الشريط الجانبي (**الشرائح**، **القوالب**، **الحملات**). (موجه للمشغل؛ مكتوب من التنفيذ الحالي.)

بناء شريحة

الشريحة هي الجمهور الذي تستهدفه الحملة. من **التسويق ← الشرائح** يمكنك البحث، وتحديث، وإنشاء الشرائح. تُظهر كل صف معايير الشريحة، وعدد المستلمين الحاليين ومتى تم تحديثها آخر مرة؛ انقر على صف لفتحه.

قائمة الشرائح. يتم تحديث عدد المستلمين عند الطلب؛ ��لنقر على صف يفتح الشريحة.

إنشاء شريحة (أو النقر على صف للتعديل) يفتح محرر الشريحة، حيث تحدد الجمهور من سمات العملاء: *العملاء بدون خدمات*، *العملاء برصيد صفر*، *أنواع الفوترة*، *أنواع الاتصال* و *حالة العميل*. احفظ الشريحة أولاً. **توليد الشريحة** بعد ذلك يحسب قائمة المستلمين، و **تصدير** يقوم **CSV** بتنزيلها.

محرر الشريحة: المعايير على اليسار، عدد المستلمين المباشر وتصدير CSV في الأسفل.

> الشرائح غير مرتبطة بالقناة: يمكن استخدام نفس الشريحة لحملة بريد إلكتروني أو SMS. تستهدف حملات البريد الإلكتروني جهات الاتصال التي لديها عنوان بريد إلكتروني. تستهدف حملات SMS جهات الاتصال التي لديها رقم هاتف.

تحميل قالب

ترسل حملات البريد الإلكتروني قالب HTML. قم بإنشاء أو تعديل من **التسويق ← القوالب**، مباشرة إلى الجسم (أو لصقه). تدعم القوالب HTML لتحميل ملف **.html** قالب واستخدم **تحميل** تخصيص Jinja2، بإدراج **لوحة المتغيرات المتاحة** ويقوم `{{customer_name}}`، `{{contact_firstname}}`، `{{contact_lastname}}` و `{{contact_email}}` عند المؤشر.

لا تستخدم حملات SMS قالبًا. تكتب الرسالة في السطر عند تأليف الحملة.

إنشاء حملة

من **التسويق ← الحملات ← إنشاء حملة**، يرشدك معالج قصير خلال الحملة. يقوم مفتاح **القناة** (البريد الإلكتروني / SMS) في أعلى المعالج بتحديد الخطوات والحقول التي تظهر.

1. **اختيار الشريحة**: اختر الجمهور. تُظهر الشريحة المحددة عدد مستلميها.

الخطوة 1: اختر الشريحة التي ستتلقى جهات الاتصال الحملة. مفتاح القناة موجود فوق الخطوات.

2. انقر على بطاقة HTML) **القالب / الرسالة**: لحملة **البريد الإلكتروني**، اختر قالب HTML القالب لاختياره؛ (المعروضة لحملة **SMS**، يتخطى المعالج هذه الخطوة. **معاينة** تظهر الخطوة. يظهر إشعار، وتكتب الرسالة في الخطوة التالية.

يحل إشعار محل خطوة القالب. تستمر في كتابة الرسالة، SMS بالنسبة لـ.

3. **تفاصيل الحملة**: الإعدادات الخاصة بالقناة:

   - **الفئة**: فئة الاتصال التي تنتمي إليها الحملة (محملة من `edm.communication_categories`). يستبعد النظام المستلمين الذين ألغوا الاشتراك من هذه الفئة لهذه القناة.
   - **البريد الإلكتروني**: *اسم الحملة، خط الموضوع، البريد الإلكتروني من، الاسم من*.
   - **SMS**: *اسم الحملة، معرف المرسل و رسالة SMS* (خصص باستخدام `{{ first_name }}`، `{{ last_name }}`، `{{ customer_name }}`).

تفاصيل حملة البريد الإلكتروني: حقول الموضوع والمرسل، بالإضافة إلى محدد الفئة.

تفاصيل حملة SMS: يتحول النموذج إلى معرف المرسل وجسم رسالة SMS.

4. **مراجعة وحفظ**: أكد القناة، الفئة، الشريحة والمحتوى، ثم احفظ كـ **مسودة**. **عرض المستلمين** يسرد جهات الاتصال في الشريحة (الاسم، البريد الإلكتروني أو الهاتف، العميل) قبل الالتزام.

مراجعة لحملة بريد إلكتروني.

*مراجعة لحملة SMS، وملاحظة الحفظ كمسودة مع رابط **عرض المستلمين**.*

تُحفظ الحملة دائمًا كـ **مسودة** أولاً. يمكنك إرسالها من صفحة تفاصيل الحملة (أدناه).

قائمة الحملات

التسويق ← **الحملات** تسرد جميع الحملات مع **القناة** (البريد الإلكتروني / SMS)، عدد المستلمين، ومعدلات الفتح والنقر بمجرد إرسالها. قم بالتصفية حسب **الجميع** / **مسودة** / **مرسلة**، وانقر على أي صف لفتح الحملة.

قائمة الحملات في لمحة. تظهر الحملات المرسلة SMS في عمود القناة البريد الإلكتروني عن معدلات الفتح والنقر الخاصة بها في السطر.

إرسال وتتبع حملة

انقر على حملة لفتح صفحة تفاصيلها:

- تظهر **المسودة** زر **إرسال الحملة**. يكون الزر مفعلًا بمجرد أن تحتوي الحملة على رسالة. إذا كان هناك SMS، شريحة و، بالنسبة للبريد الإلكتروني، قالب، أو بالنسبة لـ شيء مفقود، تسرد الصفحة ما هو مطلوب.
- بمجرد **الإرسال**، تظهر صفحة التفاصيل:
  - **إحصائيات الحملة**: بالنسبة لـ *البريد الإلكتروني*، تم التسليم، معدل الفتح، تم التسليم / فشل، مع زر *SMS*، معدل النقر، مفتوح/منقر/مرتجع. بالنسبة لـ **مزامنة** يسحب أحدث أعد��د التسليم من OmniMessage.

- **المستلمون**: قائمة لكل مستلم مع الحالة (`تم الإرسال`، `تم التسليم`، `مفتوح`، `منقر`، `مرتجع`، `فشل`). تظهر عمود العنوان عنوان البريد الإلكتروني (البريد الإلكتروني) أو رقم الهاتف (SMS).

*مسودة، جاهزة للإرسال. بمجرد الإرسال، تظهر هذه الصفحة SMS صفحة تفاصيل الحملة لحملة تقوم **المزامنة** بتحديث الأعداد من ،SMS إحصائيات التسليم/التفاعل وحالة كل مستلم؛ بالنسبة لـ OmniMessage.*

بمجرد الإرسال، تظهر نفس الصفحة لوحة **إحصائيات الحملة** وقائمة لكل مستلم:

حملة بريد إلكتروني مرسلة: عدد التسليم، معدل الفتح/النقر وإجمالي مفتوح/منقر/مرتجع، جنبًا إلى جنب مع قائمة لكل مستلم وحالة كل مستلم.

> نظرًا لأن التتبع غير مرتبط بمزود الخدمة، يتم اشتقاق **تم التسليم** من التفاعل. يثبت الفتح أو النقر وصول الرسالة. وبالتالي، يتم قياس معدل الفتح مقابل المستلمين الذين تم تسليمهم (المتفاعلين) بدلاً من إجمالي المرسلين.

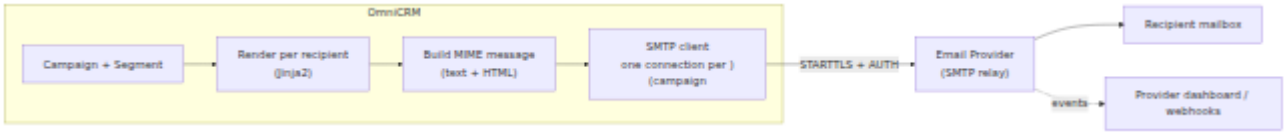
ما يراه المستلمون (إلغاء الاشتراك)

يحمل كل بريد إلكتروني رابط إلغاء الاشتراك الذي يفتح **مركز التفضيلات**: فئات البريد الإلكتروني الخاصة بالاتصال مع حالتها الحالية للاشتراك، وإجراء **حفظ التفضيلات**، وخيار **إلغاء الاشتراك من جميع البريد الإلكتروني**. يقوم إلغاء الاشتراك بنقرة واحدة من عميل البريد بإلغاء اشتراك الاتصال فقط من فئة تلك الحملة.

> تعيش ملفات الصور أعلاه في `docs/images/` (تبدأ أسماء الملفات بـ `edm-`).

تسليم البريد الإلكتروني

يتم تسليم بريد الحملة عبر **SMTP**: إلى توجيه مُعد. يجعل هذا مسار الإرسال مستقلًا عن المزود. يمكن أن تشير نفس إعدادات OmniCRM إلى Mailjet، Mailgun، Amazon SES، أو خادم SMTP الخاص بالعميل.



لكل مستلم، يقوم OmniCRM:

1. بعرض HTML القالب وأجزاء النص، والموضوع، مع متغيرات المستلم.
2. ببناء رسالة MIME `multipart/alternative` (نسخة نصية بسيطة + HTML).
3. بإرسالها عبر اتصال SMTP واحد مُعاد استخدامه للحملة بأكملها.
4. بتسجيل نتيجة لكل مستلم (`تم الإرسال` عندما يقبل التوجيه الرسالة، `فشل` عند الرفض).

تحمل كل رسالة أيضًا رأس ارتباط، في شكل `X-MJ-CustomID`، `campaign_{campaign_id}_recipient_{recipient_id}`. تعكس Mailjet هذا في أحداث الويب الخاصة بها؛ تتجاهل التوجيهات الأخرى ببساطة الرأس غير المعروف.

إعداد توجيه SMTP

يتم تكوين التوجيه تحت كتلة `mailjet.smtp` في `crm_config.yaml`. على الرغم من اسم القسم، فإن هذا هو توجيه SMTP العام المستخدم لجميع حملات EDM ويمكن أن يشير إلى أي مزود.

```yaml
mailjet:
  api_key: "your-api-key"          # يستخدم في البريد الإلكتروني المعاملاتي
الافتراضي SMTP + كاسم مستخدم
  api_secret: "your-api-secret"    # يستخدم في البريد الإلكتروني المعاملاتي
الافتراضية SMTP + ككلمة مرور
  smtp:
    host: in-v3.mailjet.com        # اسم مضيف توجيه SMTP
    port: 587                      # منفذ الإرسال
    use_tls: true                  # ترقية الاتصال باستخدام STARTTLS
    use_auth: true                 # / المصادقة باستخدام اسم المستخدم
كلمة المرور
    username: ''                   # استخدم <- فارغ mailjet api_key
    password: ''                   # استخدم <- فارغ mailjet
api_secret
    timeout: 30                    # مهلة الاتصال / الأمر بالثواني
```

المعلمات

المعلمة	النوع	مطلوب	الافتراضي	الوصف
`host`	سلسلة	نعم	`in-v3.mailjet.com`	اسم مضيف توجيه SMTP.
`port`	عدد صحيح	لا	`587`	منفذ الإرسال. استخدم `587` لـ STARTTLS، `465` لـ TLS الضمني؛ تقدم العديد من المزودين أيضًا `2525`/`25`.
`use_tls`	Boolean	لا	`true`	عندما تكون `true`، يتم ترقية الاتصال باستخدام STARTTLS بعد الاتصال. قم بتعيين `false` فقط لخادم اختبار محلي.
`use_auth`	Boolean	لا	`true`	عندما تكون `true`، يقوم OmniCRM بالمصادقة باستخدام `username`/`password`. قم بتعيين `false` فقط لخادم اختبار غير مصادق عليه.
`username`	سلسلة	لا	*(فارغ)*	اسم مستخدم SMTP. **عندما يكون فارغًا، يعود OmniCRM إلى `api_key` الخاص بـ Mailjet.** بالنسبة لأي توجيه غير Mailjet، يجب تعيين هذا بشكل صريح.
`password`	سلسلة	لا	*(فارغ)*	كلمة مرور SMTP. **عندما تكون فارغة، يعود**

المعلمة	النوع	مطلوب	الافتراضي	الوصف
				إلى OmniCRM الخاص بـ `api_secret` بالنسبة لأي Mailjet. توجيه غير Mailjet، يجب تعيين هذا بشكل صريح.
`timeout`	عدد صحيح	لا	`30`	مهلة الاتصال والأمر، بالثواني.

> **مهم:** يتم تعيين `username`/`password` الخاصة بـ API بشكل افتراضي إلى بيانات اعتماد لمجرد الراحة. عندما تشير إلى مزود مختلف، **قم دائمًا بتعيين كلاهما بشكل صريح**. خلاف ذلك، يحاول OmniCRM المصادقة على ذلك المزود باستخدام بيانات اعتماد Mailjet ويفشل.

مرجع إعداد المزود

تقدم الجداول أدناه إعدادات اتصال SMTP وآلية التتبع الأصلية لمزودين شائعين. يمكن أن تتغير أسماء المضيفين، والمنافذ، وطرق بيانات الاعتماد أو تختلف حسب المنطقة / الخطة. تأكد منها مقابل الوثائق الحالية لمزودك. ملاحظات **التتبع** هي *معلوماتية*: تصف ما يقدمه كل توجيه في وحدة التحكم الخاصة به إذا كنت تستخدمه. يرسل OmniCRM فقط SMTP ولا يكتشف تلقائيًا أو يعرض إحصائيات أصلية للمزود.

Mailjet

الإعداد	القيمة
المضيف	`in-v3.mailjet.com`
المنافذ	`587` (STARTTLS)، `465` (TLS)، `25` / `2525`
اسم المستخدم	مفتاح API (إدارة مفاتيح API ← الحساب)
كلمة المرور	مفتاح سري

افتراضي الحساب، أو لكل رسالة عبر Mailjet) **التتبع:** يتم تطبيق تتبع الفتح والنقر بواسطة رؤوس `X-Mailjet-TrackOpen` / `X-Mailjet-TrackClick`). تتوفر أحداث التفاعل في **webhooks API الإحصائيات**، وكـ **API** عبر **Mailjet لوحة معلومات**. يتضمن **API للأحداث** ؛ قم بربط الأحداث بالمستلمين باستخدام قيمةMailjet لـ webhook بالفعل معالج OmniCRM التي يرسلها `X-MJ-CustomID` OmniCRM.

Mailgun

الإعداد	القيمة
المضيف	`smtp.mailgun.org` (الولايات المتحدة) · `smtp.eu.mailgun.org` (الاتحاد الأوروبي)
المنافذ	`587` / `465` / `25` / `2525`
اسم المستخدم	`postmaster@your-sending-domain` (مخصص SMTP أو تسجيل دخول تم إنشاؤه على النطاق)
كلمة المرور	كلمة مرور بيانات اعتماد SMTP لذلك النطاق

التتبع: قم بتمكين تتبع الفتح / النقر لكل نطاق، أو لكل رسالة عبر `X-Mailgun-Track`، `X-Mailgun-Track-Opens`�` X-Mailgun-Track-Clicks`. أرفق علامة حملة باستخدام `X-Mailgun-Tag` وبيانات ارتباط مخصصة باستخدام `X-Mailgun-Variables` (JSON). تظهر **الأحداث / الإحصائيات**؛ يمكن دفع التفاعل **API**و **Mailgun** الإحصائيات في **لوحة معلومات** إلى OmniCRM عبر **webhooks Mailgun**.

Amazon SES

الإعداد	القيمة
المضيف	`email-smtp.<region>.amazonaws.com` (مثل `email-smtp.us-east-1.amazonaws.com`)
المنافذ	`587` / `2587` (STARTTLS)، `465` / `2465` (TLS)، `25`
اسم المستخدم	اسم مستخدم SES **SMTP** (تم إنشاؤه في وحدة التحكم SES)
كلمة المرور	كلمة مرور SES **SMTP** (مختلفة عن مفاتيح الوصول الخاصة بك في AWS)

التتبع: يتم تمكين تتبع الفتح والنقر من خلال **مجموعة الإعدادات**، المرفقة لكل رسالة مع رأس `X-SES-CONFIGURATION-SET`؛ أضف أبعاد الحملة باستخدام `X-SES-MESSAGE-TAGS`. يتم تسليم الأحداث (الإرسال، التسليم، الفتح، النقر، الارتداد، الشكوى) إلى وجهة حدث (SNS، Kinesis Firehose، أو EventBridge) وتظهر في **CloudWatch**. لا توجد لوحة معلومات تفاعل مدمجة، لذا خطط لوجهة حدث إذا كنت تريد إحصائيات.

SendGrid (Twilio)

الإعداد	القيمة
المضيف	`smtp.sendgrid.net`
المنافذ	`587` / `465` / `25` / `2525`
اسم المستخدم	السلسلة الحرفية `apikey`
كلمة المرور	مفتاح API لـ SendGrid مع إذن **إرسال البريد**

التتبع: `X-SMTPAPI` (JSON) يتم تعيين تتبع الفتح / النقر في **إعدادات التتبع** (الحساب) أو لكل رسالة عبر رأس مثل `filters` و`opentrack`/`clicktrack`، بالإضافة إلى `category` و`unique_args` (للارتباط). تظهر الإحصائيات في **لوحة معلومات SendGrid** و**API الأحداث** **webhook** عبر OmniCRM؛ يصل التفاعل إلى **الإحصائيات**.

Postmark

الإعداد	القيمة
المضيف	`smtp.postmarkapp.com`
المنافذ	`587` / `2525` / `25`
اسم المستخدم	**رمز API للخادم** الخاص بك
كلمة المرور	**رمز API للخادم** الخاص بك (نفس القيمة كاسم المستخدم)

التتبع: قم بتمكين الفتحات باست`X-PM-Track-Opens: true`دام والروابط باستخدام `X-PM-Track-Links: HtmlAndText`. قم بتوسيم الحملات باستخدام `X-PM-Tag` وأضف ارتباطًا باستخدام رؤوس `X-PM-Metadata-<key>`. استخدم **تدفق رسالة البث** للإرسال الجماعي / EDM. **الإحصائيات**: يصل التفاعل **API**و **Postmark** تظهر الإحصائيات في **لوحة معلومات** إلى OmniCRM عبر **webhooks**.

Mailchimp Transactional (Mandrill)

الإعداد	القيمة
المضيف	`smtp.mandrillapp.com`
المنافذ	`587` / `465` / `25`
اسم المستخدم	اسم مستخدم حساب Mailchimp/Mandrill الخاص بك (عادةً بريد الحساب الإلكتروني)
كلمة المرور	أي مفتاح API صالح لـ Mandrill

التتبع: افتراضي الحساب، أو لكل رسالة عبر `X-MC-Track` (مثل `opens,clicks_all`). قم بتوسيم باستخدام `X-MC-Tags` وأضف ارتباطًا باستخدام `X-MC-Metadata` (JSON). تظهر الإحصائيات في **لوحة معلومات** **Mandrill** وAPIإلى ؛ يصل التفاعل عبر OmniCRM **webhooks**.

> **ملاحظة:** المنتج القياسي **Mailchimp Marketing** *ليس* توجيه SMTP. فقط **Mailchimp Transactional (Mandrill)** يقدم إرسال SMTP.

Brevo (سابقًا Sendinblue)

الإعداد	القيمة
المضيف	`smtp-relay.brevo.com`
المنافذ	`587` / `465` / `25` / `2525`
اسم المستخدم	تسجيل دخول حساب Brevo (البريد الإلكتروني) الخاص بك
كلمة المرور	**مفتاح SMTP** الخاص بك (Brevo → SMTP & API → SMTP)

التتبع: تتبع الفتح / النقر على مستوى الحساب؛ الإحصائيات في **لوحة معلومات Brevo**؛ يصل التفاعل إلى OmniCRM عبر **webhooks**.

عام / مستضاف ذاتيًا SMTP

الإعداد	القيمة
المضيف	Microsoft 365 / Google Workspace (الخاص بالعميل) أو توجيه Postfix، كما هو موفر (مثل توجيه)
المنافذ	`587` عادةً (STARTTLS) أو `465` (TLS)
اسم المستخدم / كلمة المرور	كما هو موفر من قبل المشغل

التتبع: لا يوفر خادم SMTP عادي **تتبع تفاعل أصلي**. استخدم تتبع OmniCRM (أدناه) الأصلي للفتح والنقر، ومعالجة الارتداد لفشل التسليم (DSN).

تسليم الرسائل القصيرة (OmniMessage)

يمكن أن تكون الحملات أيضًا **رسائل قصيرة**، يتم تسليمها عبر خدمة OmniMessage المنفصلة. بالنسبة لـ `sms` ؛ تكون `القناة` الخاصة بالحملة `البريد الإلكتروني` (افتراضي) أو SMS، قم بتعيين `channel: sms` و `sms_message`.

المستلمون هم جهات الاتصال في الشريحة التي لديها **رقم هاتف**، يتم تصفيتها حسب فئة الحملة على قناة `sms`.

يتبع تدفق الحملة الجماعية **الإرسال** OmniMessage (يقوم OmniCRM بأداء جميع الخطوات الثلاث):

1. `POST /api/campaign_lists`: تحميل المستلمين كقائمة مسماة (`msisdn` + `var1`–`var4`).
2. `POST /api/campaigns`: إنشاء الحملة (`message`، `list_id`، `source_msisdn`، `source_smsc`، `drip_tps`).
3. `POST /api/campaigns/control` `{action: "start"}`: بدء التنقيط.

التخصيص: تدعم قوالب OmniMessage فقط `{{ var1 }}`–`{{ var4 }}` (بالإضافة إلى `{{ msisdn }}`). اكتب العناصر النائبة المسماة المألوفة ويقوم OmniCRM بتعيينها تلقائيًا: `first_name`→`var1`، `last_name`→`var2`، `customer_name`→`var3`، `email`→`var4`.

الحالة: لا تحتوي OmniMessage على **webhook** **تسليم**. استعلم باستخدام `GET /api/campaigns/:id` للحصول على `stats`. يقوم `POST /crm/email-campaign/campaigns/<id>/sms-sync` بسحب أحدث الإحصائيات إلى الحملة (`total_delivered` / `total_bounced` / `total_sent`). قم بتشغيله على جدول زمني للحصول على أرقام مباشرة.

الإعداد (`smsc` في `crm_config.yaml`):

```
smsc:
  campaigns_url: 'https://smsc:8443/api'    # قاعدة API للحملات الجماعية
  source_msisdn: 'YourBrand'                  # المرسل الافتراضي
  source_smsc: 'IMS_SMSC'                   # مسار SMSC
  drip_tps: 50                              # معدل الإرسال لكل حملة
  verify_tls: false                         # تستخدم OmniMessage
شهادة موقعة ذاتيًا
  timeout: 30
```

> أيضًا بتطبيق OmniCRM على إلغاء الاشتراك الخاص بها؛ يقوم OmniMessage تحافظ تفضيلات فئة `sms` الخاصة بالاتصال عند بناء قائمة المستلمين.

الإحصائيات والتتبع

هناك طريقتان مستقلتان لقياس الحملة. لا تتعارضان مع بعضهما البعض.





ما يقدمه كل نموذج

المقياس	مسجل عند الإرسال (كل توجيه)	في وحدة التحكم الخاصة بمزودك [1]	الأصلي OmniCRM تتبع (أي توجيه)
تم الإرسال / في الانتظار	✓	✓	✓
فشل الإرسال (مرفوض عند التوجيه)	✓	✓	✓
تم التسليم	–	✓	✓ يثبت بواسطة فتح/نقر؛ ✓ *يفترض* خلاف ذلك = (المرسل - المرتجع)
الفتحات	–	✓	✓ (بكسل التتبع)
النقرات	–	✓	✓ (إعادة توجيه الرابط)
الارتدادات	–	✓	✓ (معالجة DSN)
شكاوى البريد العشوائي	–	✓	حلقة تغذية المزود فقط
إلغاء الاشتراكات	–	✓	✓ (رابط إلغاء الاشتراك + `List-Unsubscribe`)

[1] **لا يكتشف OmniCRM أي مزود تستخدمه.** يتم تكوينه فقط مع مضيف SMTP وبيانات الاعتماد، والتي تبدو متطابقة لكل توجيه. العمود الأوسط هو ما يقدمه توجيهك، يتم عرضه في **وحدة التحكم الخاصة بذلك المزود** (ليس في OmniCRM)، ما لم تقم بتوصيل webhook الخاص به (النموذج 1) أو تستخدم تتبع OmniCRM (النموذج 2) الأصلي.

النموذج 1: تتبع مزود الخدمة الأصلي

يمكن لمعظم التوجيهات تطبيق تتبع بكسل الفتح الخاص بها وتتبع الروابط وتسجيل الأحداث. نظرًا لأن OmniCRM يحتفظ فقط بمضيف SMTP، **لا يمكنه معرفة أي مزود هو** وبيانات الاعتماد، ولا ��عرض هذه الأرقام. يمكنك قراءة هذه الأرقام عن طريق تسجيل الدخول إلى **وحدة التحكم الخاصة بذلك المزود** بنفسك. لا يحتاج هذا إلى تكوين OmniCRM. إنه ببساطة ما يقدمه توجيهك المختار.

بدلاً من ذلك، يجب على **OmniCRM** لإدخال تلك الأحداث استيعاب OmniCRM **في تحليلات webhook** لكل رسالة باستخدام حقل الارتباط الذي يحدده **OmniCRM** / يستخدمه ومطابقة كل حدث مرة أخرى إلى مستلم المزود:

المزود	يستخدمه / OmniCRM حقل الارتباط الذي يحدده
Mailjet	`X-MJ-CustomID` (معالج موجود بالفعل)
Mailgun	`X-Mailgun-Variables`
Amazon SES	`X-SES-MESSAGE-TAGS`
SendGrid	`unique_args` (عبر `X-SMTPAPI`)
Postmark	`X-PM-Metadata-*`
Mandrill	`X-MC-Metadata`

يختلف تنسيق حدث كل مزود، لذا يتطلب استيعاب أحداث مزود جديد معالج webhook خاص بالمزود.

النموذج 2: تتبع OmniCRM الأصلي (قابل للنقل)

نظرًا لأن تتبع مزود الخدمة الأصلي موجود فقط أثناء بقائك على ذلك المزود، يقوم OmniCRM بقياس التفاعل بنفسه، بشكل متطابق على **أي** توجيه. يتم **تنفيذه وتمكينه بشكل افتراضي**:

- **الفتح**: يقوم OmniCRM في 1×1 بحقن بكسل تتبع HTML. يسجل الطلب الفتح ضد المستلم.
- **النقرات**: يعيد OmniCRM كتابة الروابط لإعادة التوجيه عبر نفسه. يسجل النقر ويعيد التوجيه إلى الوجهة.

- بتحديد المستلم OmniCRM **تم التسليم**: يعد الفتح أو النقر دليلاً على التسليم. يقوم المزود. webhook **تم التسليم** عند أول ضربة بكسل / إعادة توجيه، دون الحاجة إلى يتم احتساب المستلمين الذين تم `إرسالهم` ولم يرتدوا أبدًا على أنهم *يفترض أنهم تم تسليمهم*.
- إلى (DSNs) **الارتدادات**: تستخدم الحملات مسار عائد يقوم بتوجيه إشعارات الارتداد للمعالجة. OmniCRM خادم البريد الوارد الخاص بـ.
- ورابط إلغاء الاشتراك الموجود `List-Unsubscribe` **إلغاء الاشتراكات**: يقوم رأس في الجسم بتغذية قائمة إلغاء الاشتراك الموجودة.

وتبقى بعد تغيير مزود البريد الإلكتروني OmniCRM مع هذا النموذج، تعيش جميع الإحصائيات في.

حالة الدعم الحالية

- **الأصلي على أي OmniCRM** و **تتبع** SMTP **نشط (افتراضي):** إرسال حملات **توجيه**. يوفر هذا `تم الإرسال` / `فشل` عند الإرسال، بالإضافة إلى **تم التسليم، المفتوح، المنقر، المرتجع** التي يتم اكتشافها بواسطة OmniCRM نفسه (بكسل، إعادة توجيه النقر، ارتداد VERP). لا يتطلب webhook انظر تكوين التتبع المزود.
- **اختياري:** إذا كنت تستخدم Mailjet، يمكن أن تغذي webhook الأحداث الخاصة بها مرتبط عبر) أيضًا التفاعل `X-MJ-CustomID`). لا يعتمد OmniCRM على ذلك.

تكوين التتبع

يتم تكوين تتبع OmniCRM الأصلي تحت كتلة `edm` في `crm_config.yaml`:

```
edm:
  enable_open_tracking: true      # حقن بكسل الفتح في HTML الحملة
  enable_click_tracking: true     # إعادة كتابة الروابط عبر إعادة توجيه النقر
  enable_unsubscribe: true        # إضافة رأس List-Unsubscribe + رابط
إلغاء الاشتراك في الجسم
  tracking_base_url: ''           # URL العامة التي يمكن للمستلمين الوصول
عند الفراغ crm.base_url إليها؛ يعود إلى
  tracking_secret: ''             # سر يوقع رموز التتبع؛ يعود إلى
mailjet.api_secret عند الفراغ
  bounce_domain: ''               # النطاق الذي يشير MX الخاص به إلى خادم
البريد الوارد (يمكن من ارتدادات VERP)
  bounce_prefix: 'bounce'         # بادئة الجزء المحلي لـ VERP -> bounce+
<token>@<bounce_domain>
```

المعلمة	النوع	الافتراضي	الوصف
`enable_open_tracking`	Boolean	`true`	بكسل تتبع 1×1 في جسم HTML.
`enable_click_tracking`	Boolean	`true`	إعادة كتابة الروابط المطلقة `http(s)` لإعادة التوجيه عبر OmniCRM حيث يتم تسجيل النقرات.
`enable_unsubscribe`	Boolean	`true`	إضافة رأس `List-Unsubscribe` (+ واحدة RFC 8058) ط إلغاء الاشتراك في الجسم.
`tracking_base_url`	سلسلة	*(crm.base_url)*	URL. ة لروابط التتبع. **جب أن تكون عملاء بريد المستلمين درين على الوصول يها، لذا لا تستخدم `localhost`.** ود إلى `crm.base_url` دما يكون فارغًا.
`tracking_secret`	سلسلة	*(mailjet.api_secret)*	سر المستخدم لتوقيع رموز التتبع باستخدام HMAC. يعود إلى `mailjet.api_sec` عندما يكون فارغًا.
`bounce_domain`	سلسلة	*(فارغ)*	النطاق لعناوين ارتداد VERP (`bounce+<token>@<domain>`) ص MX يجب أن يشير لى خادم البريد الوارد

المعلمة	النوع	الافتراضي	الوصف
			OmniCRM الخاص بـ م تعطيل مسار العودة .ما يكون فارغًا VERP
`bounce_prefix`	سلسلة	`bounce`	ء المحلي من عناوين ارتداد VERP.

لذا لا يحتاج التتبع ، `(campaign_id, recipient_id)` HMAC تكون الرموز موقعة باستخدام إلى عمود قاعدة بيانات إضافي ولا يمكن تزوير الروابط أو تعدادها دون السر.

> **ملاحظة حول الرموز:** الرمز هو معرفات الحملة / المستلم *الموقعة*، وليس المشفرة. يتم ترميز المعرفات باستخدام base64، لذا يمكن للمستلم الذي يقوم بفك ترميز رابط التتبع قراءة أرقام الحملة والمستلم (التسلسلية). وبالتالي، تكون الرموز محمية من العبث ولكن ليست غير شفافة. إذا كان يجب عدم كشف المعرفات، انتقل إلى رموز عشوائية لكل مستلم أو حمولة رمز مشفرة (يضيف عمودًا) `EmailCampaignRecipient` مخزنة على.

تُخدم نقاط النهاية العامة للتتبع (بدون مصادقة) بواسطة API:

نقطة النهاية	الغرض
`GET /crm/email-campaign/edm/open/<token>`	شفاف GIF يسجل فتحًا؛ يعيد بكسل 1×1.
`GET /crm/email-campaign/edm/click/<token>?u=<url>`	يسجل نقرة، ثم يعيد التوجيه 302 إلى `<url>` (http/https فقط).
`GET` أو `POST /crm/email-campaign/edm/unsubscribe/<token>`	يضيف المستلم إلى قائمة الحجب **ويسجل إلغاء الاشتراك في تفضيلات الاتصال الخاصة بالعميل (`marketing`/`email`)** (يتم احترامه في الإرسال المستقبل).

> **مهم:** يجب أن يكون `tracking_base_url` هو URL الذي يمكن للمستلمين الوصول إليه. يتطلب التقاط الارتداد أيضًا `bounce_domain` مع سجل MX يشير إلى خادم البريد الوارد ويسج�ّ الارتداد بدلاً من إنشاء VERP والذي يتعرف على مستلمي OmniCRM، الخاص بـ تذكرة.

تفضيلات الاتصال

يتم تخزين إلغاء الاشتراكات لكل اتصال في حقل `communication_preferences` على شكل JSON، بشكل `{category: {channel: opted_in}}` (بدون جدول إضافي). الفئات هي `marketing`، `newsletter`، `product_updates`، `surveys`، والقنوات هي `email`، `sms`. يستخدم نموذج إلغاء الاشتراك: يعني عدم وجود فئة / قناة أن الاتصال مشترك، لذا لا تحتاج الفئات أو القنوات الجديدة إلى تغيير في المخطط.

وتستبعد عملية اختيار `marketing`/`email`، يسجل إلغاء الاشتراك في الحملة إلغاء اشتراك `marketing`/`email` (بالإضافة إلى قائمة الحجب) مستلمي الحملة أي اتصال ألغى الاشتراك من العالمية). ملاحظة: يستخدم فلتر إلغاء الاشتراك `marketing` كفئة EDM تُعامل حملات `JSON_EXTRACT` وليس مفهرسًا، وهو أمر جيد عند أحجام الاتصال النموذجية.

فئات الاتصال

تحت `crm_config.yaml` تكون الفئات **محددة من قبل المشغل** في `edm.communication_categories` (بدون جدول)، كقائمة من `{key, name, channel, default_opt_in, active}`:

```yaml
edm:
  communication_categories:
    # أنواع رسائل التسويق عبر البريد الإلكتروني (إلغاء الاشتراك: default_opt_in true = مشترك حتى يلغوا الاشتراك)
    - {key: marketing,       name: Marketing,           channel: email, default_opt_in: true}
    - {key: newsletter,      name: Newsletter,          channel: email, default_opt_in: true}
    - {key: promos,          name: Promotions & Offers, channel: email, default_opt_in: true}
    - {key: product_updates, name: Product Updates,     channel: email, default_opt_in: true}
    - {key: surveys,         name: Surveys & Feedback,  channel: email, default_opt_in: false}
    - {key: events,          name: Events & Webinars,   channel: email, default_opt_in: false}
    # أنواع رسائل التسويق عبر SMS (الاشتراك: default_opt_in false = مستبعد حتى يشتركوا)
    - {key: marketing,       name: Marketing SMS,       channel: sms,   default_opt_in: false}
    - {key: promos,          name: Promotions SMS,      channel: sms,   default_opt_in: false}
    - {key: service_alerts,  name: Service Alerts,      channel: sms,   default_opt_in: true}
```

وتحمل **الاشتراك الافتراضي** الخاص بها، (`sms` أو `email`) كل فئة تنتمي إلى **قناة** واحدة بحيث يمكن للمشغل تشغيل، على سبيل المثال، فئة *الترويج (البريد الإلكتروني)* التي تتطلب إلغاء الاشتراك جنبًا إلى جنب مع فئة *تنبيهات SMS (sms)* التي تتطلب الاشتراك.

يتم حل الاشتراك الفعال للاتصال كالتالي: تفضل **تفضيل الاتصال الصريح**؛ بخلاف ذلك، ينطبق **`default_opt_in`** للفئة.

الغرض	نقطة النهاية
للقراءة) قائمة بفئات الاتصال فقط؛ محددة في `crm_config.yaml`).	`GET /crm/email-campaign/communication-categories`
الاشتراك المحلول لكل فئة نشطة لاتصال.	`GET /crm/email-campaign/contacts/<id>/communication-preferences`
الجسم) تعيين تفضيل واحد `{category, channel, opted_in}`).	`PUT /crm/email-campaign/contacts/<id>/communication-preferences`

تستبعد عملية اختيار .(بشكل افتراضي `marketing` تعود إلى) `category` تحدد كل **حملة فئة** المستلمين جهات الاتصال التي ألغت الاشتراك من تلك الفئة / القناة، مع تطبيق `default_opt_in` للفئات التي لا تحتوي على تفضيل صريح.

رابط **إلغاء الاشتراك هو مركز تفضيلات**: يعرض فئات البريد الإلكتروني الخاصة بالاتصال مع حالتها الحالية للاشتراك (تكون فئة الحملة الخاصة بها مميزة)، ويتيح لهم حفظ التغييرات، ويقدم *إلغاء الاشتراك من جميع البريد الإلكتروني* (إلغاء عالمي). يقوم إلغاء الاشتراك بنقرة واحدة (RFC 8058 `List-Unsubscribe-Post`) بإلغاء اشتراك الاتصال فقط من فئة تلك الحملة.

قابلية التسليم

بغض النظر عن المزود، قم بتكوين المصادقة لنطاق الإرسال الخاص بك حتى تصل الحملات إلى صندوق الوارد:

الآلية	الغرض
SPF	يصرح لخوادم المزود بإرسال البريد نيابة عن نطاقك.
DKIM	يوقع الرسائل بشكل تشفيري؛ يقدم كل مزود سجل DKIM للنشر.
DMARC	ينسق SPF / DKIM ويحدد سياسة للفشل؛ يحسن التوزيع والتقارير.
Return-Path	إلى حيث يمكن لـ (DSNs) ارتدا إشعارات توجيه بحيث يتم تعيينه معالجتها (أو المزود) OmniCRM.

الاختبار

محلي للتصحيح الذي يلتقط الرسائل بدلاً من SMTP للتطوير والاختبار، قم بتوجيه التوجيه إلى خادم تسليمها:

```
mailjet:
  smtp:
    host: 127.0.0.1
    port: 1025
    use_tls: false
    use_auth: false
```

يوفر خادم تصحيح لهذا الغرض؛ يطبع كل مستلم للرسالة، الموضوع، ومعرف الارتباط (وفي وضع وبنية الرسالة دون إرسال بريد Jinja2 التفاصيل، الجسم الكامل المعروض)، لذا يمكنك تأكيد عرض إلكتروني مباشر.

استكشاف الأخطاء وإصلاحها

الحملة عالقة أو يتم وضع علامة على المستلمين كـ `فشل` على الفور

- تم تعيينها للمزود المختار `username` / `password` وأن `host` / `port` تحقق من (عندما تكون فارغة Mailjet تعود إلى بيانات اعتماد).
- تأكد من أن عنوان المرسل هو مرسل معتمد / موثق لدى المزود.
- مكتملة (SPF / DKIM) تحقق من أن مصادقة نطاق المزود.

`{{ ... }}` يتم إرسال البريد الإلكتروني ولكن يظهر التخصيص فارغًا أو حرفيًا

- تحقق من أن اسم المتغير موجود للشريحة (مثل `first_name`).
- تحقق من أن القالب ذلك المستلم Jinja2 صالحة؛ يفشل خطأ القالب يستخدم صيغة ويتم تسجيله في رسالة الخطأ الخاصة به.

لا تظهر OmniCRM تظهر لوحة معلومات المزود الفتحات / النقرات ولكن تحليلات

- هذا متوقع ما لم يستوعب OmniCRM webhook لذلك المزود (النموذج 1) أو يتم اليوم؛ Mailjet الأصلي (النموذج 2). يتوفر استيعاب أحداث OmniCRM تمكين تتبع تتطلب مزودات أخرى معالجًا خاصًا بالمزود.

لا توجد بيانات فتح / نقر في أي مكان

- يجب تمكين تتبع الفتح / النقر لدى المزود (إعداد الحساب أو رؤوس الرسالة لكل رسالة). على توجيه عادي / مستضاف ذاتيًا، لا يوجد تتبع أصلي، لذا استخدم تتبع OmniCRM الأصلي.

إدارة مخزون eSIM

تشرح هذه الدليل كيفية تكوين مخزون eSIM (عبر الإنترنت) الرقمية لقنوات التوزيع المختلفة: مقابل المطبوعة (بطاقات فعلية).

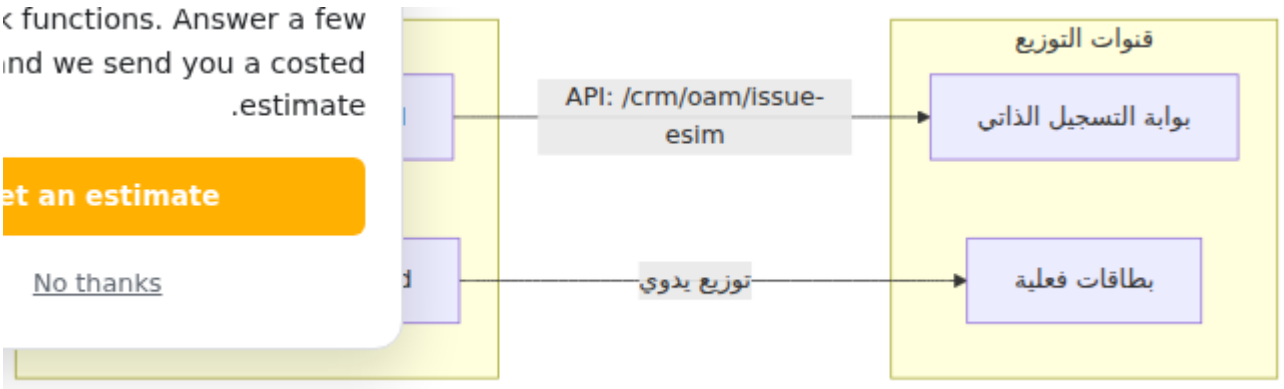
لعمليات التوفير وواجهة برمجة SIM انظر أيضًا: المخزون لإدارة المخزون العامة، توفير بطاقة التطبيقات لمصدر eSIM.

نظرة عامة

يمكن توزيع eSIMs من خلال قناتين:

1. **التوزيع الرقمي**: يتم إصدارها على الفور عبر بوابة التسجيل الذاتي، ويتم تسليمها عبر البريد الإلكتروني مع رمز QR
2. **التوزيع المطبوع**: مطبوعة مسبقًا على بطاقات كرتونية للتوزيع الفعلي أو البيع بالتجزئة

يتحكم حقل `item_location` في جدول المخزون في القناة المتاحة لـ eSIM.



تكوين قناة التوزيع

يحدد حقل `item_location` كيفية توزيع eSIM:

القيمة	الوصف	متاحة عبر
`digital`	متاحة للإصدار عبر الإنترنت / الفوري	بوابة التسجيل الذاتي، واجهة برمجة التطبيقات
`printed`	محجوزة لتوزيع بطاقات فعلية	تعيين يدوي فقط
قيم أخرى	غير متاحة للإصدار التلقائي	تعيين يدوي فقط

إعداد eSIMs الرقمية

لجعل eSIMs متاحة للتوزيع عبر الإنترنت:

1. انتقل إلى OmniCRM في **المخزون**
2. اختر عناصر مخزون eSIM المخصصة للتوزيع الرقمي
3. اضبط `item_location` على `digital`
4. احفظ التغييرات

إعداد eSIMs المطبوعة

بالنسبة لـ eSIMs التي تم طباعتها على بطاقات فعلية:

1. انتقل إلى OmniCRM في **المخزون**
2. اختر عناصر مخزون eSIM المطبوعة
3. اضبط `item_location` على `printed` (أو أي قيمة غير `digital`)
4. احفظ التغييرات

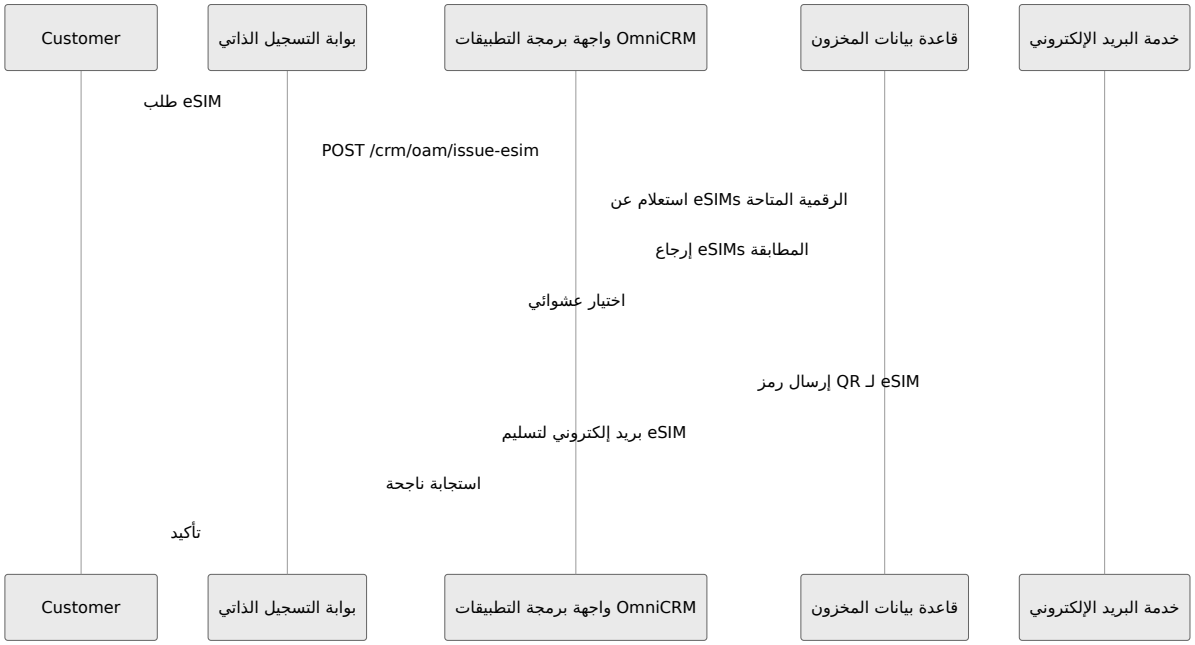
كيفية عمل الإصدار الرقمي

عندما يطلب عميل eSIM عبر بوابة التسجيل الذاتي عبر `POST /crm/oam/issue-esim`:

1. تستعلم واجهة برمجة التطبيقات عن المخزون للحصول على eSIMs المتاحة
2. يتم اعتبار eSIMs فقط التي تطابق **جميع** المعايير أدناه
3. يتم اختيار eSIM واحدة عشوائيًا من المجموعة المتاحة
4. يتم إرسال تفاصيل eSIM إلى العميل عبر البريد الإلكتروني

معايير اختيار المخزون

الغرض	الشرط	الحقل
يحدد عناصر مخزون eSIM	يحتوي على "esim" (غير حساس لحالة الأحرف)	`item`
يقتصر على قناة التوزيع الرقمية	يساوي `digital`	`item_location`
لم يتم تعيينه بعد لعميل	فارغ	`customer_id`
لم ي��م تعيينه بعد لخدمة	فارغ	`service_id`



بما في ذلك معلمات الطلب، وحدود issue-esim للحصول على تفاصيل كاملة حول نقطة النهاية المعدل، واستجابات الخطأ، انظر توفير بطاقة eSIM - واجهة برمجة التطبيقات العامة لمصدر SIM.

معالجة الأخطاء

إذا لم تكن هناك eSIMs رقمية متاحة، فإن واجهة برمجة التطبيقات تعيد:

حالة HTTP	نوع الخطأ	الرسالة
503	`NoInventory`	"لا توجد eSIMs أخرى متاحة حاليًا. يرجى المحاولة مرة أخرى لاحقًا أو اختيار SIM فعلي."

الحقول الرئيسية للمخزون لـ eSIMs

الحقل	النوع	الوصف
`item`	سلسلة	معرف نوع العنصر (يجب أن يحتوي على "esim" لعناصر eSIM)
`item_location`	سلسلة	قناة التوزيع: `digital` للإنترنت، قيم أخرى للموقع الفعلي
`itemtext1`	سلسلة	ICCID: المعرف الفريد لـ eSIM
`management_url`	سلسلة	سلسلة تفعيل LPA (مثل، `LPA:1$rsp-eu.simlessly.com$...`)
`customer_id`	عدد صحيح	العميل المعين (فارغ إذا لم يتم تعيينه)
`service_id`	عدد صحيح	الخدمة المعينة (فارغ إذا لم يتم تعيينه)

للحصول على مرجع كامل لجميع حقول عناصر المخزون، انظر المخزون - حقول عناصر المخزون.

الاستيراد التلقائي من HSS

عند استيراد SIMs من HSS باستخدام نص التوفير، يتم تعيين حقلين تلقائيًا بناءً على سجل HSS:

قناة التوزيع (`item_location`):

- يتم وضع ،(غير حساس لحالة الأحرف) **"digital"** إذا كان اسم الدفعة يحتوي على `item_location = 'digital'` للتوزيع عبر الإنترنت كـ eSIM علامة على
- إلى الموقع الفعلي الافتراضي (مثل "المكتب الرئيسي") eSIM خلاف ذلك، يتم تعيين

سلسلة تفعيل LPA (`management_url`):

- `management_url` تلقائيًا كـ HSS في SIM من سجل `lpa` يتم استيراد حقل
- eSIM لتفعيل CRM في QR يتيح ذلك إنشاء رمز
- بدون سلسلة تفعيل eSIM يتم استيراد ، `lpa` قيمة HSS إذا لم يكن لدى سجل

هذا يعني أنه يمكن التحكم في قناة التوزيع وبيانات LPA على مستوى HSS:

- قم بتسمية الدفعات بشكل مناسب (مثل "eSIMs الرقمية Q1 2026"، "دفعة بطاقات 3 مطبوعة")
- تأكد من أن eSIMs تحتوي على حقل `lpa` ممتلئ في HSS قبل الاستيراد

أفضل الممارسات

فصل المخزون

- احتفظ بمجموعات eSIM الرقمية والمطبوعة منفصلة باستخدام `item_location`
- راقب مستويات ا��مجموعة الرقمية لضمان توفرها للعملاء الذين يسجلون أنفسهم
- ضع في اعتبارك إعداد تنبيهات عندما ينخفض مخزون eSIM الرقمي

تحديثات الدفعات اليدوية

عند استيراد أو تحديث eSIMs يدويًا:

1. أثناء الاستيراد `item_location = 'digital'` اضبط **eSIMs الرقمية**:
2. أو الموقع الفعلي `item_location = 'printed'` اضبط **eSIMs المطبوعة**: (مثل "مكتب جورجتاون")

التحويل بين القنوات

لنقل eSIMs بين القنوات:

- تحديث **من المطبوعة إلى الرقمية**: `item_location` من `printed` إلى `digital`
- تحديث **من الرقمية إلى المطبوعة**: `item_location` من `digital` إلى `printed` (فقط إذا لم يتم إصدارها بعد)

استكشاف الأخطاء وإصلاحها

لا توجد eSIMs متاحة للإصدار عبر الإنترنت

الأعراض: يرى العملاء "لا توجد eSIMs متاحة حاليًا" عند طلب eSIM عبر الإنترنت.

الأسباب المحتملة:

- لا توجد eSIMs تم تعيين `item_location` لها على `digital`
- جميع eSIMs الرقمية قد تم تعيينها بالفعل (كلا من `service_id` و `customer_id` غير فارغين)
- لا تحتوي على eSIM عناصر في اسم العنصر "esim"

الحل:

1. تحقق من المخزون عن eSIMs مع `item_location = 'digital'`
2. تحقق من وجود eSIMs غير المعينة (كلا من `service_id` و `customer_id` فارغين)
3. أضف المزيد من eSIMs إلى المجموعة الرقمية أو قم بتحويل المطبوعة إذا كانت متاحة

إصدار eSIMs المطبوعة عبر الإنترنت

الأعراض: يتم إصدار eSIMs المخصصة للبطاقات الفعلية عبر البوابة.

الأسباب المحتملة:

- لم يتم تعيين `item_location` بشكل صحيح على eSIMs المطبوعة

الحل:

1. تحديث جميع eSIMs المطبوعة لتكون `item_location = 'printed'`
2. تأكد من أن استيرادات eSIMs المطبوعة الجديدة تعين القيمة الصحيحة لـ `item_location`

نظام بث الخلايا

يمكن نظام بث الخلايا في OmniCRM إرسال تنبيهات الطوارئ من مشغلي الشبكات المحمولة إلى الأجهزة المحمولة داخل مناطق جغرافية محددة. يُعتبر بث الخلايا ميزة حيوية للسلامة العامة تُستخدم في تنبيهات AMBER، وتحذيرات الطقس، وتنبيهات تسونامي، وغيرها من الإشعارات الطارئة.

الميزة الرئيسية: تنبيه الهواتف SMS على عكس رسائل القياسية، ستقوم رسائل بث الخلايا بتنبيه بشكل مسموع حتى لو كانت في وضع الصامت، أو نفدت رصيدها، أو كانت في حالة تجوال. نظرًا لأن هذه رسالة بث، فمن الممكن إرسال تنبيه إلى كل فرد من السكان الذين يحملون هاتفًا محمولًا في غضون ثوانٍ.

رابط تحذير أومنيتوتش (OWL)

يوفر نظام رابط تحذير أومنيتوتش (OWL) حلاً شاملاً لمتخصصي إد��رة الكوارث ومشغلي الشبكات المحمولة:

- **كيان بث الخلايا (CBE)** - تطبيق ويب آمن للمستخدمين المصرح لهم لإنشاء وبث رسائل التحذير الطارئة
- **مركز بث الخلايا (CBC)** - مكون تكامل الشبكة المتوافق مع المعايير الذي يتصل لتوزيع الرسائل (2G/3G/4G/5G) بالشبكات الخلوية

تم تصميم OWL للاستخدام على أي جهاز يحتوي على متصفح ويب مثل أجهزة الكمبيوتر، واللابتوبات، والأجهزة اللوحية، أو (Chrome/Firefox/Safari/Edge)، الهواتف المحمولة.

نظرة عامة

يسمح بث الخلايا (أو PWS) المعروف أيضًا بنظام التحذير العام للمشغلين بـ:

- **إرسال تنبيهات الطوارئ** - توزيع معلومات السلامة الحيوية على جميع الأجهزة في منطقة
- **استهداف المناطق الجغرافية** - البث إلى مناطق تتبع محددة أو خلايا شبكة من خلال التحكم في أي أبراج خلوية تبث الرسائل

- **دعم لغات متعددة** - تقديم رسائل التحذير بعدة لغات في وقت واحد (حتى 500 حرف ل��ل لغة)
- **إدارة دورة حياة التحذير** - إنشاء، تحديث، الموافقة، مراقبة وحذف الرسائل المذاعة
- **التكامل مع الأنظمة الخارجية** - الاتصال مع بنية CBC (مركز بث الخلايا) عبر واجهات الشبكة الخلوية المتعددة
- **المصادقة الثنائية** - عملية موافقة آمنة باستخدام كلمات مرور لمرة واحدة تعتمد على الوقت (TOTP)
- **قاعدة الشخصين** - متطلب اختياري للموافقة من شخص ثانٍ قبل نقل الرسالة

لا يتطلب بث الخلايا قوائم المشتركين ويمكنه الوصول إلى جميع الأجهزة القادرة، SMS على عكس في منطقة جغرافية على الفور، مما يجعله مثاليًا لتنبيهات السلامة العامة الحرجة من حيث الوقت. في معظم الشبكات، يستغرق البث إلى جميع الأجهزة أقل من 10 ثوانٍ.

حالات الاستخدام

يستخدم بث الخلايا لـ:

- **تحذيرات الطوارئ** - الكوارث الطبيعية (الزلازل، التسونامي، الفيضانات، الحرائق)
- **تنبيهات AMBER** - إشعارات اختطاف الأطفال
- **تحذيرات الطقس** - تحذيرات الطقس القاسي، تنبيهات الأعاصير
- **السلامة العامة** - تهديدات إرهابية، انسكابات كيميائية، إجلاءات
- **رسائل الاختبار** - اختبار النظام وحملات التوعية العامة (رسائل الاختبار الشهرية لا تنبه المستخدمين ولكن سيتم استلامها)

يكون الاستهداف الجغرافي غالبًا ذا أهمية قصوى في حالات الطوارئ. على سبيل المثال، يجب ألا تُرسل تنبيهات تسونامي التي تنصح الأشخاص القريبين من المناطق الساحلية بالبحث عن ارتفاعات أعلى إلى الأشخاص في المناطق الداخلية البعيدة عن الخطر. من خلال التحكم في أي أبراج خلوية تبث رسائل التحذير الطارئة، يمكن تحديد نطاق الإرسال ليقتصر على المنطقة الجغرافية المناسبة.

دورة حياة رسالة التحذير الطارئة

يتطلب نقل رسالة تحذير الطوارئ اتخاذ إجراءات سريعة مع توفير معلومات دقيقة ومصادقة لضمان صحة الرسالة.

تتكون دورة حياة الرسالة من أربع مراحل:

1. **تعريف الرسالة** - نوع الرسالة، محتوى الرسالة، إعدادات انتهاء الصلاحية
2. **استهداف الرسالة** - المناطق الجغرافية وأبراج الخلايا التي سيتم البث إليها
3. **موافقة الرسالة** - تفويض/تحقق من هوية المشغل وموافقة الشخص الثاني (إذا لزم الأمر)
4. **مراجعة الرسالة** - التأكيد النهائي قبل الإرسال

بعد البث، يمكن مراقبة الرسائل، وتحديثها مع تطور المواقف، وإيقافها عندما يتراجع الخطر الفوري.

المرحلة 1: تعريف الرسالة

تحدد هذه الخطوة المعلمات الأساسية لرسالة التحذير الطارئة التي سيتم بثها:

معرف الرسالة
تحتوي أنواع الرسائل المختلفة على معرفات مختلفة، والتي يتم التعامل معها بشكل مختلف من قبل الهواتف المستقبلة. على سبيل المثال، يجب ألا تنبه رسالة الاختبار الشهرية المستخدمين الفعليين ولكنهم سيستقبلون الرسالة.

ملاحظة: تحتوي كل قالب رسالة متاح في نظام OWL المناسب على معرف الرسالة (MI) بالفعل مضمن، لذا لا يحتاج مستخدمو النظام إلى تحديد ذلك يدويًا عند إنشاء رسالة تحذير.

نص الرسالة
نص محدود إلى 500 حرف يحتوي على الرسالة التي سيتم عرضها على المستخدمين النهائيين. يمكن تقديم الرسائل بـ **لغات متعددة** عن طريق إضافة نص اللغة الثانية أسفل الأولى. تذكر أن حد الـ 500 حرف ينطبق على الرسالة الكاملة بما في ذلك جميع اللغات.

قوالب الرسالة
يمكن تعريف رسائل التحذير من المخاطر مسبقًا كـ "قوالب" لسيناريوهات مختلفة متوقعة مثل:

- الفيضانات
- التسونامي
- الزلازل
- الاختبارات الدورية
- سيناريوهات الكوارث الأخرى

تساعد القوالب في توفير الوقت أثناء الطوارئ. يمكن تعديل هذه القوالب حسب الحاجة عند تعريف الرسالة، أو يمكن كتابة الرسائل من الصفر.

انتهاء صلاحية الرسالة وا��تكرارات
تمتلك الرسائل الطارئة عمرًا محدودًا تكون فيه ذات صلة. عند تعريف الرسالة:

- **تنتهي (بالدقائق)** - المدة التي ستستمر فيها الرسالة في البث
- **تكرارات الرسالة** - عدد المرات التي سيتم إعادة بثها

ستظهر الرسالة للمستخدم مرة واحدة فقط على كل هاتف. ومع ذلك، ستستمر مواقع الخلايا في نقل الرسائل حتى يتم الوصول إلى وقت انتهاء الصلاحية لضمان استلام الأشخاص الذين يدخلون منطقة التغطية من الخارج الرسالة.

المرحلة 2: الاستهداف

يتم إرسال رسائل بث الخلايا على مستوى برج الخلايا، ويمكن تحديد النطاق الجغرافي من خلال اختيار الأبراج التي تبث الرسالة.

الاستهداف الاختياري
هذه الخطوة اختيارية. عدم إدخال أي معلومات استهداف يعني أن جميع أبراج الخلايا ستنقل رسالة التحذير الطارئة.

المناطق المس��هدفة المحددة مسبقًا
يمتلك نظام OWL قاعدة بيانات لجميع أبراج الخلايا ويمكنه تحديد المناطق المستهدفة على الخريطة. يمكن استهداف المناطق باستخدام مناطق محددة مسبقًا (تم تحديدها مسبقًا للاختيار السريع) أو من خلال رسم مناطق مخصصة على الخريطة.

أدوات رسم الخريطة
يمكن إنشاء مناطق استهداف مخصصة باستخدام:

- أداة المضلع - رسم حدود تغطية دقيقة
- أداة الدائرة - تنبيهات سريعة تعتمد على نصف القطر
- أداة المستطيل - تغطية متوافقة مع الشبكة

تتيح ميزة "إضافة منطقة جديدة" تحديد مناطق استهداف مخصصة يمكن حفظها للاستخدام في المستقبل.

المرحلة 3: الموافقة

تتحقق عملية الموافقة من أن الشخص الذي يصدر رسالة التحذير الطارئة مخول للقيام بذلك.

المصادقة الثنائية
تستخدم كلمات مرور لمرة واحدة تعتمد على الوقت (TOTP) عبر:

- **رمز مادي** (مثل RSA SecurID)
- **حل قائم على التطبيق** (Google Authenticator، Authy، Microsoft Authenticator، أو تطبيقات أخرى متوافقة مع TOTP)

يجب أن يكون لدى المستخدمين الذين سيقومون بإنشاء أو الموافقة على مسودات رسائل التنبيه تطبيق مصادق على هواتفهم الذكية لتوليد رمز التفويض الذي سيطلبه النظام.

إعداد 2FA
عند تكوين 2FA لأول مرة:

1. قم بتثبيت تطبيق مصادق على هاتفك الذكي (Google Authenticator، Authy، Microsoft Authenticator، إلخ)
2. انتقل إلى إعدادات حساب OWL الخاص بك وقم بمسح رمز QR باستخدام تطبيق المصادقة الخاص بك
3. أدخل رمز التحقق لتأكيد الإعداد
4. احفظ رموز النسخ الاحتياطي في مكان آمن
5. اختبر توليد الرموز قبل الطوارئ

للحصول على تعليمات إعداد 2FA التفصيلية، راجع المصادقة الثنائية.

إذا قمت بتغيير الأجهزة المحمولة أو توقف التطبيق عن المزامنة مع حساب OWL الخاص بك، اتصل بمسؤول النظام للحصول على المساعدة. يمكن للمسؤولين إعادة ��عيين رموز 2FA من صفحة **المستخدمون والأدوار** ← **المستخدمون**.

قاعدة الشخصين
حيثما تتطلب العملية موافقة من شخص ثانٍ، يجب على الشخص الذي يصدر التنبيه إدخال كلمة المرور لمرة واحدة المعتمدة على الوقت للشخص الآخر قبل السماح بمتابعة العملية. يوفر ذلك إشرافًا ويقلل من مخاطر سوء الاستخدام.

أدوار المستخدم الدقيقة
يمكن تكوين أدوار المستخدم الفردية لـ:

- السماح فقط لبعض المستخدمين بإرسال رسائل محددة مسبقًا
- تقييد استهداف الرسائل لمناطق معينة
- تتطلب سير عمل موافقة إضافية

المرحلة 4: المراجعة

بمجرد اكتمال مراحل تعريف الرسالة، والاستهداف، والموافقة، يجب على المشغل مراجعة الرسالة قبل البث النهائي. بمجرد أن يكون راضيًا عن تفاصيل الرسالة، يمكنه نقل الرسالة.

سرعة النقل: في معظم الشبكات، يستغرق البث إلى جميع الأجهزة في الشبكة أقل من 10 ثوانٍ.

المرحلة 5: المراقبة والتحديثات

بمجرد بدء بث الرسالة، يمكن للمشغلين مراقبة وإدارة الرسائل المرسلة.

تعليقات الشبكة
تُرجع الشبكات الخلوية معلومات حول مواقع الخلايا التي بثت الرسالة. إذا كانت إحدى مواقع الخلايا غير متصلة أو غير متاحة، فسيتم الإبلاغ عن ذلك للمشغل.

إعادة النقل التلقائي
إذا أصبحت أي مواقع خلايا غير متصلة متاحة مرة أخرى بينما لا يزال التحذير الطارئ نشطًا، ستتلقى جميع الهواتف المتصلة بتلك الخلية الرسالة.

التحديثات أثناء الطيران
بمجرد البث، يمكن أن تكون الرسالة:

- محدثة مع تطور الوضع
- معدلة بمحتوى نص الرسالة الجديد
- مسترجعة/موقوفة في أي وقت

السجلات التاريخية
يمكن عرض جميع المعلومات حول الرسائل التاريخية ومراجعتها لأغراض التدقيق.

هيكل الرسالة

تتكون كل رسالة بث خلايا من:

تكوين الرسالة

- **معرف الرسالة** - معرف فريد لنوع التحذير (مثل، ETWS، 4370 لتحذير الزلزال، 4371 لتحذير التسونامي ETWS)

- **الفئة** - فئة التحذير (عادية، طارئة، عالية، شديدة)
- **فترة التكرار** - الثواني بين تكرارات البث
- **عدد البث** - عدد المرات التي سيتم بث الرسالة
- **فترة التحذير** - المدة بالثواني التي تكون فيها التحذير صالحًا
- **مؤشر القناة** - نوع القناة المستخدمة للبث

الرسائل المحلية

يمكن أن تتضمن كل رسالة CBC عدة متغيرات لغوية:

- **اللغة** - رمز اللغة ISO (en، es، fr، zh، إلخ)
- **نص الرسالة** - نص التحذير بتلك اللغة (حتى 1395 حرفًا)

يقوم النظام تلقائيًا ببث جميع المتغيرات اللغوية، مما يسمح للمستلمين بعرض التنبيهات بلغتهم المفضلة.

مناطق التتبع

تعرف الاستهداف الجغرافي للتحذير:

- **منطقة التتبع** - معرف جغرافي (معرف الخلية، رمز منطقة التتبع)
- **المشغل** (MCC-MNC) - رم�� مشغل الشبكة المحمولة
- **نوع RAT** - تقنية الوصول اللاسلكي (LTE، 5G، UMTS، GSM)

يمكن تحديد مناطق تتبع متعددة لتغطية مناطق أكبر أو مشغلين متعددين.

إنشاء رسالة بث الخلايا

عبر واجهة الويب:

1. من التنقل الرئيسي CBC **انتقل إلى بث الخلايا** - الوصول إلى واجهة إدارة
2. **انقر على "إنشاء تنبيه"** - يفتح نموذج إنشاء الرسالة
3. **تكوين معلمات الرسالة:**
    - معرف الرسالة (مثل، 4370 لتنبيهات الزلازل)
    - الفئة (عادية، عالية، شديدة)
    - فترة التكرار (عادة 5-60 ثانية)
    - عدد البث (999 للاستمرار، أو عدد محدد)

- فترة التحذير (المدة بالثواني)
- مؤشر القناة (عادة "أساسي")

4. **إضافة رسائل محلية:**
   - انقر على "إضافة لغة"
   - اختر اللغة من القائمة المنسدلة
   - أدخل نص الرسالة (حد أقصى 1395 حرفًا لـ GSM7، أقل لـ Unicode)
   - كرر للغات إضافية
5. **تحديد مناطق التتبع:**
   - انقر على "إضافة منطقة تتبع"
   - أدخل رمز منطقة التتبع
   - اختر المشغل (مجموعة MCC-MNC)
   - اختر نوع RAT (LTE، 5G، إلخ)
   - كرر لمناطق جغرافية إضافية

6. تحقق من جميع التفاصيل وانقر على "إنشاء تنبيه" - **مراجعة وإنشاء**"

عبر API:

نقطة النهاية: `PUT /crm/cbc/`

إذن مطلوب: `CREATE_CBC_MESSAGE`

جسم الطلب:

```
{
  "messageIdentifier": "4370",
  "category": "emergency",
  "repetitionPeriod": 10,
  "numberOfBroadcasts": 999,
  "warningPeriodSec": 3600,
  "channelIndicator": "basic",
  "localized_messages": [
    {
      "language": "en",
      "messageBody": "EARTHQUAKE WARNING: Magnitude 6.5 earthquake
detected. Take cover immediately. Drop, Cover, Hold On."
    },
    {
      "language": "es",
      "messageBody": "ADVERTENCIA DE TERREMOTO: Terremoto de
magnitud 6.5 detectado. Cúbrase inmediatamente. Agáchese, Cúbrase,
Agárrese."
    }
  ],
  "tracking_areas": [
    {
      "tracking_area": "12345",
      "operator": "310-410",
      "rat_type": "LTE"
    },
    {
      "tracking_area": "12346",
      "operator": "310-410",
      "rat_type": "5G"
    }
  ]
}
```


الاستجابة:

```
{
  "cbc_message_id": 123,
  "cbc_unique_id": "550e8400-e29b-41d4-a716-446655440000",
  "messageIdentifier": "4370",
  "category": "emergency",
  "repetitionPeriod": 10,
  "numberOfBroadcasts": 999,
  "warningPeriodSec": 3600,
  "channelIndicator": "basic",
  "initiating_user": 5,
  "approving_user": null,
  "created": "2025-01-10T14:30:00Z",
  "localized_messages": [...],
  "tracking_areas": [...]
}
```


تُرسل الرسالة على الفور إلى مركز بث الخلايا للإرسال.

إدارة الرسائل الحالية

عرض جميع الرسائل

نقطة النهاية: `GET /crm/cbc/`

إذن مطلوب: `VIEW_CBC_MESSAGE`

يُرجع قائمة بجميع رسائل CBC مع حالتها، وطوابعها الزمنية، وتكوينها.

تحديث رسالة

نقطة النهاية: `PATCH /crm/cbc/{cbc_message_id}`

إذن مطلوب: `UPDATE_CBC_MESSAGE`

تحديث محتوى الرسالة، أو مناطق التتبع، أو معلمات البث. تُعاد إرسال الرسائل المحدثة إلى CBC.

جسم الطلب:

```
{
  "cbc_message_id": 123,
  "numberOfBroadcasts": 500,
  "localized_messages": [
    {
      "language": "en",
      "messageBody": "UPDATED: Earthquake warning still in effect.
Aftershocks possible."
    }
  ]
}
```

حذف رسالة

نقطة النهاية: `DELETE /crm/cbc/{cbc_message_id}`

إذن مطلوب: `DELETE_CBC_MESSAGE`

يُزيل الرسالة من قاعدة البيانات ويحاول إلغائها على CBC.

سير عمل الموافقة

تدعم رسائل بث الخلايا سير عمل موافقة اختياري لتنبيهات ذات أهمية عالية:

1. **المستخدم المبادر** (`initiating_user` field) - عضو الطاقم الذي ينشئ التنبيه
2. **المستخدم الموافق** (`approving_user` field) - المدير الذي يوافق على التنبيه قبل البث

إذا كانت الموافقة مطلوبة:

- يتم إنشاء الرسالة مع `approving_user = null`
- تُحتفظ الرسالة في حالة "معلقة الموافقة"
- يقوم المستخدم الموافق بمراجعة الرسالة ويوافق عليها أو يرفضها
- عند الموافقة، يتم تعيين `approving_user` ويتم بث الرسالة

يمكن تكوين هذا سير العمل بناءً على سياسة المنظمة.

معرفات الرسالة

تتبع معرفات الرسائل القياسية 3GPP TS 23.041:

نظام تحذير الزلازل والتسونامي (ETWS):

- **4370** - تحذير زلزال ETWS
- **4371** - تحذير تسونامي ETWS
- **4372** - مجمع تحذير زلزال وتسونامي ETWS
- **4373-4378** - أنواع الطوارئ الأخرى ETWS
- **4379** - رسالة اختبار ETWS

تنبيهات الطوارئ اللاسلكية (WEA) / نظام التنبيه المحمول التجاري (CMAS):

- **4352** - تنبيه رئاسي
- **4353-4355** - تنبيهات شديدة
- **4356-4359** - تنبيهات خطيرة
- **4360-4363** - تنبيهات AMBER
- **4364-4367** - رسائل السلامة العامة
- **4368-4369** - اختبارات الدولة/المحلية
- **4380-4381** - رسائل اختبار

نطاقات مخصصة:

- **0-999** - محجوزة لتنبيهات محددة للمشغل
- **1000-4095** - أنواع الرسائل المخصصة

التكامل مع مركز بث الخلايا

إل�� آلية لتسليم الرسائل إلى الشبكات الخلوية الفردية. يتصل (CBE) يحتاج كيان بث الخلايا مركز بث الخلايا OWL (CBC) بكل شبكة خلوية لإرسال رسائل التحذير الطارئة إلى الجمهور.

التكرار عبر الشبكات المتعددة
في حالة عدم توفر مشغل شبكة خلوية واحد (انقطاع الخدمة أو عدم وجود تغطية)، إذا كانت شبكة خلوية أخرى تعمل، سيستمر المستخدمون في تلقي رسائل التحذير الطارئة عبر الشبكات المتاحة الأخرى.

لكل مشغل CBC حالات
منفصلة لكل CBC بتشغيل حالة OWL لضمان عدم وجود اتصال بين الشبكات المتنافسة، تقوم مشغل شبكة متنقلة؛ لا يتم مشاركتها بين المشغلين.

تكوين OWL CBC

يتم تكوين عنوان URL الخاص بـ CBC في `crm_config.yaml`:

```
cbc_url: "http://cbc.example.com:8080"
```

نقل الرسالة:

عند إنشاء أو تحديث رسالة CBC:

1. يقوم OmniCRM CBE بتخزين الرسالة في قاعدة بياناته
2. يتم تنسيق الرسالة لواجهة برمجة تطبيقات CBC
3. يتم إرسال طلب HTTP POST إلى `{cbc_url}/alerts/send`
4. يتصل CBC بالشبكة الخلوية عبر الواجهة المناسبة (CBSP، SBc-AP، أو N50)
5. يقر CBC باستلام الرسالة ويبدأ البث
6. تتلقى الأجهزة في مناطق التتبع المستهدفة التحذير

حذف الرسالة:

عند حذف رسالة، يحاول CBE إلغائها على CBC الجاري لإيقاف البث.

نقاط تكامل الشبكة الخلوية

تمتلك الأجيال المختلفة من الشبكات الخلوية (2G/3G/4G/5G) واجهات فريدة للتواصل مع مركز بث الخلايا. بناءً على التقنيات المستخدمة من قبل كل شبكة خلوية، يجب تكوين الواجهة الصحيحة.

يدعم مركز بث الخلايا OWL 2 واجهات بث الخلايا 2G و3G و4G و5G وقد تم دمجه مع العديد من مكونات الشبكة الخلوية المستخدمة بشكل شائع.

CBSP - 2G/3G - وحدة التحكم في محطة القاعدة (BSC)

يربط واجهة بروتوكول خدمة بث الخلايا (CBSP) مركز CBC بوحدة التحكم في محطة القاعدة (BSC) 2G التي تتحكم في محطات القاعدة (GSM).

- مع نشرات وحدة التحكم في الشبكة G3وG تستخدم لبث رسائل بث الخلايا 2 اللاسلكية/وحدة التحكم في محطة القاعدة المدمجة
- BSC يمكن تكوينها كعميل أو خادم حسب بائع
- OWL CBC في الشبكة و BSCs يجب إنشاء اتصال بين جميع
- CBSP قد تعطل يتم مراقبة الواجهات باستمرار مع التنبيه للإشارة إلى ما إذا كان رابط

ملاحظة: حددت 3GPP بروتوكول بث منطقة الخدمة (SABP) للاستخدام في نشرات RNC المستقلة. يمكن استخدام ذلك إذا لزم الأمر عندما لا يتم دعم CBSP 3 لخلايا G على RNC/BSC المدمجة، ومع ذلك قد تكون هناك حاجة لاختبارات إضافية ودعم من بائع RNC.

SBc-AP - 4G/5G غير المستقل - MME/IWF

يربط واجهة SBc-AP مركز OWL CBC بـ MME (كيان إدارة التنقل) 4 الذي يخدم 4G و5G eNodeB/gNodeBs.

- LTE 4G تستخدم لشبكات
- 5 غير المستقل (معظم النشر اعتبارًا من 2025) Gتستخدم أيضًا لـ
- OWL CBC في الشبكة و MMEs يجب إنشاء اتصال بين جميع
- SBc-AP قد يتم مراقبة الواجهات باستمرار مع التنبيه للإشارة إلى ما إذا كان رابط تعطل

N50 - 5G مستقل - AMF

وظيفة إدارة الوصول (AMF) بـ OWL CBC مركز N50 المستقلة، تربط واجهة 5G لشبكات 5G gNodeBs التي تخدم 5G (والتنقل).

- OWL CBC الواجهة موجودة في
- من طرف ثالث بسبب العدد القليل من شبكات AMFs لم يتم اختبارها بشكل مكثف مع 5G SA المتاحة تجاريًا في 2025
- 5G SA ستتم دعمها بالكامل مع زيادة شيوع نشرات

متطلبات الشبكة لمشغل الشبكة المحمولة

يجب أن تكون الشبكة موجودة بين OWL CBC وشبكة مشغل الشبكة المحمولة للوصول إلى الواجهات الموضحة أعلاه.

يتم التعامل مع ذلك على أساس كل حالة على حدة، ولكنه يتطلب عمومًا:

- MNO وشبكة CBC اتصال مخصص/ألياف بين

- فصل منطقي لكل واجهة
- الاتصال بكل نقطة تكامل (MME، RNC، BSC) في الشبكة الخلوية

المعدات الشبكية المدعومة

تم اختبار OWL CBC ويتكامل مع مكونات الشبكة الخلوية المستخدمة بشكل شائع من كبار البائعين:

تكامل بيانات موقع الخلية

يدعم OWL استخراج البيانات تلقائيًا من:

- **Nokia NetAct**
- **Huawei U2000 / U2020**
- **ZTE NetNumen / ZXPOS**
- **Ericsson ENM**

بدلاً من ذلك، يمكن توفير بيانات موقع الخلية لفريق العمليات في أومنيتوتش بشكل دوري عبر البريد الإلكتروني.

إدارة المستخدمين والتحكم في الوصول

التحكم في الوصول القائم على الدور (RBAC)

يستخدم نظام OWL التحكم في الوصول القائم على الدور (RBAC): يتم تعيين الأشخاص (المستخدمون المسجلون) لدور واحد أو أكثر، وكل دور هو مجموعة من الأذونات. الأذونات هي أصغر وحدة وصول (مثل، إنشاء تنبيه رسالة مسودة). الوصول الفعلي لمستخدم مسجل هو اتحاد الأذونات من جميع الأدوار المعينة.

مكونات RBAC:

- **المستخدمون** - أشخاص حقيقيون يقومون بتسجيل الدخول إلى نظام OWL
- **الأذونات** - قدرات صغيرة (مثل، الموافقة على مسودة الرسالة، إنشاء رسالة، عرض التقارير)
- **الأدوار** - مجموعات مسماة من الأذونات (مثل، موافقو الرسائل، منشئو الرسائل)
- **التعيين** - يحصل المستخدمون على دور واحد أو أكثر؛ تتجمع الأذونات

فوائد RBAC:

1. **حماية البيانات** - يرى المستخدمون فقط ما يُسمح لهم برؤيته وفعل ما يُسمح لهم بفعله
2. **ملاءمة تشغيلية** - تعكس الأدوار وظائف العمل (مدير، منشئ رسالة، موافق رسالة)
3. **إدارة بسيطة** - منح الوصول عن طريق تعيين الأدوار؛ تجنب الإدارة الدقيقة لكل مستخدم

أذونات النظام

تتبع أذونات النظام عمومًا أنماط CRUD بأربع خيارات:

- **عرض** - قراءة أو تصفح الرسائل والتقارير
- **إنشاء** - إنشاء أو إضافة تنبيه رسالة
- **تحديث** - تحرير أو تعديل تنبيه رسالة مسودة
- **حذف** - حذف أو إزالة تنبيه رسالة مسودة

الأذونات الأساسية لـ CBC:

- `CREATE_CBC_MESSAGE` - إنشاء رسائل بث جديدة
- `VIEW_CBC_MESSAGE` - عرض الرسائل الحالية وحالتها
- `UPDATE_CBC_MESSAGE` - تعديل محتوى الرسالة أو معلمات البث
- `DELETE_CBC_MESSAGE` - حذف الرسائل وإلغاء البث

قم بتعيين هذه الأذونات للأدوار بناءً على مسؤوليات السلامة العامة في منظمتك.

للحصول على معلومات شاملة حول إدارة أدوار المستخدمين والأذونات، راجع المستخدمون والأدوار.

إدارة حساب المستخدم

تسجيل الدخول لأول مرة

عند إعداد مستخدم جديد في النظام، يتم تزويده ببيانات اعتماد تسجيل الدخول من قبل مسؤول النظام:

1. انتقل إلى عنوان URL لتسجيل الدخول في OWL الخاص بك المقدم من مسؤول النظام
2. أدخل اسم المستخدم وكلمة المرور الخاصة بك

3. إذا تم تمكين 2FA لحسابك، ستتم مطالبتك بإعداده 2FA (أعلاه انظر قسم 2FA)

أفضل ممارسة: يجب على المستخدمين تغيير كلمة المرور الأولية الخاصة بهم على الفور بعد تسجيل الدخول الأول لأغراض الأمان. انتقل إلى إعدادات ملف تعريف المستخدم الخاص بك لتغيير كلمة المرور الخاصة بك.

لحصول على مزيد من التفاصيل حول تدفقات المصادقة، راجع تدفقات المصادقة والتحكم الإداري.

إعادة تعيين كلمة المرور

يمكن للمستخدمين إعادة تعيين كلمات مرورهم الخاصة:

1. من صفحة تسجيل الدخول إلى النظام، حدد "نسيت كلمة المرور"
2. اتبع عملية التحقق عبر البريد الإلكتروني
3. قم بتعيين كلمة مرور جديدة

إعادة تعيين كلمة مرور المسؤول:

يمكن لمسؤولي النظام إعادة تعيين كلمات المرور للمستخدمين من صفحة **المستخدمون والأدوار** → **المستخدمون**:

1. انتقل إلى **المستخدمون والأدوار** → **المستخدمون**
2. حدد المستخدم الذي يحتاج إلى إعادة تعيين كلمة المرور
3. انقر على **إعادة تعيين كلمة المرور**
4. يقوم النظام بإنشاء كلمة مرور مؤقتة عشوائية آمنة (12 حرفًا)
5. تُعرض كلمة المرور المؤقتة للمسؤول
6. قدم كلمة المرور المؤقتة للمستخدم عبر قناة آمنة (هاتف أو شخصيًا، وليس عبر البريد الإلكتروني أبدًا)
7. يمكن للمستخدم تسجيل الدخول باستخدام كلمة المرور المؤقتة
8. **مهم:** يجب على المستخدم تغيير هذه الكلمة المرور على الفور بعد تسجيل الدخول

للحصول على مزيد من التفاصيل، راجع المستخدمون والأدوار.

إضافة مستخدمين جدد للنظام

يمكن لمسؤولي النظام إضافة مستخدمين جدد عن طريق الانتقال إلى **المستخدمون والأدوار** → **المستخدمون**:

المعلومات المطلوبة لكل مستخدم مسجل جديد:

- الاسم الأول
- الاسم الأخير
- البريد الإلكتروني
- كلمة المرور (مؤقتة)
- رقم الهاتف المحمول
- الدور (الأدوار)

أفضل ممارسة: تنفيذ نموذج طلب مستخدم جديد يتماشى مع السياسات والإجراءات الحالية لمنظمتك. أضف المستخدمين فقط بعد الحصول على الموافقة المناسبة.

تحرير ملفات تعريف المستخدمين

يمكن لمسؤولي النظام تغيير الأذونات والوصول للمستخدمين الحاليين:

1. انتقل إلى **المستخدمون والأدوار** ← **المستخدمون**
2. حدد المستخدم الذي تريد تحريره
3. تعديل الأدوار المعينة حسب الحاجة
4. حفظ التغييرات

حذف مستخدمي النظام

يمكن لمسؤولي النظام حذف المستخدمين من **المستخدمون والأدوار** ← **المستخدمون**.

تحذير: بمجرد حذف مستخدم من نظام OWL، لا يمكن التراجع عن هذا الإجراء. يجب توخي الحذر الشديد قبل تنفيذ هذه المهمة.

عرض جميع مستخدمي النظام

في أي وقت، يمكن لمسؤول النظام عرض قائمة بجميع المستخدمين المسجلين ورؤية الأذونات والأدوار التي يمتلكها كل منهم عن طريق الانتقال إلى **المستخدمون والأدوار** ← **المستخدمون**.

تاريخ رسائل التنبيه

يحتفظ نظام OWL بسجل لجميع رسائل التنبيه المرسلة لأغراض التدقيق ولدعم المراجعة التشغيلية.

الوصول إلى تاريخ الرسائل:

انتقل إلى **إدارة بث الخلايا** لع��ض:

- جميع الرسائل المرسلة سابقًا
- حالة الرسالة (نشطة، منتهية، ملغاة)
- طابع الزمن للإرسال
- المستخدمون المبادرون والموافقون
- تفاصيل الاستهداف الجغرافي
- محتوى الرسالة ومعلماتها

يساعد ذلك في:

- تدقيق النظام
- مراجعة إجراءات التشغيل المستخدمة من قبل NDMO ووكالات التنبيه الأخرى
- فهم فعالية الرسالة
- متطلبات الامتثال والتقارير

تصور خريطة بث الخلايا

توفر خريطة بث الخلايا واجهة **بصرية** لتخطيط مناطق التغطية الجغرافية وتحديد أي أبراج خلوية ستبث التنبيهات.

الوصول إلى الخريطة

انتقل إلى:

أو مباشرة:

إذن مطلوب: `VIEW_CBC_MESSAGE` أو `CREATE_CBC_MESSAGE`

ميزات الخريطة

تعرض الخريطة:

مواقع أبراج الخلايا:

- **أيقونات الأبراج** - يتم عرض مواقع مواقع الخلايا كأيقونات أبراج خلوية
- **مؤشرات القطاع** - أسهم توجيهية تظهر اتجاه القطاع الهوائي
- **أسماء ��لأبراج** - معرفات وأسماء المواقع
- **بيانات التغطية** - محملة من ملف `cellSites.txt`

أدوات الرسم:

- **أداة المضلع** - رسم مناطق تغطية مخصصة
- **أداة الدائرة** - إنشاء مناطق بث دائرية
- **أداة المستطيل** - تعريف مناطق تغطية مستطيلة

تحليل التغطية:

- **قائمة المواقع** - جدول يظهر جميع مواقع الخلايا ضمن المنطقة المرسومة
- **إدارة المناطق** - حفظ وتحميل مناطق التغطية المحددة مسبقًا
- **عدد المواقع** - عدد الأبراج في منطقة التغطية في الوقت الحقيقي

استخدام الخريطة

الخطوة 1: تحميل بيانات موقع الخلية

عند تحميل الصفحة. `/cellSites.txt` تقوم الخريطة بتحميل مواقع أبراج الخلايا تلقائيًا من يحتوي هذا الملف على:

- **lat/lng** - إحداثيات البرج
- **site_name** - المعرف
- **sectors** - اتجاهات الهوائي بالدرجات

الخطوة 2: رسم منطقة التغطية

اختر أداة رسم:

1. انقر على **المضلع** لرسم أشكال مخصص ��
    - انقر لإضافة نقاط
    - انقر نقرتين لإكمال
    - ينشئ حدود تغطية دقيقة
2. انقر على **الدائرة** لرسم مناطق دائرية
    - انقر على نقطة المركز

    - اسحب لتعيين نصف القطر
    - مفيد لتنبيهات سريعة تعتمد على نصف القطر
3. انقر على **المستطيل** لرسم مناطق مستطيلة
    - انقر على زاوية واحدة
    - اسحب إلى الزاوية المقابلة
    - فعال للتغطية المتوافقة مع الشبكة

الخطوة 3: مراجعة المواقع المغطاة

بعد الرسم، يقوم النظام:

- بحساب مواقع الخلايا التي تقع ضمن المنطقة
- يعرض القائمة في جدول أسفل الخريطة
- يظهر أسماء المواقع وإحداثياتها
- يوفر عدد الأبراج المتأثرة

جدول التغطية المثال:

الخطوة 4: تصدير بيانات التغطية

استخدم بيانات التغطية لـ:

- **تحديد رموز مناطق التتبع** للبث
- **تقدير نطاق التنبيه** بناءً على عدد الأبراج
- **تخطيط الاستجابة للطوارئ** من خلال فهم الم��اطق المتأثرة
- **حفظ تعريفات المناطق** للاستخدام في المستقبل

إدارة المناطق

حفظ المناطق:

يمكن حفظ مناطق التغطية المحددة مسبقًا لسيناريوهات شائعة:

- **المنطقة الساحلية** - لتحذيرات التسونامي/ارتفاع العواصف
- **مناطق الحرائق** - لتنبيهات الإجلاء
- **المراكز الحضرية** - لتنبيهات AMBER في المناطق المأهولة
- **الممرات السريعة** - لتنبيهات المرور/الطقس

تنسيق بيانات المنطقة:

تُخزن المناطق في `site_data.json`:

```
[
  {
    "area": "Coastal Region",
    "polygon": [
      {
        "coord": [
          {"lat": -33.8688, "lng": 151.2093},
          {"lat": -33.8650, "lng": 151.2070},
          {"lat": -33.8600, "lng": 151.2150},
          {"lat": -33.8688, "lng": 151.2093}
        ]
      }
    ]
  }
]
```

تحميل المناطق المحفوظة:

1. اختر المنطقة من القائمة المنسدلة
2. يظهر المضلع على الخريطة تلقائيًا
3. يتم تحديث قائمة المواقع لعرض الأبراج المغطاة
4. تعدي�ل المضلع إذا لزم الأمر

تكوين الخريطة

واجهة برمجة تطبيقات خرائط Google:

تتطلب الخريطة مفتاح واجهة برمجة تطبيقات خرائط Google الذي يتم تكوينه في البيئة:

```
VITE_GOOGLE_API_KEY=your_google_maps_api_key_here
```

بيانات موقع الخلية:

قم بتحديث `public/cellSites.txt` بمواقع أبراج الخلايا الخاصة بشبكتك:

```
# التنسيق: lat,lng,name,sector1,sector2,sector3
-33.8688,151.2093,SYD_CENTRAL,0,120,240
-33.8650,151.2070,SYD_HARBOUR,45,165,285
```

احصل على إحداثيات موقع الخلية من:

- أدوات تخطيط الشبكة
- سجلات تركيب الأبراج
- المسوحات الميدانية باستخدام GPS
- تكوين وحدة التحكم في الشبكة اللاسلكية (RNC)

تصميم الخريطة:

يمكن تكوين أنماط خريطة مخصصة في `mapStyles.js` لـ:

- تسليط الضوء على خدمات الطوارئ
- عرض الميزات الطبوغرافية
- التأكيد على مراكز السكان
- مطابقة العلامة التجارية للمنظمة

حالات الاستخدام

تخطيط الطوارئ:

- التخطيط ا��مسبق لمناطق التغطية للمناطق المعروفة بالخطر
- اختبار استراتيجيات استهداف مختلفة
- تقدير نطاق السكان
- التنسيق مع خدمات الطوارئ

التحقق من التنبيه:

- تأكيد أن الرسالة ستصل إلى المنطقة المستهدفة
- تحديد الفجوات في التغطية
- تجنب التنبيه المفرط للمناطق المجاورة
- التحقق من رموز مناطق التتبع

تحليل الشبكة:

- تصور توزيع أبراج الخلايا
- تحديد التداخلات في التغطية
- تخطيط نشر الأبراج لتحسين تغطية التنبيه
- تحسين كفاءة البث

أفضل الممارسات

استنادًا إلى الخبرة مع العملاء حول العالم، توصي أومنيتوتش بأفضل الممارسات التالية لجميع نشرات OWL.

محتوى الرسالة:

- اجعل الرسائل مختصرة وواضحة (أقل من 360 حرفًا لتنبيهات الصفحة الواحدة)
- استخدم جميع الأحرف الكبيرة لتنبيهات الطوارئ (يعزز القراءة)
- تضمين عناصر العمل المحددة ("الإجلاء الفوري"، "ابحث عن مأوى")
- تجنب المصطلحات التقنية
- اختبار الرسائل مع الأجهزة الفعلية قبل الطوارئ

دعم اللغة:

- دائمًا تقديم الرسائل باللغة الرئيسية للمنطقة
- تضمين لغات إضافية للمناطق متعددة الثقافات
- التأكد من أن الترجمات مناسبة ثقافيًا
- اختبار الأحرف الخاصة ودعم Unicode

الاستهداف الجغرافي:

- استخدام أصغر مناطق التتبع اللازمة لتجنب إرهاق التنبيه
- مراعاة كثافة السكان عند تعيين فترة التكرار
- اختبار الاستهداف الجغرافي قبل الطوارئ
- الحفاظ على توثيق دقيق لمناطق التتبع

الاختبار:

- استخدام معرفات الرسائل الاختبارية (4379، 4380-4381) للتدريبات
- جدولة اختبارات النظام المنتظمة

- يعمل CBC التحقق من أن تكامل
- تدريب الموظفين على إجراءات الطوارئ

إرهاق التنبيه:

- استخدامه فقط في حالات الطوارئ الحقيقية
- تجنب البث المفرط
- تعيين فترات التكرار المناسبة
- استخدام مستويات الشدة بشكل مناسب

أمان حساب المستخدم

أمان رمز 2FA:

- مثل بطاقات الوصول إلى المباني FAحماية الرموز المادية لـ 2
- الإبلاغ عن الرموز المفقودة أو المسروقة على الفور
- بانتظام لضمان عمله FAاختبار 2
- وتخزينها بأمان في وضع عدم الاتصال FAحفظ رموز النسخ الاحتياطي عند إعداد 2
- الاتصال بمسؤول النظام إذا كنت بحاجة إلى إعادة تعيين FA2

للحصول على مزيد من المعلومات حول إعداد FA2 راجع المصادقة الثنائية، واسترداده.

صيانة البيانات

المناطق المستهدفة المحددة مسبقًا:

مع تغير الحدود الجغرافية، ووقوع التطورات، وتحول مناطق الخطر، هناك حاجة لمراجعة المناطق المستهدفة المحددة مسبقًا. تقترح أومنيتوتش مراجعة هذه البيانات **سنويًا من قبل NDMO**، مع دعم من وكالات التنبيه الأخرى عند الاقتضاء.

يمكن تعريف التحديثات للحدود باستخدام عدة منصات GIS شائعة أو Google Earth، ثم تقديمها عبر البريد الإلكتروني إلى فريق عمليات أومنيتوتش الذي سيجري التغييرات على النظام.

قوالب الرسائل المحددة مسبقًا:

يجب مراجعة قوالب الرسائل المحددة مسبقًا **مرة واحدة على الأقل سنويًا من قبل NDMO**، لضمان دعم من وكالات التنبيه الأخرى عند الاقتضاء:

- أن محتوى الرسالة لا يزال يعكس بدقة الخطر والدعوة للعمل
- أن معلومات الاتصال والتعليمات محدثة
- أن الترجمات اللغوية تظل دقيقة
- أن نبرة الرسالة وإلحاحها مناسبة
- أن الإشارات إلى خدمات الطوارئ أو الإجراءات محدثة

يمكن تقديم التحديثات عبر البريد الإلكتروني إلى فريق عمليات أومنيتوتش الذي سيجري التغييرات على النظام.

بيانات موقع الخلية:

- مراجعة وتحديث بيانات موقع الخلية كلما حدثت تغييرات في الشبكة
- تكرار التحديث النموذجي: شهريًا أو ربع سنوي
- التنسيق مع فرق تخطيط الشبكة في MNO
- التحقق من الدقة بعد التحديثات أو التوسعات الكبرى في الشبكة

بنية النظام

تم تصميم جميع منتجات أومنيتوتش لدعم نشرات موزعة جغرافيًا.

خيارات النشر

يمكن تشغيل جميع المكونات كـ:

- **حاويات (K8s)** - نشرات حاوية منسقة بواسطة Kubernetes
- **آلات افتراضية** - VMware، Proxmox، HyperV
- **سحابة خاصة** - بنية تحتية سحابية محلية
- **سحابة عامة** - AWS، GCP
- **أجهزة معدنية** - نشر مباشر على الأجهزة

البنية الموزعة

تسمح البنية الموزعة بـ:

- وتوزيع الرسائل حتى إذا OWL CBE لمكاتب إدارة الكوارث المحلية الوصول إلى أصبحت منطقة ما معزولة عن الشبكة الوطنية
- وجود كيانات بث خلايا متعددة ومراكز بث خلايا متعددة لكل مشغل/كل بلد

- لوكالات الاستجابة للكوارث المحلية (الحكومة البلدية، الشرطة، الإطفاء، إلخ) إصدار الوصول (NDMO) تنبيهات لمنطقتها حتى إذا فقد المكتب الوطني لإدارة الكوارث

هذا ذو قيمة خاصة عندما يكون لدى MNOs شبكة خلوية موزعة مع موارد BSC/MME المحلية.

الشبكة بين CBC و CBE

لضمان عدم وجود اتصال بين الشبكات المتنافسة، يمتلك كل مشغل شبكة متنقلة حالة CBC منفصلة (غير مشتركة).

متطلبات الشبكة بين CBC و CBE:

- يتم تشفير جميع حركة المرور بين CBC و CBE
- المصادقة تعتمد على الشهادات المتبادلة
- الاتصال من CBE إلى CBC عبر TLS على منفذ TCP 443
- التنسيق مطلوب بين CBE أو الوكالة المستضيفة لـ NDMO و MNOs

اعتبارات الوصول

ستحتاج الوكالة المستضيفة لـ CBE إلى تحديد إجراءات الوصول للمستخدمين النهائيين (مثل Citrix، VPN، إلخ)، مع الأخذ في الاعتبار أن النظام يجب أن يكون متاحًا في السيناريوهات غير المثالية مثل:

- انقطاع واسع النطاق لشبكات الطاقة العامة
- فشل شبكات الاتصالات
- الكوارث الطبيعية التي تؤثر على البنية التحتية

متطلبات النشر

متطل��بات VM لـ CBE (NDMO / الحكومة)

آلات افتراضية 3x:

- **2x VMs لـ CBE** (يفضل في مراكز بيانات/مناطق توفر مختلفة)
- **1x VM للمراقبة**

تتطلب كل VM:

- 50 **التخزين:** GB
- 2 **وحدة المعالجة المركزية:** x وحدة معالجة مركزية افتراضية
- 8 **الذاكرة العشوائية:** GB
- **نظام التشغيل:** نظام التشغيل الأساسي المقدم من أومنيتوتش
- **الشبكة:** السماح بحركة المرور إلى TCP 443 لحركة مرور CBC على منفذ VMs لـ TLS للتحكم في CBCs

متطلبات CBC لـ VM (MNO)

آلات افتراضية 3x:

- **2x VMs لـ CBC** (يفضل في مراكز بيانات/مناطق توفر مختلفة)
- **1x VM للمراقبة**

تتطلب كل VM:

- 50 **التخزين:** GB
- 2 **وحدة المعالجة المركزية:** x وحدة معالجة مركزية افتراضية
- 8 **الذاكرة العشوائية:** GB
- **نظام التشغيل:** نظام التشغيل الأساسي المقدم من أومنيتوتش
- **الاتصال:** بكل من نقاط التكامل في الشبكة الخلوية (MME، RNC، BSC)
- **الشبكة:** السماح بحركة المرور من VMs لـ CBE على منفذ TCP 443 للتحكم في CBE بواسطة TLS حركة مرور

خطوات التكامل

يتضمن نشر OWL الخطوات التالية:

1. **تحديد مشغل/مشغلي الشبكة المحمولة الجدد** - تحديد مشغلي الشبكة المحمولة المشاركين
2. **إدارة المشروع (الإعداد)** - إنشاء حوكمة المشروع والإدارة
3. **اختيار موقع استضافة لـ CBE** - تحديد مكان استضافة CBE
4. **تحديد المستخدمين وتدفقات/إجراءات الرسائل** - إنشاء سير عمل الموافقة وأدوار المستخدمين
5. **تحديد مضلعات لمناطق الاستهداف وقوالب الرسائل** - إعداد السيناريوهات الشائعة مسبقًا

6. **تخصيص عناوين IP** - تخصيص عناوين IP لـ NDMO وMNOs
7. **إعداد VPN من موقع إلى موقع** - إنشاء اتصال آمن مع فريق أومنيتوتش
8. **نشر VMs لـ CBE إلى NDMO** - تثبيت كيان بث الخلايا
9. **نشر VMs لـ CBC إلى MNO** - تثبيت مركز بث الخلايا في كل مشغل
10. **تكوين عناصر الشبكة** - إعداد الاتصال بـ CBC من معدات الشبكة الخلوية
11. **الشبكة بين CBC و CBE لـ VMs** - إنشاء اتصال آمن
12. **الشبكة بين CBC لـ VMs وعناصر الشبكة** - الاتصال بـ BSC/MME/AMF
13. **إعداد الوصول إلى واجهة برمجة التطبيقات لـ NMS** - تكوين تكامل بيانات موقع الخلية
14. **إعداد المراقبة واختبارها** - التحقق من المراقبة والتنبيه
15. **التحقق/اختبار التنبيهات التجريبية** - إجراء اختبار النظام
16. **اختبار التحذير العام** - إجراء اختبار شامل من النهاية إلى النهاية

حدود الأحرف

تمتلك رسائل بث الخلايا حدودًا صارمة للأحرف بناءً على الترميز:

ترميز GSM 7-bit (الإنجليزية، الأحرف اللاتينية الأساسية):

- صفحة واحدة: 93 حرفًا
- صفحات متعددة: 15 صفحة × 93 = 1395 حرفًا كحد أقصى

ترميز Unicode UCS-2 (الكتابات غير اللاتينية، الرموز التعبيرية):

- صفحة واحدة: 41 حرفًا
- صفحات متعددة: 15 صفحة × 41 = 615 حرفًا كحد أقصى

منصة OWL:

- نص الرسالة محدود إلى 500 حرف
- تعرض واجهة الويب عدد الأحرف المتبقية وتحذر عند الاقتراب من الحدود

المرا��بة والسجلات

يتم تسجيل نشاط بث الخلايا لأغراض التدقيق:

- أحداث إنشاء الرسالة، التحديث، والحذف
- المستخدمون المبادرون والموافقون
- الطوابع الزمنية ومعرفات الرسائل
- استجابات واجهة برمجة التطبيقات لـ CBC والأخطاء
- تفاصيل الاستهداف الجغرافي

يمكن الوصول إلى السجلات عبر سجل النشاط أو استعلامات قاعدة البيانات:

```
SELECT * FROM cbc
WHERE created >= '2025-01-01'
ORDER BY created DESC;
```

التكامل مع الأجهزة المحمولة

تُستقبل رسائل بث الخلايا بواسطة الأجهزة المحمولة المتوافقة:

دعم الأجهزة:

- تدعم معظم الهواتف الذكية من عام 2015 فصاعدًا بث الخلايا
- قد تحتوي الهواتف الأساسية على دعم محدود
- يجب أن تكون الجهاز متصلًا بالشبكة (لا حاجة إلى بيانات/رصيد SMS)
- يعمل SMS حتى أثناء الازدحام الشبكي عندما تفشل

تجربة المستخدم:

- يظهر التنبيه كإشعار ملء الشاشة
- يتم تشغيل نغمة تنبيه فريدة
- يستمر التنبيه حتى يتم الاعتراف به
- لا حاجة للاشتراك من المستخدم
- لا يمكن حظره من قبل المستخدمين لتنبيهات رئاسية/شديدة

اختبار استقبال الجهاز:

للتحقق من أن الأجهزة يمكنها استقبال التنبيهات:

1. إرسال رسالة اختبار (معرف 4379 أو 4380-4381)

2. التأكد من أن الجهاز في منطقة التتبع المستهدفة
3. التحقق من أن الجهاز لديه بث الخلايا مفعل في الإعدادات
4. التحقق مع نماذج أجهزة متعددة وإصدارات نظام التشغيل

الوظائف الإضافية

يمكن توسيع منصة OWL مع ميزات اختيارية لتكملة رسائل بث الخلايا:

الرسائل النصية الجماعية / SMS

إرسال SMS عادية للأفراد لإشعارات إضافية.

- غالبًا ما تستخدم لتنبيهات منخفضة المستوى لإبلاغ الناس بأن التهديد الفوري قد زال، بطريقة أقل إزعاجًا
- أبطأ بكثير من بث الخلايا ولكن يمكن أن تشمل آلية تأكيد للتحقق من استلام الرسالة
- مفيدة للتواصل المستهدف للمتابعة

المكالمات الصوتية للخط الثابت

الاتصال تلقائيًا بأرقام الخط الثابت وتشغيل رسائل تحذير الطوارئ.

- تشغيل رسالة مسجلة مسبقًا
- تقديم الرسالة التحذيرية للطوارئ عبر تحويل النص إلى كلام
- تصل إلى السكان الذين ليس لديهم هواتف محمولة
- يمكن التحقق من تسليم الرسالة من خلال إتمام المكالمة

النشر عبر وسائل التواصل الاجتماعي

نشر رسائل التحذير الطارئة تلقائيًا على القنوات الرسمية لوسائل التواصل الاجتماعي.

- يوسع نطاق الوصول إلى ما وراء الشبكة الخلوية
- يوفر مرجعًا لأولئك الذين فاتتهم التنبيه الأولي
- يسمح برسائل موسعة تتجاوز حدود الأحرف

البث التلقائي عبر الراديو/التلفزيون

بث رسائل الطوارئ تلقائيًا عبر الراديو والتلفزيون.

- تشغيل رسالة مسجلة مسبقًا
- تقديم الرسالة التحذيرية للطوارئ عبر تحويل النص إلى كلام
- تصل إلى السكان أثناء انقطاع الشبكة
- تكمل بث الخلايا لتغطية شاملة

أجهزة التحذير الخارجية

يمكن لمركز OWL CBC الاتصال بمجموعة متنوعة من المصادر الخارجية:

- منصات وسائل التواصل الاجتماعي
- واجهات برمجة التطبيقات العامة
- أنظمة المكالمات الصوتية
- أنظمة البث الإذاعي
- الإنذارات والأجراس الفيزيائية
- اللافتات الإلكترونية

يمكن استكشاف خيارات التخصيص كجزء من مرحلة التصميم.

التخصيص والصيانة

الاختبارات الدورية

يجب إجراء اختبارات دورية للحل على فترات منتظمة لضمان:

- أن الحل وجميع المكونات تعمل بشكل صحيح
- أن جميع الموظفين على دراية بالعمليات والإجراءات المطلوبة لإصدار رسائل التحذير الطارئة
- أن نقاط التكامل تظل عاملة
- أن قوالب الرسائل محدثة وفعالة

الاختبارات الموصى بها:

- رسائل اختبار شهرية باستخدام معرفات 4379، 4380-4381
- اختبارات نظام كاملة ربع سنوية تشمل سير عمل الموافقة
- اختبارات تحذير عامة سنوية مع إشعار مسبق
- جلسات تدريب منتظمة للمستخدمين المصرح لهم

صيانة بيانات موقع الخلية

عندما يضيف المشغلون أو يزيلون مواقع الخلايا، أو يغيرون مناطق/معرفات التتبع لمواقع الخلايا، يجب مشاركة هذه المعلومات مع فريق أومنيتوتش لضمان دقة بيانات أداة التخطيط.

التكامل التلقائي للبيانات

يدعم OWL استخراج البيانات تلقائيًا من:

- Nokia NetAct
- Huawei U2000 / U2020
- ZTE NetNumen / ZXPOS
- Ericsson ENM

التحديثات اليدوية للبيانات

بدلاً من ذلك، يمكن توفير بيانات موقع الخلية لفريق عمليات أومنيتوتش بشكل دوري عبر البريد الإلكتروني بتنسيقات مختلفة.

تكرار التحديث: مراجعة وتحديث بيانات موقع الخلية كلما حدثت تغييرات في الشبكة، عادةً شهريًا أو ربع سنوي.

المناطق المستهدفة المحددة مسبقًا

مع تغير الحدود الجغرافية، ووقوع التطورات، وتحول مناطق الخطر، هناك حاجة لمراجعة المناطق المستهدفة المحددة مسبقًا المستخدمة في مرحلة الاستهداف.

المراجعة السنوية: يجب مراجعة المناطق المستهدفة المحددة مسبقًا سنويًا من قبل NDMO (المكتب الوطني لإدارة الكوارث).

عملية التحديث: يمكن تعريف التحديثات للحدود باستخدام عدة منصات شائعة أو GIS Google Earth، ثم تقديمها إلى فريق عمليات أومنيتوتش.

قوالب الرسائل المحددة مسبقًا

المراجعة السنوية: يجب مراجعة قوالب الرسائل المحددة مسبقًا سنويًا من قبل NDMO لضمان:

- أن محتوى الرسالة يعكس الإجراءات الطارئة الحالية
- أن الترجمات اللغوية دقيقة
- أن معرفات الرسائل مناسبة
- أن معلومات الاتصال والتعليمات محدثة

عملية التحديث: يمكن تقديم التحديثات عبر البريد الإلكتروني إلى فريق عمليات أومنيتوتش.

تدفقات موافقة الرسائل

تمتلك المناطق المختلفة متطلبات مختلفة حول كيفية إصدار الرسائل وتدفقات الموافقة.

قاعدة الشخصين: يُنصح باستخدام قاعدة الشخصين حيثما كان ذلك عمليًا لضمان الإشراف في تقديم الرسائل.

أدوار المستخدم الدقيقة: يمكن تكوين أدوار المستخدم الفردية لـ:

- السماح فقط لبعض المستخدمين بإرسال رسائل محددة مسبقًا
- تقييد الاستهداف لمناطق معينة
- تتطلب خطوات موافقة إضافية
- تقليل مخاطر سوء الاستخدام

صيانة الأمان / 2FA

يمكن لفريق عمليات أومنيتوتش الدعم في:

- إعادة تعيين رموز 2FA
- إعادة إصدار الرموز المفقودة/التالفة/المنتهية
- تدقيق الأمان وإدارة الرموز

إجراءات أمان الرموز:

عند إصدار الرموز، يتم تفصيل إجراء للخطوات التي يجب اتخاذها إذا أصبحت رمزًا مفقودًا أو غير محسوب. يجب على الشخص الذي يقبل الرمز اتباع هذه الإجراءات لضمان عدم إساءة استخدام

النظام.

الإجراءات الفورية للرموز المفقودة:

1. الإبلاغ عن الرمز المفقود على الفور إلى الأشخاص المخولين
2. يتم تعطيل الرمز في النظام
3. يتم إجراء مراجعة أمنية
4. يتم إصدار رمز جديد بعد التحقق من الأمان

التكامل مع واجهة برمجة التطبيقات الخارجية

تتوفر مجموعة كاملة من واجهات برمجة التطبيقات لكيان بث الخلايا للسماح للأنظمة الخارجية بالتكامل والتفاعل مع CBE.

قدرات واجهة برمجة التطبيقات:

- **التقارير/المراقبة** - التحقق من حالة ونطاق الرسائل المرسلة
- **إنشاء الرسائل** - إنشاء وبث رسائل جديدة برمجيًا
- **التحقق من الصحة** - التحقق دوريًا من صحة النظام مع توليد حركة مرور رسائل اختبار روتينية
- **استعلامات الحالة** - استرداد حالة الرسالة، وإحصائيات التسليم، وقياسات النظام

راجع قسم مرجع واجهة برمجة التطبيقات أدناه للحصول على وثائق مفصلة حول النقاط النهائية.

مرجع واجهة برمجة التطبيقات

تتطلب جميع نقاط نهاية CBC المصادقة والأذونات المناسبة.

إنشاء رسالة:

```
PUT /crm/cbc/
```

الحصول على جميع الرسائل:

```
GET /crm/cbc/
```

تحديث رسالة:

`PATCH /crm/cbc/{cbc_message_id}`

حذف رسالة:

`DELETE /crm/cbc/{cbc_message_id}`

للحصول على مواصفات واجهة برمجة التطبيقات `/crm/docs/` في Swagger راجع وثائق التفصيلية.

البحث العالمي

تقدم ميزة البحث العالمي واجهة بحث **موحدة** للعثور بسرعة على العملاء، جهات الاتصال، الخدمات، المخزون، والمواقع عبر قاعدة بيانات OmniCRM بالكامل.

انظر أيضًا: العملاء، المخزون، إدارة الخدمات.

الوصول إلى البحث العالمي

من أي مكان في نظام إدارة العلاقات مع العملاء:

انقر على أيقونة البحث في شريط التنقل العلوي أو انتقل إلى:

تظهر صفحة البحث العالمي مع مربع بحث كبير وخيارات تصفية.

كيف يعمل

يقوم البحث العالمي بإجراء **بحث عبر الكيانات** عبر خمسة أنواع من البيانات:

ما الذي يتم البحث عنه:

1. **العملاء** - اسم العميل
2. **جهات الاتصال** - الاسم الأول، الاسم الأخير، عنوان البريد الإلكتروني، رقم الهاتف
3. **المواقع** - اسم الموقع
4. **المخزون** - أرقام التسلسل، المعرفات (itemtext1، itemtext2)، ICCIDs
5. **الخدمات** - اسم الخدمة، UUID الخدمة

سلوك البحث:

- **مطابقة جزئية** (على سبيل المثال، "Smith" يتطابق مع "John Smith" و "Smithson")
- **غير حساسة لحالة الأحرف** - "john" يتطابق مع "John"، "JOHN"، و "john"
- **كيانات متعددة** - البحث الواحد يعيد النتائج من جميع أنواع الكيانات
- **نتائج مقسمة** - تظهر 10 نتائج لكل صفحة بشكل افتراضي

إجراء بحث

البحث الأساسي

1. أدخل مصطلح البحث الخاص بك في مربع البحث
2. انقر على **"بحث"** أو اضغط على Enter

مصطلحات البحث المثال:

- اسم العميل: `"Acme Corp"`
- رقم الهاتف: `"1234567890+"` أو `"1234567890"`
- البريد الإلكتروني: `"john"` أو `"john@example.com"`
- رقم التسلسل: `"ICCID8944"` `8944"` أو فقط`"`
- UUID `123"` :الخدمة`e4567-e89b"`

تضمين الحسابات المغلقة

بشكل افتراضي، يعيد البحث النتائج من حسابات العملاء **المفتوحة** فقط.

للبحث عبر جميع الحسابات بما في ذلك المغلقة:

1. تحقق من مربع **"تضمين الحسابات المغلقة"**
2. انقر على **"بحث"** مرة أخرى

سيبحث هذا في:

- العملاء الذين لديهم `customer_status = "مغلق"`
- جهات الاتصال، الخدمات، المواقع، والمخزون المرتبط بالعملاء المغلقين

حالات استخدام البحث عن الحسابات المغلقة:

- العثور على سجلات العملاء التاريخية
- تحديد المعدات من الخدمات التي تم إلغاء تفعيلها
- البحث عن أرقام الهواتف القديمة أو الخدمات
- استعادة بيانات العملاء لإعادة التفعيل

فهم نتائج البحث

تنسيق عرض النتائج

تُعرض النتائج في قائمة قابلة للتمرير تظهر:

> معرف العميل: 123 النوع: عميل John Smith
>
> معرف العميل: 123 النوع: جهة اتصال (جهة اتصال) John Smith
>
> الهاتف المحمول - +44 7700 900123 معرف العميل: 123 النوع: خدمة

كل نتيجة تظهر:

- **الاسم/العنوان** - المعرف الأساسي (رابط قابل للنقر)
- **معرف العميل** - العميل الرئيسي الذي ينتمي إليه
- **النوع** - نوع الكيان (عميل، جهة اتصال، موقع، مخزون، خدمة)

أنواع النتائج موضحة

نتائج العملاء:

النقر يفتح صفحة نظرة عامة على العميل تظهر جميع التفاصيل، الخدمات، جهات الاتصال، إلخ.

نتائج جهات الاتصال:

النقر يفتح صفحة العميل مع علامة التبويب جهات الاتصال نشطة، متجهة إلى جهة الاتصال المحددة.

نتائج المواقع:

النقر يفتح صفحة العميل مع علامة التبويب المواقع نشطة.

نتائج المخزون:

النقر يفتح صفحة العميل مع علامة التبويب المخزون نشطة. إذا كان المخزون غير مخصص (لا يوجد `customer_id`)، فإنه يرتبط بقائمة المخزون الرئيسية بدلاً من ذلك.

نتائج الخدمات:

النقر يفتح صفحة العميل مع علامة التبويب الخدمات نشطة، مبرزًا الخدمة المحددة.

التنقل من النتائج

جميع نتائج البحث هي **روابط قابلة للنقر** تنقل مباشرة إلى الصفحة ذات الصلة:

نمط الرابط:

- `/customers/{customer_id}` - سجلات العملاء
- `/customers/{customer_id}#4` - جهات الاتصال (علامة التبويب 4)
- `/customers/{customer_id}#2` - المواقع (علامة التبويب 2)
- `/customers/{customer_id}#8` - المخزون (علامة التبويب 8)
- `/customers/{customer_id}#3` - الخدمات (علامة التبويب 3)
- `/inventory-items-list` - مخزون غير مخصص

تقوم علامة التجزئة (`#`) تلقائيًا بتحديد علامة التبويب الصحيحة عند تحميل صفحة العميل.

تقسيم الصفحات

تُقسم النتائج إلى صفحات تحتوي على 10 عناصر لكل صفحة:

```
عرض النتائج 11-20 من 47
```

انتقل عبر الصفحات باستخدام:

- أزرار **السابق/التالي**
- **أرقام الصفحات** - انقر على الصفحة المحددة
- **لوحة المفاتيح** - الأسهم اليسرى/اليمنى (إذا تم تنفيذها)

سيناريوهات البحث الشائعة

السيناريو 1: العثور على عميل بواسطة الهاتف

يتصل المستخدم، ويقدم رقم الهاتف.

> النتائج: • John Smith (جهة اتصال) - 0123-555 - الهاتف المحمول • معرف العميل: 123
> (خدمة) - معرف العميل: 123

انقر على أي من النتيجتين للوصول إلى حساب العميل.

السيناريو 2: تحديد بطاقة SIM

يحتاج الفني إلى معرفة أي عميل لديه بطاقة SIM معينة.

> النتائج: • 89445380000000001234 (مخزون) - معرف العميل: 456

انقر على النتيجة لرؤية تعيين SIM، تفاصيل العميل.

السيناريو 3: العثور على عميل غير نشط

تحتاج إلى تحديد موقع عميل أغلق حسابه منذ 6 أشهر.

> ☑ تضمين الحسابات المغلقة
>
> النتائج: • Acme Corporation (عميل) - 789 :معرف العميل

السيناريو 4: البحث بواسطة البريد الإلكتروني

يرسل العميل بريدًا إلكت��ونيًا إلى الدعم، يحتاج الموظف إلى العثور على حسابه.

> النتائج: • John Smith (جهة اتصال) - 123 :معرف العميل

السيناريو 5: العثور على خدمة بواسطة UUID

تظهر سجلات التزويد UUID الخدمة، تحتاج إلى معرفة أي عميل.

> النتائج: • الهاتف المحمول - +44 7700 900123 (خدمة) - معرف العميل: 456

نصائح البحث

لأفضل النتائج:

- **استخدم مصطلحات جزئية** - "Smith" أفضل من "John Smith" للحصول على نتائج أوسع
- **جرّب التباينات** - إذا لم يعمل "John"، جرب الهاتف أو البريد الإلكتروني
- **تضمين الحسابات المغلقة** - عند البحث عن البيانات التاريخية
- **كن محددًا للمعدات** - استخدم أرقام التسلسل الكاملة للمخزون
- **ابحث عن UUID الخدمة** - عندما لا تكون المعرفات الأخرى معروفة

ما الذي يتم البحث عنه (حسب الكيان):

العملاء:

- اسم العميل فقط (ليس العنوان، الملاحظات، أو الحقول الأخرى)

جهات الاتصال:

- الاسم الأول
- الاسم الأخير
- عنوان البريد الإلكتروني
- رقم الهاتف

المواقع:

- اسم الموقع فقط

المخزون:

- itemtext1 (عادةً ICCID، عنوان، رقم التسلسل MAC)
- itemtext2 (عادةً IMSI، معرف ثانوي)
- *ملاحظة: لا يتم البحث في itemtext3-20 أو ملاحظات المخزون*

الخدمات:

- اسم الخدمة
- UUID الخدمة

ما لا يتم البحث عنه:

- عناوين العملاء
- ملاحظات العملاء
- أوصاف المعاملات
- تفاصيل الفواتير
- سجلات التزويد
- إدخالات سجل النشاط
- ملاحظات المخزون (بخلاف itemtext1/2)

مرجع API

نقطة نهاية البحث العالمي

```
GET /utilities/search_everything?
search=Smith&page=1&per_page=10&search_closed_records=false
Authorization: Bearer <token>
```

معلمات الاستعلام:

- `search` (مطلوب) - مصطلح البحث
- `page` (اختياري) - رقم الصفحة (افتراضي: 1)
- `per_page` (اختياري) - النتائج لكل صفحة (افتراضي: 10)
- `search_closed_records` (اختياري) - تضمين الحسابات المغلقة (افتراضي: false)

الاستج��بة:

```
{
  "data": [
    {
      "id": 123,
      "name": "John Smith",
      "customer_id": 123,
      "type": "customer"
    },
    {
      "id": 456,
      "name": "John Smith",
      "customer_id": 123,
      "type": "contact"
    },
    {
      "id": 789,
      "name": "Mobile - +44 7700 900123",
      "customer_id": 123,
      "type": "service"
    }
  ],
  "pagination": {
    "current_page": 1,
    "per_page": 10,
    "total_pages": 5,
    "total_items": 47
  }
}
```


منطق البحث (الجهة الخلفية):

تقوم الجهة الخلفية بإجراء اتحاد SQL عبر جميع جداول الكيانات:

```sql
-- العملاء
SELECT customer_id AS id,
       customer_name AS name,
       customer_id,
       'customer' AS type
FROM customer
WHERE customer_name LIKE '%Smith%'
  AND customer_status = 'Open'

UNION ALL

-- جهات الاتصال
SELECT contact_id AS id,
       CONCAT(contact_firstname, ' ', contact_lastname) AS name,
       customer_id,
       'contact' AS type
FROM customer_contact
WHERE (contact_firstname LIKE '%Smith%' OR
       contact_lastname LIKE '%Smith%' OR
       contact_email LIKE '%Smith%' OR
       contact_phone LIKE '%Smith%')

UNION ALL

-- المواقع
SELECT site_id AS id,
       site_name AS name,
       customer_id,
       'site' AS type
FROM customer_site
WHERE site_name LIKE '%Smith%'

UNION ALL

-- المخزون
SELECT inventory_id AS id,
       itemtext1 AS name,
       customer_id,
       'inventory' AS type
FROM inventory
WHERE itemtext1 LIKE '%Smith%' OR
      itemtext2 LIKE '%Smith%'
```

```
UNION ALL

-- الخدمات
SELECT service_id AS id,
       service_name AS name,
       customer_id,
       'service' AS type
FROM customer_service
WHERE service_name LIKE '%Smith%' OR
      service_uuid LIKE '%Smith%'
```

ثم يتم تقسيم النتائج وإعادتها.

اعتبارات الأداء

أداء البحث:

- تستخدم عمليات البحث استعلامات LIKE مع أحرف البدل (`%term%`)
- لا يوجد فهرس نص كامل تم تنفيذه حاليًا
- عمليات بحث أبطأ (عميل >100k) قد تواجه قواعد البيانات الكبيرة
- يتم تحديد النتائج بـ 10 لكل صفحة لأغراض الأداء

نصائح التحسين:

- كن محددًا في مصطلحات البحث لتقليل مجموعة النتائج
- استخدم فلتر الحسابات المغلقة لتقليل نطاق البحث
- ضع في اعتبارك إضافة فهارس قاعدة ا��بيانات على الحقول التي يتم البحث عنها بشكل متكرر

استكشاف الأخطاء وإصلاحها

لم يتم العثور على نتائج (لكن السجل موجود)

- **السبب:** مصطلح البحث لا يتطابق مع تنسيق البيانات المخزنة
- **أمثلة:**
  - الهاتف مخزن كـ "+44 7700 900123"، البحث عن "07700900123" لن يتطابق

    - البحث، "<john.smith@example.com>" البريد الإلكتروني مخزن كـ لن يتطابق "john" عن
- **الحل:** جرّب التباينات، استخدم المطابقات الجزئية التي توجد بالتأكيد

البحث بطيء جدًا

- **السبب:** قاعدة بيانات كبيرة، استعلام معقد عبر جداول متعددة
- **الحل:**
    - استخدم مصطلحات بحث أكثر تحديدًا
    - حصر البحث على الحسابات المفتوحة فقط (إلغاء تحديد الحسابات المغلقة)
    - اتصل بالمسؤول حول فهرسة قاعدة البيانات

النتائج ترتبط بعميل خاطئ

- **السبب:** وجود عملاء/جهات اتصال متعددة بنفس الاسم
- **الحل:** استخدم معرف العميل للتفريق، ��و ابحث بواسطة معرف فريد (البريد الإلكتروني، الهاتف)

الحسابات المغلقة لا تظهر

- **السبب:** مربع "تضمين الحسابات المغلقة" غير محدد
- **الحل:** تحقق من المربع وابحث مرة أخرى

الوثائق ذات الصلة

- `basics_customers` - إدارة العملاء
- `basics_navigation` - التنقل العام
- `administration_inventory` - عمليات البحث في المخزون

نظام الشحن وإعادة التعبئة

يوفر نظام الشحن في OmniCRM للعملاء **بوابة إعادة تعبئة مسبقة الدفع للخدمة الذاتية** لإضافة رصيد أو تمديد صلاحية الخدمة عبر بوابة الخدمة الذاتية. تُستخدم هذه الميزة عادةً لـ:

- بطاقات SIM مسبقة الدفع وخدمات البيانات فقط - **خدمات البيانات المحمولة**
- أجهزة واي فاي المحمولة وأجهزة الإنترنت المحمولة - **خدمات النقاط الساخنة**
- الوصول إلى الإنترنت مسبق الدفع - **خدمات الإنترنت اللاسلكي الثابت**

لمعلومات حول توفير الخدمات المحمولة وإدارة مخزون بطاقات SIM انظر أيضًا: توفير بطاقة SIM.

نظرة عامة

يسمح نظام الشحن للعملاء بشراء أيام إضافية من الخدمة من خلال عملية دفع مبسطة متعددة الخطوات مع معالجة دفع Stripe المدمجة.

الميزات الرئيسية:

- بوابة خدمة ذاتية للعملاء (لا حاجة لتدخل الموظفين)
- اختيار مدة مرنة (1-30 يومًا)
- عرض الاستخدام في الوقت الحقيقي قبل الشراء
- معالجة دفع آمنة مدعومة من Stripe
- استرداد تلقائي إذا فشل الشحن
- إنشاء الفواتير والمعاملات
- تكامل نظام التوفير لتفعيل الخدمة

الوصول إلى بوابة الشحن

يتم الوصول إلى بوابة الشحن عبر **رابط عام** يمكن للعملاء زيارته دون تسجيل الدخول إلى نظام CRM:

كيف يصل العملاء إليها:

- رابط مباشر يُرسل عبر الرسائل النصية القصيرة عند انخفاض الرصيد
- رمز QR على المواد المطبوعة
- رابط على بوابة الخدمة الذاتية
- مشاركة عبر دعم العملاء

تكتشف البوابة تلقائيًا خدمة العميل بناءً على عنوان IP أو IMSI الخاص بالطلب.

عملية الشحن

تتكون عملية الشحن من **4 خطوات**:

الخطوة 1: اختيار البيانات

يختار العملاء عدد الأيام من الخدمة التي يرغبون في شرائها.

واجهة المستخدم:

- **تحكم منزلق** - اختر من 1 إلى 30 يومًا
- **حساب السعر المباشر** - يظهر التكلفة الإجمالية بناءً على الاختيار
- **عرض تاريخ انتهاء الصلاحية** - يحسب ويظهر متى ستنتهي الخدمة
- **عرض الاستخدام الحالي** - يظهر الرصيد المتبقي/تاريخ الانتهاء قبل الشحن

عرض المثال:

تكوين الأسعار:

- يتم تكوين السعر لكل يوم عبر متغير البيئة `VITE_TOPUP_PRICE_PER_DAY`
- الافتراضي: 10 دولارات أمريكية لكل يوم
- يتم تعيين العملة عبر `VITE_CURRENCY_CODE`

الخطوة 2: معلومات الفوترة

يقدم العملاء تفاصيل الاتصال الخاصة بهم للمعاملة:

- **الاسم الأول**
- **اسم العائلة**

- **عنوان البريد الإلكتروني**

تُستخدم هذه المعلومات لـ:

- إنشاء الفاتورة
- بريد إلكتروني لإيصال الدفع
- سجلات المعاملات
- معالجة الاسترداد (إذا لزم الأمر)

الخطوة 3: الدفع

نماذج الدفع المستضافة من قبل البائع معالجة الدفع الآمنة عبر (Stripe Elements، PayPal SDK).

طرق الدفع المدعومة:

- بطاقات الائتمان (Visa، Mastercard، Amex)
- بطاقات الخصم
- المحافظ الرقمية (Apple Pay، Google Pay، PayPal) *إذا تم تمكينها من قبل بائع الدفع*

ميزات الأمان:

- تكامل بائع الدفع متوافق مع PCI
- لا يتم تخزين تفاصيل البطاقة في OmniCRM
- دعم المصادقة 3D Secure
- نقل الدفع مشفر

تدفق الدفع:

1. عرض نموذج الدفع الآمن مع إدخال البطاقة
2. يدخل العميل تفاصيل الدفع
3. تتم معالجة الدفع عبر البائع المكون
4. يتم خصم البطاقة على الفور
5. يتم التعامل مع نجاح/فشل الدفع

ملاحظة

> يقوم النظام، (غير متاح OCS، مثل، خطأ في الشبكة) إذا نجح الدفع ولكن فشل توفير الشحن تلقائيًا ببدء **استرداد كامل** لطريقة دفع العميل.

الخطوة 4: الإكمال

شاشة النجاح:

> تم تمديد خدمتك. تاريخ انتهاء الصلاحية الجديد: 17 يناير 2025
>
> معرف المعاملة: TXN-123456 تم إرسال الإيصال إلى: <customer@example.com>

شاشة الفشل:

إذا فشل الشحن، يعرض النظام خطأ ويقوم تلقائيًا بمعالجة استرداد:

> لم نتمكن من إكمال الشحن الخاص بك. تم استرداد دفعتك.
>
> خطأ: غير قادر على الاتصال بنظام الفوترة
>
> يرجى المحاولة مرة أخرى أو الاتصال بالدعم.

معالجة الخلفية

عند إكمال العميل للدفع، يحدث ما يلي تلقائيًا:

1. التحقق من الدفع

يتحقق النظام من:

- حالة نية الدفع هي `succeeded`
- مبلغ الدفع يتطابق مع الأيام المحددة (`days × price_per_day`)
- لم تتم معالجة نية الدفع من قبل (يمنع الشحن المزدوج)

2. عملية الشحن

```
- API endpoint: POST /oam/topup_dongle
- يتحقق من service_uuid و IMSI
- يستدعي OCS/CGRateS لإضافة الرصيد
- ينشئ وظيفة توفير (play_topup_hotspot)
```

3. إنشاء السجلات

ينشئ النظام سجلات قاعدة بيانات متعددة:

- **سجل HotspotTopup** - يتتبع معاملة الشحن
  - payment_intent_id
  - service_uuid
  - imsi
  - الأيام المشتراة
  - topup_amount
  - الحالة (نجاح/فشل/مسترد)
- **سجل المعاملة** - معاملة مالية
  - العنوان: "شحن النقاط الساخنة - 7 أيام"
  - المبلغ: topup_amount (إيجابي)
  - مرتبط بـ customer_id و service_id
- **سجل الفاتورة** - فاتورة الدفع
  - تحتوي على معاملة الشحن
  - مُعلمة كمدفوعة على الفور
  - مرجع الدفع: Stripe payment_intent_id
- **معاملة الدفع** - معاملة ائتمانية معاكسة
  - العنوان: "الدفع لـ [عنوان الفاتورة]"
  - المبلغ: topup_amount (ائتمان - سلبي)
  - يربط دفع الفاتورة بحساب العميل

4. وظيفة التوفير

يتم إنشاء وظيفة توفير مع playbook `play_topup_hotspot` التي:

- تتصل بواجهة برمجة التطبيقات OCS/CGRateS
- تضيف رصيدًا إلى الحساب
- تمدد تاريخ انتهاء الصلاحية

- تنشئ إدخال سجل النشاط
- ترسل إشعار تأكيد (إذا تم تكوينه)

تنتظر واجهة برمجة التطبيقات حتى تكتمل عملية التوفير (الاستطلاع بفواصل زمنية قدرها 0.2 ثانية، بحد أقصى 25 تكرارًا) قبل إرجاع النجاح للعميل.

5. استرداد تلقائي عند الفشل

إذا فشل أي خطوة بعد الدفع:

```
if topup_provisioning_failed:
    refund = stripe.Refund.create(
        payment_intent=payment_intent_id,
        reason='requested_by_customer'  # استرداد تلقائي من النظام
    )
    status_message = "فشل الشحن. يتم استرداد الدفع..."
```

يتم معالجة الاسترداد تلقائيًا ويتم إبلاغ العميل على الشاشة.

نقاط نهاية واجهة برمجة التطبيقات

نقطة نهاية الشحن

```
POST /oam/topup_dongle
Content-Type: application/json

{
  "service_uuid": "123e4567-e89b-12d3-a456-426614174000",
  "imsi": "310120123456789",
  "days": 7,
  "payment_intent_id": "pi_1234567890abcdef",
  "topup_amount": 70.00
}
```

الاستجابة (نجاح):

```
{
  "result": "OK",
  "status": 200,
  "provision_id": 456,
  "payment_intent_id": "pi_1234567890abcdef",
  "service_uuid": "123e4567-e89b-12d3-a456-426614174000",
  "invoice_id": 789
}
```

الاستجابة (فشل):

```
{
  "result": "Failed",
  "Reason": "OCS connection timeout",
  "status": 500
}
```

تحقق من صحة:

- جميع الحقول المطلوبة موجودة (service_uuid، imsi، days، payment_intent_id، topup_amount)
- يتطابق topup_amount مع الأيام: `topup_amount × 100 == days × 1000` (بالسنتات)
- توجد نية الدفع في Stripe
- يتطابق مبلغ نية الدفع مع: `payment_intent.amount == topup_amount × 100`
- حالة نية الدفع هي `succeeded`
- لم تتم معالجة نية الدفع بالفعل (يتحقق من جدول `HotspotTopup`)

نقطة نهاية الاستخدام

يسترجع معلومات الاستخدام الحالية وخدمة العميل:

```
GET /oam/usage
```

الاستجابة:

```
{
  "imsi": "310120123456789",
  "service": {
    "service_uuid": "123e4567-e89b-12d3-a456-426614174000",
    "service_name": "0412345678 - بيانات الهاتف المحمول",
    "service_status": "نشط"
  },
  "balance": {
    "expiry": "2025-01-10T23:59:59Z",
    "unlimited": true
  },
  "requestingIp": "203.0.113.45"
}
```

تستخدم هذه النقطة عنوان IP تلقائيًا لتحديد خدمة العميل المطلوب.

التكوين

متغيرات البيئة

قم بتكوين هذه في ملف OmniCRM-UI `.env`:

```
# تكوين بوابة الشحن
VITE_TOPUP_PRICE_PER_DAY=10
VITE_CURRENCY_CODE=AUD
VITE_SELF_CARE_NAME="YourCompany"

# تكوين الدفع
VITE_PAYMENT_VENDORS="stripe"
VITE_STRIPE_PUBLISHABLE_KEY=pk_live_...
```

تكوين Stripe

يستخدم نظام الشحن نوايا الدفع من Stripe:

1. في لوحة معلومات Stripe الخاصة بك **تفعيل نوايا الدفع**
2. **تكوين Webhook** (اختياري ولكن موصى به) لاستقبال تحديثات حالة الدفع

3. **إعداد طرق الدفع** (البطاقات، المحافظ، إلخ)
4. **وضع الاختبار** - استخدم مفاتيح الاختبار للتطوير

```
# التطوير
VITE_STRIPE_PUBLISHABLE_KEY=pk_test_...

# الإنتاج
VITE_STRIPE_PUBLISHABLE_KEY=pk_live_...
```

تكوين playbook

يجب تكوين playbook التوفير `play_topup_hotspot.yaml` لـ:

- قبول متغير `days`
- الاتصال بواجهة برمجة التطبيقات OCS/CGRateS
- إضافة رصيد إلى الحساب
- تحديث تاريخ انتهاء الخدمة

هيكل playbook المثال:

```yaml
- name: شحن خدمة النقاط الساخنة
  hosts: localhost
  tasks:
    - name: إضافة رصيد إلى OCS
      uri:
        url: "{{ ocs_api_url }}/add_balance"
        method: POST
        body:
          imsi: "{{ imsi }}"
          days: "{{ days }}"
          service_uuid: "{{ service_uuid }}"
```

إشعارات الرصيد المنخفض

يمكن للنظام إرسال إشعارات تلقائية عندما يكون رصيد العميل منخفضًا:

إشعارات SMS:

عند تنشيطها بواسطة أحداث OCS (`Action_Balance_Low`, `Action_Balance_Out`, `Action_Balance_Expired`):

إشعارات البريد الإلكتروني:

تم تكوينها في خطط العمل OCS/CGRateS لإرسال تنبيهات الرصيد.

محفزات الإشعارات:

- `Action_Balance_Low` - الرصيد أقل من العتبة (مثل، تبقى 2 يوم)
- `Action_Balance_Out` - الرصيد مستنفد
- `Action_Balance_Expired` - انتهت الخدمة

تتضمن كل إشعار رابط بوابة الشحن لسهولة وصول العملاء.

استكشاف الأخطاء وإصلاحها

المشكلات ��لشائعة

"نظام الدفع غير متاح"

- **السبب:** فشل تحميل مكتبة Stripe أو مفتاح قابل للنشر غير صالح
- **الإصلاح:**
  - تحقق من أن `VITE_STRIPE_PUBLISHABLE_KEY` تم تعيينه بشكل صحيح
  - تحقق من أن حساب Stripe نشط
  - تحقق من وحدة تحكم المتصفح بحثًا عن أخطاء JavaScript

"فشل الشحن. يتم استرداد الدفع..."

- **السبب:** فشل وظيفة التوفير (OCS غير متاح، خطأ في playbook، إلخ.)
- **الإصلاح:**
  - تحقق من سجلات التوفير: `GET /crm/provision/provision_id/<id>`
  - تحقق من أن واجهة برمجة التطبيقات OCS/CGRateS قابلة للوصول
  - راجع playbook `play_topup_hotspot.yaml` بحثًا عن أخطاء
  - تحقق من سجلات Ansible

"تمت معالجة نية الدفع بالفعل"

- **السبب:** يحاول العميل إعادة استخدام نفس الدفع (مثل، تحديث بعد النجاح)
- **الإصلاح:** هذا يعمل كما هو مصمم لمنع الفوترة المزدوجة. يجب على العميل بدء شحن جديد إذا لزم الأمر.

"مبلغ نية الدفع لا يتطابق"

- **السبب:** عدم تطابق بين حساب واجهة المستخدم والتحقق من الخلفية
- **الإصلاح:**
  - تحقق من أن `VITE_TOPUP_PRICE_PER_DAY` يتطابق مع توقعات الخلفية (افتراضي 10 دولارات)
  - تحقق من أن تكوين العملة متسق
  - امسح ذاكرة التخزين المؤقت للمتصفح وأعد المحاولة

مراقبة الشحنات

عرض سجلات الشحن:

استعلام عن جدول `HotspotTopup` لرؤية جميع محاولات الشحن:

```
SELECT
  hotspot_topup_id,
  service_uuid,
  days,
  topup_amount,
  status,
  payment_intent_id,
  created
FROM hotspot_topup
WHERE status = 'Failed'
ORDER BY created DESC;
```

تحقق من حالة التوفير:

```
GET /crm/provision/provision_id/<provision_id>
```

يعرض الحالة التفصيلية لوظيفة توفير الشحن.

لوحة معلومات Stripe:

راقب المدفوعات، والاستردادات، والمعاملات الفاشلة في لوحة معلومات Stripe الخاصة بك على <https://dashboard.stripe.com>

اعتبارات الأمان

أمان الدفع:

- يتم التعامل مع جميع بيانات البطاقة بواسطة Stripe (متوافق مع PCI 1 المستوى)
- لا يتم تخزين بيانات الدفع الحساسة في قاعدة بيانات OmniCRM
- تمنع نوايا الدفع الرسوم غير المصرح بها
- تحقق من المبلغ على كل من جانب العميل والخادم

منع الاحتيال:

- اكتشاف نية الدفع المكررة يمنع الفوترة المزدوجة
- تتبع عنوان IP لتوافق الاستخدام
- يضمن التحقق من IMSI أن الشحن يذهب إلى الخدمة الصحيحة
- تحد الاستردادات التلقائية من التعرض المالي

التحكم في الوصول:

- بوابة الشحن عامة (حسب التصميم - يحتاج العملاء إلى الوصول)
- تتطلب نقطة نهاية الاستخدام تحديد خدمة صالحة (IP أو IMSI)
- يمنع التحقق من الخلفية الشحنات العشوائية للخدمات
- يمكن للمسؤول عرض جميع سجلات الشحن عبر واجهة CRM

أفضل الممارسات

للمشغلين:

1. **اختبار تدفق الاسترداد** - اختبار سيناريوهات الفشل بانتظام لضمان عمل الاستردادات
2. **مراقبة الشحنات الفاشلة** - إعداد تنبيهات لمعدلات الفشل العالية

3. **الحفاظ على بساطة playbooks** - يجب أن تكون playbooks الشحن سريعة وموثوقة
4. **التحقق من الاتصال بـ OCS** - التأكد من أن واجهة برمجة التطبيقات OCS متاحة دائمًا
5. **مراجعة التسعير** - تحديث `VITE_TOPUP_PRICE_PER_DAY` حسب الحاجة

للعملاء:

1. **إضافة رابط الشحن إلى المفضلة** - وصول سريع عند الحاجة
2. **حفظ إشعارات الرصيد المنخفض** - تحتوي الرسائل النصية القصيرة على رابط مباشر
3. **تحديث البريد الإلكتروني** - يتم إرسال الإيصالات إلى البريد الإلكتروني المسجل
4. **التحقق من انتهاء الصلاحية قبل السفر** - شحن قبل مغادرة منطقة التغطية

للمطورين:

1. **معالجة Webhooks Stripe** - تنفيذ معالجات Webhook لتحديثات حالة الدفع
2. **تنفيذ idempotency** - تحقق دائمًا من payment_intent_id قبل المعالجة
3. **تسجيل بشكل موسع** - تحتاج فشل الشحن إلى معلومات استكشاف الأخطاء التفصيلية
4. **اختبار مسارات الأخطاء** - تحقق من أن أتمتة الاسترداد تعمل بشكل صحيح
5. **مراقبة الأداء** - يجب أن تكتمل استطلاعات التوفير في <5 ثوانٍ

الوثائق ذات الصلة

- `payments_process` - معالجة الدفع العامة
- `concepts_provisioning` - نظرة عامة على نظام التوفير
- دليل نظام الدفع - تفاصيل تكامل بائع الدفع
- `payments_transaction` - إدارة المعاملات
- `payments_invoices` - معالجة الفواتير

مسرد المصطلحات

يتم سرد المصطلحات بالترتيب الأبجدي. تعريفات المصطلحات المستخدمة في وثائق OmniCRM. حيثما يكون للمفهوم دليل خاص به، يتم تضمين رابط إلى الصفحة ذات الصلة.

A

Addon - ميزة إضافية اختيارية مرتبطة بخدمة موجودة، مثل معزز البيانات أو ميزة ذات قيمة مضافة. يتم فوترتها جنبًا إلى جنب مع الخدمة الأم بدلاً من بيعها كمنتج مستقل.

Ansible - محرك الأتمتة الذي تستخدمه OmniCRM لتوفير الخدمات. تؤدي تعريفات المنتجات التي تنفذ التكوين الأساسي. انظر التوفير القائم على Ansible إلى تشغيل دفاتر التشغيل الخاصة بـ Ansible.

AuC (مركز المصادقة) - الوظيفة التي تخزن المواد السرية لمفتاح SIM وتحسب متجهات المصادقة المستخدمة للتحقق من المشترك على الشبكة المحمولة. في OmniCRM، يتم ملء AuC كجزء من توفير SIM إلى HSS.

C

CGRateS - محرك التصنيف والفوترة في الوقت الحقيقي المدمج مع OmniCRM. يقوم بتصنيف الاستخدام (الصوت، الرسائل القصيرة، البيانات) مقابل الوجهات ويطبق الأرصدة، ويتم تكوينه من خلال ملفات تعريف السمات، ومرشحات الحساب، وأسعار الوجهات.

Cell Broadcast - قدرة الرسائل من واحد إلى العديد تُستخدم لدفع التنبيهات الطارئة والمعلومات العامة إلى جميع الأجهزة في منطقة جغرافية محددة. انظر نظام البث الخلوي.

E

eSIM - بطاقة SIM مدمجة يتم توفيرها عبر الهواء بدلاً من إدخال بطاقة فعلية. تدعم OmniCRM إصدار eSIM والاستيراد. انظر توفير بطاقة SIM.

H

HSS (خادم المشتركين الرئيسي) - قاعدة بيانات المشتركين المركزية لشبكة النواة المحمولة. تحتوي على بيانات الاشتراك والمصادقة وتستجيب للطلبات من الشبكة عندما يتصل مشترك. تقوم OmniCRM بكتابة سجلات المشتركين إلى HSS أثناء توفير SIM وتدعم عدة خلفيات HSS.

I

IMS (نظام الوسائط المتعددة عبر بروتوكول الإنترنت) - الإطار الأساسي للشبكة الذي يتم توفير المشتركين في IP. والوسائط المتعددة المعتمدة على (VoLTE) يقدم خدمات الصوت بحيث تعمل خدمات الصوت HSS جنبًا إلى جنب مع IMS.

IMSI (معرف المشترك الدولي للهاتف المحمول) - المعرف الفريد المرتبط بـ SIM الذي تستخدمه الشبكة لتحديد المشترك. تقوم OmniCRM بربط IMSI أثناء التوفير بحساب الفوترة.

Inventory - مجموعة الموارد المادية والمنطقية (بطاقات SIM، أرقام الهواتف المحمولة، التي يمكن تعيينها للعملاء. تنتقل العناصر عبر حالات محددة أثناء تخزينها، وتعيينها، (والمعدات وتفعيلها. انظر إدارة المخزون.

M

MSISDN (رقم ISDN لل��حطة الهاتف المحمول) - رقم الهاتف القابل للاتصال المرتبط بمشترك. IMSI لـ SIM هو الهوية الشبكية، بينما MSISDN هو الرقم الذي يتصل به الآخرون.

O

OCS (نظام الفوترة عبر الإنترنت) - النظام المسؤول عن التحكم في الائتمان في الوقت الحقيقي، وتفويض الفوترة وفرض الرسوم عند حدوثها. يتيح تكامل تطبيق OmniCRM لـ OCS الأرصدة ومنع الإنفاق الزائد على الخدمات المدفوعة مسبقًا.

P

Playbook - روتين أتمتة أنسيبل مسمى يقوم بأداء مهمة توفير، مثل تنشيط SIM أو تكوين معدات العملاء. يتم تشغيل دفتر التشغيل الخاص بتوفير المنتج عند إنشاء الخدمة المرتبطة. انظر دفاتر تشغيل Ansible.

PLMN (شبكة الهاتف المحمول العامة) - شبكة مشغل الهاتف المحمول، يتم تحديدها بواسطة رمز. تستخدم فوترة البيانات وجهات PLMN (الشبكة المتصل بها العميل)، بينما تستخدم خدمات الصوت والرسائل القصيرة وجهات جغرافية.

Product - تعريف قابل للبيع في الكتالوج. يحدد المنتج التسعير، وسلوك التوفير، والخدمة (الخدمات) التي ينشئها عند الشراء. انظر المنتجات والخدمات.

Proforma Invoice - فاتورة أولية تصدر قبل استحقاق الدفع أو قبل إتمام البيع. توضح ما سيتم فرضه ولكنها ليست في حد ذاتها طلبًا للدفع. انظر الفواتير.

Provisioning - عملية تحويل منتج تم شراؤه إلى خدمة عاملة من خلال تنفيذ التكوينات المطلوبة للشبكة والنظام، عادةً من خلال دفاتر تشغيل Ansible. انظر التوفير.

R

RBAC (التحكم في الوصول القائم على الدور) - نموذج الأذونات الذي يحكم ما يمكن لكل مستخدم رؤيته وفعله في OmniCRM. يتم تجميع الأذونات في أدوار، ويتم تعيين الأدوار للمستخدمين. انظر التحكم في الوصول القائم على الدور.

S

Self-Care Portal - البوابة الموجهة للعملاء حي�ث يمكن للمشتركين عرض حساباتهم، واستخدامهم، وفواتيرهم، وخدماتهم دون الحاجة إلى الاتصال بالدعم. انظر بوابة الخدمة الذاتية.

Self-Signup - تدفق يسمح لعميل جديد بالتسجيل وتوفير خدمة بنفسه، بما في ذلك توفير SIM، دون تدخل الموظفين.

Service - حالة نشطة ومزودة من منتج على حساب العميل. يؤدي شراء منتج إلى إنشاء خدمة واحدة أو أكثر، يمكن إدارة كل منها أو تعديلها أو تعليقها. انظر المنتجات والخدمات.

SIM (وحدة تعريف المشترك) - البطاقة أو الوحدة المدمجة التي تحدد مشتركًا لشبكة الهاتف المحمول وتخزن مفتاح المصادقة. انظر توفير بطاقة SIM.

T

Tag - علامة تُطبق على العملاء أو السجلات لتجميعها وتصفيةها، على سبيل المثال حسب الشريحة أو الحملة. انظر العلامات.

Transaction - حركة مالية واحدة على حساب، ��ثل رسوم، أو دفع، أو ائتمان، أو استرداد. تشكل المعاملات تاريخ الفوترة للحساب. انظر المعاملات.

Numbers

2FA (المصادقة الثنائية) - خطوة تسجيل دخول إضافية تتطلب رمزًا لمرة واحدة بجانب كلمة المرور، مما يحمي حسابات الموظفين من بيانات الاعتماد المساومة. انظر المصادقة الثنائية.

دليل دورة حياة المنتج الشاملة

يوفر هذا الدليل جولة شاملة عبر دورة حياة المنتج في OmniCRM، بدءًا من إنشاء تعريف المنتج مرورًا بتوفير الخدمات، وإضافة الملحقات، وإلغاء التوفير. سنغطي استراتيجية التسعير، وتكامل Ansible، وسنقدم أمثلة من العالم الحقيقي طوال الوقت.

نظرة عامة: رحلة المنتج إلى الخدمة

تتبع دورة حياة المنتج في OmniCRM هذه المراحل:

1. **تعريف المنتج** - يقوم المسؤول بإنشاء قالب المنتج مع قواعد التسعير والتوفير
2. **إنشاء الخدمة** - يقوم العميل بطلب المنتج، ويقوم النظام بتوفير مثيل الخدمة
3. **دورة حياة الخدمة** - يستخدم العميل الخدمة، ويضيف الملحقات/المدفوعات الإضافية، ويعدل الخدمة
4. **إلغاء التوفير** - يتم إنهاء الخدمة، ويتم تحرير الموارد

فه�? التسعير: الجملة مقابل التجزئة

كل منتج وخدمة في OmniCRM له بعدان تسعيريان: **الجملة** و **التجزئة**.

تكلفة الجملة

تمثل تكلفة الجملة التكلفة الفعلية لتقديم الخدمة:

- تكاليف البنية التحتية والنطاق الترددي
- رسوم الترخيص
- تكاليف المعدات
- النفقات التشغيلية

تكلفة التجزئة

تكلفة التجزئة هي المبلغ المفروض على العميل.

تكاليف الإعداد

تحتوي كل من الجملة والتجزئة على متغيرات تكلفة إعداد لرسوم التوفير لمرة واحدة:

- `wholesale_setup_cost` - تكلفتك للتوفير
- `retail_setup_cost` - المبلغ المفروض على العميل للتفعيل

مثال:

```
{
  "retail_cost": 15.00,
  "wholesale_cost": 5.00,
  "retail_setup_cost": 0.00,
  "wholesale_setup_cost": 1.00
}
```

المرحلة 1: إنشاء تعريف المنتج

المنتجات هي قوالب تحدد ما يتم توفيره وكيف يتم فرض الرسوم على العملاء.

إنشاء منتج بطاقة SIM موبايل

إنشاء منتج بطاقة SIM موبايل مسبقة الدفع مع 20 جيجابايت من البيانات شهريًا.

الخطوة 1: الانتقال إلى إدارة المنتجات

المنتجات ← **إنشاء منتج** من واجهة المستخدم الإدارية، انتقل إلى.

الخطوة 2: تعريف المعلومات الأساسية

```
{
  "product_name": "Prepaid Mobile 20GB",
  "product_slug": "prepaid-mobile-20gb",
  "category": "standalone",
  "service_type": "mobile",
  "enabled": true,
  "icon": "fa-solid fa-sim-card",
  "comment": "Prepaid mobile SIM with 20GB data, unlimited calls & texts"
}
```

تفسيرات الحقول:

- `product_name` - الاسم الذي يواجه العميل والذي يظهر في الكتالوج
- `product_slug` - معرف آمن للعنوان يستخدم في استدعاءات API والروابط
- `category` - "standalone" (مقابل ملحق/حزمة) تعني أن هذا ينشئ خدمة جديدة
- `service_type` - يجمع المنتجات ذات الصلة، ويستخدم لتصفية الملحقات
- `enabled` - يجب أن يكون صحيحًا ليكون المنتج قابلًا للطلب
- `icon` - أيقونة FontAwesome المعروضة في واجهة المستخدم
- `comment` - ملاحظات داخلي�ة للرجوع إليها من قبل الموظفين

الخطوة 3: تعيين الأسعار

```
{
  "retail_cost": 15.00,
  "wholesale_cost": 5.00,
  "retail_setup_cost": 0.00,
  "wholesale_setup_cost": 1.00,
  "contract_days": 30
}
```

تفصيل التسعير:

- الإيرادات الشهرية لكل عميل: 15.00£
- التكلفة الشهرية للتقديم: 5.00£
- هامش الربح الشهري: 10.00£ (200% زيادة، 67% هامش)
- ربح الإعداد: -1.00£ (مدعوم لجذب العملاء)

- مدة العقد: 30 يومًا (تجديد شهري)

الخطوة 4: تحديد أهلية العميل

```
{
  "residential": true,
  "business": false,
  "customer_can_purchase": true,
  "available_from": "2025-01-01T00:00:00Z",
  "available_until": null
}
```

- يمكن للعملاء السكنيين الطلب
- لا يمكن للعملاء التجاريين (خط إنتاج مختلف)
- تمكين الشراء الذاتي
- متاح من 1 يناير 2025 فصاعدًا
- لا يوجد تاريخ انتهاء (عرض مستمر)

الخطوة 5: تكوين التجديد التلقائي

```
{
  "auto_renew": "prompt",
  "allow_auto_renew": true
}
```

- `"prompt"` - اسأل العميل إذا ��انوا يريدون التجديد التلقائي عند الشراء
- `"true"` - تجديد تلقائي بدون سؤال
- `"false"` - لا تجديد تلقائي أبدًا (مدفوعات يدوية فقط)
- `allow_auto_renew: true` - يمكن للعميل تمكين/تعطيل التجديد التلقائي لاحقًا

الخطوة 6: تحديد متطلبات المخزون

تحدد متطلبات المخزون الموارد المادية أو الافتراضية التي يجب تخصيصها عند توفير هذا المنتج.
هذه خطوة حاسمة تربط كتالوج المنتج الخاص بك بـ نظام إدارة المخزون.

```
{
  "inventory_items_list": "['SIM Card', 'Mobile Number']"
}
```

ما هي عناصر المخزون؟

عناصر المخزون هي موارد قابلة للتتبع مخزنة في نظام المخزون الخاص بـ OmniCRM. كل عنصر لديه:

- **النوع** - محدد بواسطة قالب المخزون (مثل "SIM Card"، "Mobile Number"، "Modem")
- **السمات الفريدة** - أرقام تسلسلية، عناوين MAC، أرقام هواتف، إلخ.
- **الحالة** - في المخزون، مخصص، غير مفعل، إلخ.
- **الموقع** - موقع ��ادي أو منطقي

كيف تعمل متطلبات المخزون:

قائمة `inventory_items_list` هي قائمة بايثون (كنص) تحتوي على أسماء أنواع المخزون. يجب أن يتطابق كل اسم تمامًا مع اسم قالب المخزون الموجود.

مثال على متطلبات المخزون:

```
# موبايل منتج بطاقة SIM
inventory_items_list: "['SIM Card', 'Mobile Number']"

# خدمة الإنترنت الثابت
inventory_items_list: "['Modem Router', 'Static IP Address']"

# خدمة رقمية (لا توجد عناصر مادية)
inventory_items_list: "[]"

# لاسلكي ثابت مع CPE
inventory_items_list: "['Fixed Wireless CPE', 'IPv4 Address',
'IPv6 Prefix']"
```

عملية اختيار المخزون

عندما يقوم المستخدم بتوفير منتج مع متطلبات المخزون، يفرض النظام عملية اختيار إلزامية:

1. تم النقر على زر التوفير

بعد اختيار المنتج، ينقر المستخدم على "توفير". بدلاً من التوفير الفوري، يتحقق النظام من `inventory_items_list`.

2. تظهر نافذة اختيار المخزون

إذا كان المخزون مطلوبًا، تظهر نافذة حوار مع قائمة منسدلة منفصلة لكل نوع من أنواع المخزون:

3. تصفية المخزون المتاح

تظهر القائمة المنسدلة لكل نوع من أنواع المخزون العناصر التي هي:

- **النوع الصحيح** - يتطابق مع اسم قالب المخزون تمامًا
- **حالة متاحة** - `item_state` هو "جديد" أو "في المخزون" (ليس "مخصص" أو "تالف")
- **غير مخصص** - `service_id` و `customer_id` هما NULL
- **في المخزون في الموقع** - يتم تصفيته اختياريًا حسب موقع المستودع/المتجر

مثال على خيارات القائمة المنسدلة:

بالنسبة لنوع المخزون "SIM Card"، قد تظهر القائمة المنسدلة:

كل خيار يعرض:

- معرف المخزون أو رقم المرجع
- المعرف الأساسي (`itemtext1` - مثل ICCID لبطاقة SIM، رقم للهاتف)
- الموقع الحالي (`item_location`)

4. الاختيار مطلوب للمتابعة

قاعدة حاسمة: لا يمكن أن يستمر التوفير بدون اختيار جميع عناصر المخزون المطلوبة.

- يتم تعطيل زر "متابعة" حتى يتم إجراء اختيارات في جميع القوائم المنسدلة
- يجب على المستخدم اختيار عنصر واحد لكل نوع من أنواع المخزون
- يتحقق النظام من الاختيارات قبل المتابعة

5. يتم تمرير المخزون المحدد إلى Ansible

بمجرد أن ينقر المستخدم على "متابعة"، يتم تمرير معرفات المخزون المحددة إلى برنامج Ansible كمتغيرات:

```
# اختار المستخدم:
# - معرف المخزون بطاقة SIM: 5001
# - 5002 :معرف المخزون رقم الهاتف

# المتغيرات المرسلة إلى Ansible:
{
    "product_id": 42,
    "customer_id": 123,
    "SIM Card": 5001,            # معرف المخزون
    "Mobile Number": 5002,        # معرف المخزون
    "access_token": "eyJ..."
}
```

ملاحظة: يتطابق اسم المتغير مع نوع المخزون تمامًا. يستخدم البرنامج `hostvars[inventory_hostname]['SIM Card']` للوصول إلى معرف المخزون.

6. يسترجع البرنامج تفاصيل المخزون الكاملة

يستخدم برنامج Ansible معرف المخزون لاسترجاع التفاصيل الكاملة:

```
- name: Get SIM Card details from inventory
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/inventory/inventory_id/{{ hostvars[inventory_hostname]['SIM Card'] }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: api_response_sim

- name: Extract ICCID and IMSI
  set_fact:
    iccid: "{{ api_response_sim.json.itemtext1 }}"
    imsi: "{{ api_response_sim.json.itemtext2 }}"
```

الآن يحتوي البرنامج على جميع تفاصيل بطاقة SIM (ICCID، IMSI، إلخ.) لتوفير المشترك في HSS.

7. تغير حالة المخزون إلى "مخصص"

بعد إنشاء سجل الخدمة، يقوم البرنامج بتحديث المخزون لربطه بالخدمة:

```
- name: Assign SIM Card to Service
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ hostvars[inventory_hostname]['SIM
Card'] }}"
    method: PATCH
    body:
      {
        "service_id": "{{
service_creation_response.json.service_id }}",
        "customer_id": "{{ customer_id }}",
        "item_state": "Assigned"
      }
```

مهم: يحدث تخصيص المخزون **أثناء تنفيذ البرنامج** كوظيفة محددة، وليس عند النقر على زر التوفير. هذا يعني:

- **خطر التخصيص المزدوج**: بين النقر على "توفير" وتخصيص المخزون، يمكن لمستخدم آخر نظريًا اختيار نفس عنصر المخزون
- **أفضل ممارسة**: للعمليات ذات الحجم الكبير، تنفيذ قفل المخزون أو استخدام معاملات قاعدة البيانات
- **التراجع عند الفشل**: إذا فشل البرنامج قبل تخصيص المخزون، يبقى المخزون غير مخصص ومتاحة لإعادة الاستخدام

لماذا لا يتم التخصيص في وقت سابق؟

لا يتم تخصيص المخزون عند النقر على "توفير" لأن:

1. **مطلوب معرف الخدمة**: لا يوجد `service_id` حتى يتم إنشاء الخدمة في البرنامج
2. **بساطة التراجع**: إذا فشل التوفير مبكرًا (مثل فشل إنشاء حساب OCS)، لا يحتاج المخزون إلى تنظيف
3. **المرونة**: يمكن للبرنامج أن يقرر عدم تخصيص المخزون بناءً على منطق شرطي

التعامل مع التوفير��ت الفاشلة:

عندما يفشل التوفير بعد تخصيص المخزون، يجب على كتلة الإنقاذ تحرير المخزون:

```
rescue:
  - name: Release inventory on failure
    uri:
      url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ hostvars[inventory_hostname]['SIM
Card'] }}"
      method: PATCH
      body:
        {
          "service_id": null,
          "customer_id": null,
          "item_state": "In Stock"
        }
    when: service_id is defined  # فقط إذا تم إنشاء الخدمة
```

هذا يضمن عدم ترك المخزون في حالة "مخصص" لخدمة غير موجودة أو فاشلة.

عندما تكون قائمة المخزون فارغة

يتم تخطي اختيار المخزون تمامًا، (قائمة فارغة) `inventory_items_list: "[]"` إذا كانت ويستمر التوفير على الفور. هذا شائع بالنسبة لـ:

- VPN **المنتجات الرقمية** - تراخيص البرمجيات، حسابات
- **ملحقات الخدمة** - مدفوعات إضافية للبيانات التي لا تحتاج إلى أجهزة جديدة
- **الخدمات الافتراضية** - التي لا تستهلك موارد قابلة للتتبع

لأنه يضيف فقط `inventory_items_list: "[]"` لديه "GB Data Boost **مثال:** ملحق "5G رصيدًا إلى خدمة موجودة دون الحاجة إلى أجهزة جديدة.

إعداد قالب المخزون

قبل استخدام نوع المخزون في `inventory_items_list`، يجب عليك إنشاء قالب المخزون:

1. الانتقال إلى **الإدارة** → **المخزون** → **القوالب**
2. إنشاء قالب بالاسم الدقيق (مثل "SIM Card")
3. تعريف الحقول:
   - `itemtext1_label`: "ICCID"
   - `itemtext2_label`: "IMSI"
   - `itemtext3_label`: "PUK Code"

4. إضافة عناصر المخزون من هذا النوع إلى المخزون

للحصول على تفاصيل كاملة حول إنشاء وإدارة قوالب المخزون، انظر إدارة المخزون.

عناصر متعددة من نفس النوع

بينما تكون `inventory_items_list` مصفوفة، فإن وجود أنواع مكررة (مثل `"['SIM Card', 'SIM Card']"`) **غير موصى به** لأنه قد يسبب ارتباكًا في واجهة المستخدم وتسمية متغيرات البرنامج.

بالنسبة للسيناريوهات التي تتطلب عناصر مشابهة متعددة:

الخيار 1: إنشاء أسماء قوالب مخزون متميزة

```
# خدمة هاتف مزدوج SIM
inventory_items_list: "['Primary SIM Card', 'Secondary SIM Card', 'Mobile Number']"
```

إنشاء قوالب منفصلة: "Primary SIM Card" و "Secondary SIM Card" بنفس الحقول ولكن بأسماء مختلفة.

الخيار 2: استخدام عنصر مخزون مجمع واحد

```
# مجموعة مزدوجة SIM
inventory_items_list: "['Dual SIM Kit', 'Mobile Number']"
```

حيث يحتوي قالب المخزون "Dual SIM Kit" على حقول لكل من بطاقتي SIM (`itemtext1`: ICCID الأساسي، `itemtext2`: ICCID الثانوي، إلخ.).

سيناريوهات المخزون الشائعة

خدمة الهاتف المحمول:

```
inventory_items_list: "['SIM Card', 'Mobile Number']"
```

- بطاقة SIM: مادية أو eSIM مع ICCID/IMSI
- رقم الهاتف: رقم الهاتف (MSISDN)

الإنترنت الثابت:

```
inventory_items_list: "['Modem Router', 'Static IP Address']"
```

- جهاز مودم: جهاز CPE مع عنوان MAC
- عنوان IP ثابت: IPv4 من مجموعة العناوين

لاسلكي ثابت:

```
inventory_items_list: "['Fixed Wireless CPE', 'IPv4 Address',
'IPv6 Prefix']"
```

- CPE: معدات في مقر العميل (هوائي، مودم)
- IPv4: عنوان IP عام
- IPv6 Prefix: بادئة /64 أو /56

ملاحظة: المواعيد والجدولة **ليست** عناصر مخزون. استخدم أنظمة جدولة/تقويم منفصلة لمواعيد التثبيت.

خدمة VoIP:

```
inventory_items_list: "['DID Number']"
```

- رقم DID: رقم الهاتف المباشر

ملاحظة: أسماء مستخدمي SIP بواسطة برنامج **تُولد برمجيًا** وكلمات المرور وتكوينات الحساب التوفير، وليس من اختيار المخزون.

الألياف/GPON:

```
inventory_items_list: "['ONT Device', 'GPON Port', 'IPv4 Address',
'Fiber Drop Cable']"
```

- جهاز ONT: محطة الشبكة الضوئية مع رقم تسلسلي
- منفذ GPON: منفذ محدد على OLT مع اتصال الألياف

- عنوان IPv4: IP عام أو خاص
- كابل الألياف: كابل الألياف المادي من الشارع إلى المباني (مراقب لإدارة الأصول)

تأجير المعدات:

```
inventory_items_list: "['Rental Modem']"
```

- تتبع أي مودم مع أي عميل
- مهم لاسترداد المعدات عند الإلغاء

لماذا تعتبر متطلبات المخزون مهمة

1. منع التخصيص المزدوج

بدون تتبع المخزون، يمكنك عن غير قصد:

- تخصيص نفس بطاقة SIM لعميلين
- تخصيص نفس عنوان IP لعدة خدمات
- شحن نفس الرقم التسلسلي للمعدات إلى مواقع مختلفة

يضمن اختيار المخزون أن كل عنصر مخصص لخدمة واحدة فقط.

2. مسار التدقيق

يخلق تخصيص المخزون مسار تدقيق كامل:

- أي بطاقة SIM مع أي عميل
- متى تم تخصيصها
- أي خدمة تستخدم أي رقم هاتف
- تاريخ المعدات (من كان لديه، ومتى، ولأي خدمة)

3. تخطيط الموارد

تتبع مستويات المخزون:

- تنبيه عند نفاد بطاقات SIM
- إعادة الطلب قبل نفاد المخزون
- تخطيط جداول الفنيين بناءً على توفر CPE

- إدارة تخصيص مساحة عنوان IP

4. تتبع التكاليف

ربط تكلفة الجملة بعنصر محدد:

- تتبع تكلفة كل بطاقة SIM
- حساب استهلاك المعدات
- تحديد العناصر المفقودة أ�ً المسروقة
- تكلفة السلع المباعة بدقة (COGS)

5. إلغاء التوفير

عند إلغاء الخدمة، يمكن أن يكون المخزون:

- تم إعادته إلى المخزون (بطاقات SIM، مودمات)
- متقاعد (معدات تالفة)
- تم إرجاعه إلى البائع (معدات للإيجار)
- محتفظ به لفترة السماح (أرقام الهواتف قبل الإصدار)

استكشاف مشكلات اختيار المخزون

المشكلة: تظهر رسالة "لا يوجد مخزون متاح"

الأسباب:

- لا توجد عناصر مخزون من النوع المطلوب في قاعدة البيانات
- جميع العناصر "مخصصة" بالفعل لخدمات أخرى
- تم وضع علامة على العناصر كـ "تالف" أو "خارج الخدمة"
- اسم قالب المخزون لا يتطابق تمامًا (حساس لحالة الأحرف)

الحل:

1. تحقق من وجود قالب المخزون: **الإدارة** → **المخزون** → **القوالب**
2. تحقق من تطابق اسم القالب تمامًا (بما في ذلك المسافات، الحالة)
3. أضف عناصر المخزون من هذا النوع: **الإدارة** → **المخزون** → **إضافة عنصر**
4. تحقق من أن العناصر في حالة "جديد" أو "في المخزون"
5. تحقق من أن العناصر ليست مخصصة بالفعل (`service_id` يجب أن يكون NULL)

المشكلة: لا تظهر نافذة اختيار المخزون

الأسباب:

- `inventory_items_list` فارغة: `"[]"`
- `inventory_items_list` أو غير محدد NULL
- فئة المنتج هي "ملحق" وترث مخزون الخدمة الأصلية

الحل:

- إذا كان المخزون مطلوبًا، قم بتعيين `inventory_items_list: "['Type1', 'Type2']"`
- تحقق من أن تعريف المنتج محفوظ بشكل صحيح
- تحقق من استجابة API ليتضمن المنتج `inventory_items_list`

"**المشكلة:** يفشل البرنامج مع "المخزون غير موجود

الأسباب:

- يشير البرنامج إلى اسم متغير خاطئ
- لم يتم تمرير معرف المخزون بشكل صحيح
- تم حذف المخزون بين الاختيار والتوفير

الحل:

- تحقق من أن البرنامج يستخدم المتغير الصحيح: `hostvars[inventory_hostname]['SIM Card']`
- تحقق من أن المتغير هو عدد صحيح: `{{ hostvars[inventory_hostname]['SIM Card'] | int }}`
- أضف معالجة الأخطاء في البرنامج للمخزون المفقود

انظر إدارة المخزون للحصول على تفاصيل كاملة حول إنشاء القوالب، وإضافة العناصر، وإدارة مستويات المخزون.

الخطوة 7: تحديد الميزات والشروط

الميزات والشروط هي محتوى تسويقي وقانوني يواجه العميل يساعد العملاء على فهم ما يشترونه والالتزامات المعنية.

```
{
  "features_list": "20GB High-Speed Data. Unlimited Calls & Texts. EU Roaming Included. No Contract. 30-Day Expiry",
  "terms": "Credit expires after 30 days. Data, calls, and texts valid only within expiry period. Fair use policy applies. See website for full terms."
}
```


الغرض والقيمة التجارية

قائمة الميزات - التسويق والمبيعات:

تخدم قائمة الميزات وظائف تجارية حيوية متعددة:

1. **تمييز المنتج** - يساعد العملاء على مقارنة المنتجات بسرعة واختيار المنتج المناسب
   - الميزات - "Prepaid Mobile 50GB" مقابل "Prepaid Mobile 20GB" توضح الف��ق بوضوح
   - بدون الميزات، يرى العملاء السعر فقط، مما يفوت قيمة العرض
2. **التواصل التسويقي** - نقاط البيع الرئيسية معروضة بشكل بارز
   - "EU Roaming Included" يجذب المسافرين الدوليين
   - "No Contract" يجذب العملاء الذين يتجنبون الالتزام
   - تدفع الميزات قرارات الشراء
3. **توقعات العملاء** - تحدد توقعات واضحة حول ما هو مشمول
   - تقلل من مكالمات الدعم ("هل يشمل ذلك المكالمات؟" → مدرجة بوضوح)
   - تمنع سوء الفهم وطلبات الاسترداد
   - تبني الثقة من خلال الشفافية
4. **الخدمة الذاتية** - تمكن العملاء من اختيار المنتجات المناسبة بأنفسهم
   - يقرأ العميل الميزات، ويفهم العرض، ويتخذ قرارًا مستنيرًا
   - يقلل من الحاجة إلى شرح موظفي المبيعات
   - يسرع من عملية الشراء
5. **تحسين محركات البحث والاكتشاف** - يمكن فهرسة الميزات للبحث
   - يبحث العميل عن "خطة هاتف محمول مكالمات غير محدودة" → يظهر المنتج
   - يحسن من إمكانية البحث في كتالوج المنتجات

الشروط والأحكام - القانونية والامتثال:

تخدم الشروط أغراض قانونية وتشغيلية:

1. **الحماية القانونية** - تحمي العمل من النزاعات والمسؤولية
    - "تنتهي الائتمان بعد 30 يومًا" - لا يمكن للعميل المطالبة باسترداد الأموال بعد 31 يومًا
    - "تنطبق سياسة الاستخدام العادل" - تمنع الإساءة (توصيل المكتب بالكامل على خطة الهاتف المحمول)
    - تخلق اتفاقية ملزمة
2. **إدارة التوقعات** - تمنع عدم رضا العملاء
    - "صالحة فقط خلال فترة انتهاء الصلاحية" - يعرف العميل موعد انتهاء الاستخدام
    - "لا يمكن استردادها" (للملحقات) - تمنع المشتريات الاحتيالية
    - تقلل من استردادات الأموال والشكاوى
3. **الامتثال التنظيمي** - تلبي المتطلبات القانونية
    - تتطلب قوانين حماية المستهلك شروطًا واضحة
    - تفرض لوائح الاتصا��ات الإفصاح
    - يمكن الإشارة إلى شروط GDPR/الخصوصية
4. **الحدود التشغيلية** - تحدد نطاق الخدمة والقيود
    - "غير مسؤول عن المناطق الميتة" - تخضع لتغطية الشبكة
    - "حتى" "قد تختلف السرعة" - تدير التوقعات بشأن السرعات
    - "يضمن استرداد المعدات المستأجرة" - يجب إرجاع المعدات
5. **مسار التدقيق** - يثبت أن العميل تم إبلاغه
    - قبل العميل الشروط عند الشراء
    - يسجل النظام توقيت القبول
    - قابل للدفاع عنه في النزاعات أو الإجراءات القانونية

مثال من العالم الحقيقي:

يشتري العميل خطة "مكالمات ورسائل نصية غير محدودة"، ثم يستخدمها للتسويق عبر الهاتف (10,000 مكالمة/يوم). بدون الشروط:

- العميل: "لقد قلت غير محدود!"
- المزود: "كنا نعني الاستخدام الشخصي..."
- العميل: "هذا ليس ما أعلنتم عنه!"
- النتيجة: نزاع، احتمال تقديم شكوى للجهات التنظيمية، ضرر للعلامة التجارية

مع الشروط: "تنطبق سياسة الاستخدام العادل. الخدمة للاستخدام الشخصي فقط. الاستخدام التجاري محظور."

- المزود: يشير إلى الشروط التي قبلها العميل
- لا يمكن للعميل الادعاء بالجهل
- أساس قانوني لتعليق الخدمة
- تم حل النزاع لصالح المزود

تنسيق قائمة الميزات:

فهم التنسيق الصحيح أمر حيوي لأن **التنسيق غير الصحيح يكسر عرض واجهة المستخدم**. قد تظهر الميزات كسلسلة طويلة واحدة بدلاً من نقاط نقطية، أو قد لا تظهر على الإطلاق.

يمكن تنسيق حقل `features_list` بطريقتين:

الخيار 1: سلسلة مفصولة بنقطة (موصى به)

تفصل الميزات بنقطة ومسافة (". "). تقوم واجهة المستخدم بتقسيمها على هذا الفاصل وتعرض كل ميزة كنقطة نقطية.

لماذا هذا التنسيق؟

- سهل التحرير - فقط اكتب الميزات مع نقاط بينهما
- لا توجد أحرف خاصة للهروب
- يعمل بشكل موثوق عبر جميع مكونات واجهة المستخدم
- سهل التحديث دون كسر بناء جملة JSON

صحيح مقابل غير صحيح:

الخيار 2: سلسلة مصفوفة JSON

```
"['20GB High-Speed Data', 'Unlimited Calls & Texts', 'EU Roaming Included']"
```

لاحظ أن هذه **سلسلة تحتوي على** JSON. يمكن أيضًا لواجهة المستخدم تحليل مصفوفات **JSON**، وليست مصفوفة JSON فعلية في قاعدة البيانات.

لماذا يوجد هذا التنسيق؟

- يسمح بالميزات التي تحتوي على نقاط فيها (مثل "حتى 100 ميغابت في الثانية. تخضع للتوافر.")
- أسهل التوليد البرمجي من السكربتات/API
- مستوردة من كتالوجات منتجات خارجية تستخدم المصفوفات

مهم: يجب أن يكون هذا بناء جملة قائمة بايثون صحيحة كسلسلة. علامات اقتباس مفردة حول كل عنصر، وعلامات اقتباس مزدوجة حول السلسلة الكاملة.

أي تنسيق يجب استخدامه؟

- **مفصولة بنقطة** - لإنشاء المنتجات يدويًا في واجهة المستخدم (أبسط، أقل عرضة للأخطاء)
- **مصفوفة JSON** - لإنشاء المنتجات بناءً على API/سكريبت (أكثر قوة للميزات المعقدة)

كلا التنسيقين ينتجان نفس المخرجات في واجهة المستخدم - إنهم فقط يؤثرون على كيفية إدخال البيانات.

أين تظهر الميزات في واجهة المستخدم:

1. كتالوج المنتجات (عرض العميل)

عندما يتصفح العملاء المنتجات المتاحة، تظهر الميزات كنقاط نقطية على كل بطاقة منتج:

2. صفحة تفاصيل المنتج

عند النقر على "عرض التفاصيل" تظهر معلومات المنتج الكاملة بما في ذلك:

- اسم المنتج والأيقونة
- التسعير (التكلفة الشهرية، تكلفة الإعداد)
- قائمة الميزات الكاملة (نقاط نقطية)
- الشروط والأحكام (انظر أدناه)
- التوافر والأهلية

3. تأكيد التوفير

أثناء التوفير، تظهر الميزات للمستخدم للمراجعة قبل التأكيد:

20 • :الميزاتlGB High-Speed Data • Unlimited Calls & Texts • EU Roaming Included • No Contract • 30-Day Expiry

التكلفة: £15.00/شهر تكلفة الإعداد: £0.00

[تأكيد وتوفير] [إلغاء]

4. تفاصيل الخدمة (بعد التوفير)

بعد أن تصبح الخدمة نشطة، تظهر الميزات على صفحة تفاصيل الخدمة للرجوع إليها من قبل العميل.

تنسيق الشروط والأحكام:

هو نص عادي يمكن أن يتضمن أسطر جديدة `terms` حقل:

أين تظهر الشروط في واجهة المستخدم:

1. صفحة تفاصيل المنتج

تظهر الشروط في قسم منكمش يتوسع عند النقر عليه:

2. تأكيد الطلب

أثناء التوفير، يتطلب مربع اختيار من المستخدم قبول الشروط:

زر [توفير] معطل حتى يتم التحقق

3. الفواتير

قد تتضمن شروط الخدمة على الفواتير كحواشي توضيحية.

أفضل الممارسات:

- **الميزات:** اجعلها موجزة (أقل من 50 حرفًا لكل منها)، وركز على الفوائد الرئيسية
- **الشروط:** تضمين المتطلبات القانونية الحرجة، وسياسات انتهاء الصلاحية، وسياسات الاستخدام العادل
- **كلاهما:** تحديث عند تغيير المنتج ل��بقاء العملاء على علم

الخطوة 8: ربط برنامج توفير Ansible

```
{
  "provisioning_play": "play_local_mobile_sim",
  "provisioning_json_vars": "{
    \"days\": 30,
    \"data_gb\": 20,
    \"voice_minutes\": \"unlimited\",
    \"sms_count\": \"unlimited\"
  }"
}
```

- `provisioning_play` - اسم برنامج Ansible (بدون امتداد .yaml)
- `provisioning_json_vars` - المتغيرات الافتراضية المرسلة إلى البرنامج
- يجب أن يوجد البرنامج في: `OmniCRM-API/Provisioners/plays/play_local_mobile_sim.yaml`

تعريف المنتج الكامل

```
{
  "product_name": "Prepaid Mobile 20GB",
  "product_slug": "prepaid-mobile-20gb",
  "category": "standalone",
  "service_type": "mobile",
  "enabled": true,
  "icon": "fa-solid fa-sim-card",
  "comment": "Prepaid mobile SIM with 20GB data, unlimited calls & texts",

  "retail_cost": 15.00,
  "wholesale_cost": 5.00,
  "retail_setup_cost": 0.00,
  "wholesale_setup_cost": 1.00,
  "contract_days": 30,

  "residential": true,
  "business": false,
  "customer_can_purchase": true,
  "available_from": "2025-01-01T00:00:00Z",
  "available_until": null,

  "auto_renew": "prompt",
  "allow_auto_renew": true,

  "inventory_items_list": "['SIM Card', 'Mobile Number']",

  "features_list": "[
    '20GB High-Speed Data',
    'Unlimited Calls & Texts',
    'EU Roaming Included',
    'No Contract',
    '30-Day Expiry'
  ]",
  "terms": "Credit expires after 30 days. Data, calls, and texts valid only within expiry period. Fair use policy applies.",

  "provisioning_play": "play_local_mobile_sim",
  "provisioning_json_vars": "{
    \"days\": 30,
    \"data_gb\": 20,
    \"voice_minutes\": \"unlimited\",
    \"sms_count\": \"unlimited\"
```

```
  }"
}
```

إنشاء منتج ملحق

تعمل الملحقات على تعزيز أو تعديل الخدمات الموجودة. تأتي في نوعين: **ملحقات افتراضية** (بدون موارد مادية) و **ملحقات الأجهزة** (تتطلب المخزون).

مثال 1: ملحق افتراضي (5GB Data Boost)

ملحق رقمي يضيف بيانات إلى خدمة موبايل موجودة:

```
{
  "product_name": "5GB Data Boost",
  "product_slug": "5gb-data-boost",
  "category": "addon",
  "service_type": "mobile",
  "enabled": true,
  "icon": "fa-solid fa-plus",
  "comment": "Add 5GB extra data to existing mobile service",

  "retail_cost": 5.00,
  "wholesale_cost": 1.50,
  "retail_setup_cost": 0.00,
  "wholesale_setup_cost": 0.00,
  "contract_days": 0,

  "residential": true,
  "business": true,
  "customer_can_purchase": true,

  "auto_renew": "false",
  "allow_auto_renew": false,

  "inventory_items_list": "[]",
  "relies_on_list": "",

  "features_list": "5GB High-Speed Data. Valid for 7 Days",
  "terms": "Data expires after 7 days or when exhausted. Cannot be
refunded.",

  "provisioning_play": "play_topup_charge_then_action",
  "provisioning_json_vars": "{
    \"data_gb\": 5,
    \"days\": 7
  }"
}
```

مثال 2: ملحق الأجهزة (تأجير مودم)

ملحق يوفر معدات مادية لخدمة الألياف الموجودة:

```
{
  "product_name": "WiFi 6 Modem Rental",
  "product_slug": "wifi6-modem-rental",
  "category": "addon",
  "service_type": "internet",
  "enabled": true,
  "icon": "fa-solid fa-router",
  "comment": "Add WiFi 6 modem to fiber service - rental",

  "retail_cost": 10.00,
  "wholesale_cost": 3.00,
  "retail_setup_cost": 0.00,
  "wholesale_setup_cost": 45.00,
  "contract_days": 30,

  "residential": true,
  "business": true,
  "customer_can_purchase": true,

  "auto_renew": "true",
  "allow_auto_renew": true,

  "inventory_items_list": "['Rental Modem']",
  "relies_on_list": "",

  "features_list": "WiFi 6 (802.11ax). Dual-band 2.4GHz + 5GHz. Up to 40 devices. Parental controls",
  "terms": "Equipment rental. Must be returned on service cancellation or £150 replacement fee applies. Equipment remains property of provider.",

  "provisioning_play": "play_addon_assign_modem",
  "provisioning_json_vars": "{
    \"device_type\": \"modem_router\",
    \"requires_configuration\": true
  }"
}
```

الاختلافات الرئيسية للملحقات:

- `category: "addon"` - يتم تطبيقها على الخدمة الموجودة، وليست مستقلة
- `contract_days: 0` (افتراضية) - تكرار الفوترة (إيجار متكرر) `30` أو

- `inventory_items_list: "[]"` أو (افتراضية) `"['Rental Modem']"` (أجهزة) - موارد مادية
- `auto_renew: "false"` أو (لمرة واحدة) `"true"` (إيجار) - سلوك متكرر
- `relies_on_list: ""` - يعني فارغ أنه ينطبق على أي خدمة من نوع `service_type` المطابق

لماذا تحتاج ملحقات الأجهزة إلى المخزون:

تتطلب ملحقات الأجهزة `inventory_items_list` لأنها:

1. **تتبع المعدات** - معرفة أي مودم مع أي عميل
2. **منع نفاد المخزون** - لا يمكن توفير الملحق إذا لم يكن هناك مودمات في المخزون
3. **الاسترداد** - عند إلغاء العميل، معرفة أي معدات يجب استردادها
4. **تتبع التكاليف** - ربط تكلفة الجملة برقم تسلسلي محدد
5. **الاستهلاك** - تتبع قيمة المعدات خلال فترة الإيجار
6. **الضمان** - تحديد الوحدات المعيبة بواسطة الرقم التسلسلي

تدفق توفير الملحقات مع المخزون:

عندما يضيف العميل "تأجير مودم WiFi 6" إلى خدمة الألياف الخاصة به:

1. **تم اختيار الملحق** - ينقر العميل على "إضافة إلى الخدمة"
2. **تظهر نافذة اختيار المخزون** - تمامًا مثل الخدمات المستقلة:
3. **تمت معالجة الدفع** - يتم فرض £10.00 إيجار شهري
4. **تم تخصيص المودم** - يتم تحديث المخزون:
   - `service_id`: مرتبط بخدمة الألياف
   - `customer_id`: مرتبط بالعميل
   - `item_state`: "مخصص"
5. **يتم تفعيل الشحن** - يتم إبلاغ نظام التنفيذ بشحن المودم
6. **التثبيت** - يستلم العميل المودم، ويقوم بتوصيله إلى ONT
7. **الفوترة المتكررة** - يتم فرض £10/شهر حتى يتم إلغاء الملحق

إلغاء توفير الملحقات:

عندما يلغي العميل تأجير المودم:

1. **تم بدء الإلغاء** - ينقر العميل على "إزالة الملحق"

2. **تم بدء عملية الإرجاع**:
    - يتم إرسال بريد إلكتروني مع تعليمات الإرجاع
    - يتم إنشاء ملصق شحن مدفوع مسبقًا
    - فترة سماح لمدة 14 يومًا قبل فرض غرامة
3. **تمت إعادة المعدات**:
    - يتم تحديث المخزون: `item_state` = "في المخزون" (بعد إعادة التأهيل)
    - أو `item_state` = "تالف" (إذا كانت معيبة)
    - مرتبط بالعميل التالي بمجرد إعادة التأهيل
4. **لا عودة**:
    - بعد 14 يومًا، يتم فرض غرامة استبدال قدرها £150
    - يتم وضع علامة على المخزون: `item_state` = "مفقود"
    - يتم استرداد تكلفة الجملة (£45) + قيمة الاستبدال

التسعير ��لملحقات:

يمكن تسعير الملحقات بشكل مختلف عن الخدمات المستقلة:

- الملحقات الافتراضية عادةً لا تحتوي على تكاليف إعداد
- قد تحتوي الملحقات المادية على تكاليف إعداد جملة للمعدات
- تستخدم الملحقات الإيجارية `contract_days` لتكرار الفوترة

المرحلة 2: عملية التوفير

عندما يطلب العميل منتج "Prepaid Mobile 20GB"، يقوم OmniCRM بتنظيم التوفير عبر Ansible.

مخطط تدفق التوفير

طلب العميل → اختيار المخزون → إنشاء وظيفة التوفير ↓ ↓ تفويض الدفع ← تجميع المتغيرات ← تخصيص المخزون → → إنشاء سجل الخدمة → إعداد حساب OCS ↓ ↓ تنفيذ برنامج Ansible الخدمة نشطة

تدفق التوفير خطوة بخطوة

1. يبدأ العميل الطلب

من صفحة العميل:

- ينقر الموظف على "إضافة خدمة"
- يختار "Prepaid Mobile 20GB" من دوار المنتج
- تظهر تفاصيل المنتج والتسعير

2. اختيار المخزون

يطلب النظام المخزون المطلوب:

- **بطاقة SIM** - تظهر القائمة المنسدلة بطاقات SIM المتاحة في المخزون
  - مثال: "SIM-00123 - ICCID: 8944..."
- **رقم الهاتف المحمول** - تظهر القائمة المنسدلة أرقام الهواتف المتاحة
  - مثال: "+44 7700 900123"

يختار الموظف أو العميل العناصر من المخزون المتاح.

3. تأكيد التسعير

يعرض النظام التسعير النهائي:

- تكلفة الإعداد: £0.00 (تفعيل مجاني)
- التكلفة الشهرية: £15.00
- المستحق اليوم: £15.00 (الشهر الأول)
- تاريخ التجديد: 30 يومًا من اليوم

إذا تم تمكين مطالبة التجديد التلقائي، يختار العميل:

- [ ] تجديد هذه الخدمة تلقائيًا كل 30 يومًا

4. تم النقر على زر التوفير

عند النقر على "توفير"، يقوم API:

- بإنشاء سجل `Provision` بحالة "قيد التشغيل" (status=1)
- بدمج المتغيرات من المنتج + الطلب + اختيارات المخزون
- يولد خيطًا في الخلفية لتنفيذ برنامج Ansible
- يعيد provision_id إلى واجهة المستخدم لتتبع الحالة

5. تجميع المتغيرات

يمزج النظام المتغيرات من مصادر متعددة:

من المنتج:

```
{
  "days": 30,
  "data_gb": 20,
  "voice_minutes": "unlimited",
  "sms_count": "unlimited"
}
```

من الطلب:

```
{
  "product_id": 42,
  "customer_id": 123,
  "SIM Card": 5001,
  "Mobile Number": 5002
}
```

المضاف من النظام:

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "initiating_user": 7
}
```

المتغيرات النهائية المرسلة إلى Ansible:

```
{
    "product_id": 42,
    "customer_id": 123,
    "SIM Card": 5001,
    "Mobile Number": 5002,
    "days": 30,
    "data_gb": 20,
    "voice_minutes": "unlimited",
    "sms_count": "unlimited",
    "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "initiating_user": 7
}
```

6. تنفيذ برنامج Ansible

ينفذ البرنامج `play_local_mobile_sim.yaml` مع هذه المتغيرات.

فهم برنامج توفير Ansible

افحص برنامج توفير حقيقي لفهم ما يحدث خلف الكواليس.

مثال على برنامج توفير بطاقة SIM موبايل

الموقع: `OmniCRM-API/Provisioners/plays/play_local_mobile_sim.yaml`

الهيكل العام:

```yaml
- name: Mobile SIM Provisioning
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Main block
      block:
        # 1. Load configuration
        # 2. Fetch product details from API
        # 3. Fetch customer details from API
        # 4. Fetch inventory details from API
        # 5. Create account in OCS (CGRateS)
        # 6. Add balances and allowances to OCS
        # 7. Create service record in CRM
        # 8. Assign inventory to service
        # 9. Record transactions
        # 10. Send welcome notifications

      rescue:
        # Rollback on failure
        # - Remove OCS account
        # - Release inventory
        # - Log error
```

شرح تفصيلي للبرنامج:

المهمة 1: تحميل التكوين

```yaml
- name: Include vars of crm_config
  ansible.builtin.include_vars:
    file: "../../crm_config.yaml"
    name: crm_config
```

يحمّل تكوين النظام بما في ذلك:

- عنوان URL وبيانات اعتماد OCS/CGRateS
- عنوان URL الأساسي لـ CRM
- تكوين المستأجر

المهمة 2: است��جاع تفاصيل المنتج

```
- name: Get Product information from CRM API
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/product/product_id/{{
product_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    return_content: yes
  register: api_response_product
```

ما الذي تفعله:

- يستدعي `GET /crm/product/product_id/42`
- يسترجع تعريف المنتج الكامل
- يخزنه في متغير `api_response_product`

من المنتج، فإننا نسترجع المنتج `provisioning_json_vars` **لماذا:** على الرغم من أن لدينا الكامل للحصول على:

- أحدث التسعير (قد يكون قد تغير منذ بدء الطلب)
- اسم المنتج لتسمية الخدمة
- قائمة الميزات للتوثيق
- تكاليف الجملة لتتبع الهامش

المهمة 3: تعيين حقائق الحزمة

```
- name: Set package facts
  set_fact:
    package_name: "{{ api_response_product.json.product_name }}"
    monthly_cost: "{{ api_response_product.json.retail_cost }}"
    setup_cost: "{{ api_response_product.json.retail_setup_cost
}}"
```

يستخرج القيم المستخدمة بشكل شائع إلى متغيرات بسيطة لسهولة القراءة.

المهمة 4: استرجاع تفاصيل المخزون

```
- name: Get SIM information from CRM API
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ hostvars[inventory_hostname]['SIM
Card'] }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: api_response_sim

- name: Set IMSI from Inventory response
  set_fact:
    imsi: "{{ api_response_sim.json.itemtext2 }}"
    iccid: "{{ api_response_sim.json.itemtext1 }}"
```

ما الذي تفعله:

- يبحث عن معرف مخزون بطاقة SIM 5001
- يسترجع تفاصيل SIM:
  - `itemtext1` = ICCID (رقم بطاقة SIM)
  - `itemtext2` = IMSI (هوية المشترك)
- يفعل نفس الشيء لرقم الهاتف المحمول (يسترجع رقم الهاتف)

لماذا هذا مهم:

- IMSI مطلوب لتوفير المشترك في HSS (خادم المشتركين المنزليين)
- يتم تسجيل ICCID في ملاحظات الخدمة لأغراض استكشاف الأخطاء وإصلاحها
- يتم عرض رقم الهاتف (MSISDN) للعميل ويستخدم للتوجيه

إنشاء UUID للخدمة المهمة 5:

```
- name: Generate UUID Fact
  set_fact:
    uuid: "{{ 99999999 | random | to_uuid }}"

- name: Set Service UUID
  set_fact:
    service_uuid: "Local_Mobile_SIM_{{ uuid[0:8] }}"
```

ما الذي تفعله:

- يولد UUID عشوائي
- ينشئ service_uuid مثل `Local_Mobile_SIM_a3f2c1d8`

لماذا:

- UUID الخدمة هو المعرف الفريد في OCS/CGRateS
- يستخدم لجميع عمليات الشحن
- يجب أن يكون فريدًا عالميًا عبر جميع الخدمات

المهمة 6: إنشاء حساب OCS

```
- name: Create account in OCS
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    headers:
      Content-Type: "application/json"
    body:
      {
        "method": "ApierV2.SetAccount",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "ActionPlanIds": [],
          "ExtraOptions": {
            "AllowNegative": false,
            "Disabled": false
          },
          "ReloadScheduler": true
        }]
      }
  register: ocs_create_response
```

ما الذي تفعله:

- يستدعي واجهة برمجة التطبيقات JSON-RPC لـ CGRateS
- ينشئ حسابًا جديدًا باستخدام service_uuid

- يضبط الحساب على نشط (ليس معطلاً)
- يمنع الرصيد السالب (وضع مسبق)

لماذا:

- هو المكان الذي تحدث فيه جميع عمليات الشحن OCS حساب
- يتم تخزين الأرصدة (البيانات، الصوت، الرسائل القصيرة، المال) هنا
- يتم تتبع الاستخدام وتقييمه في الوقت الفعلي

المهمة 7: إضافة رصيد البيانات

```
- name: Add 20GB Data Balance
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "DATA_20GB_Monthly",
            "Value": 21474836480,
            "ExpiryTime": "+720h",
            "Weight": 10,
            "DestinationIDs": "*any"
          }
        }]
      }
```

ما الذي تفعله:

- يضيف رصيد بيانات 20 جيجابايت إلى الحساب
- القيمة: 21474836480 بايت (20 * 1024 * 1024 * 1024)
- تنتهي صلاحيتها في 720 ساعة (30 يومًا)
- الوزن 10 (يي��م استهلاك الوزن الأعلى أولاً)

المهمة 8: إضافة صوت ورسائل نصية غير محدودة

```
- name: Add Unlimited Voice
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Account": "{{ service_uuid }}",
          "BalanceType": "*voice",
          "Balance": {
            "ID": "VOICE_Unlimited",
            "Value": 999999999,
            "ExpiryTime": "+720h"
          }
        }]
      }
```

- يضيف 999,999,999 ثانية من الصوت (غير محدود بشكل أساسي)
- تنتهي صلاحيتها في 30 يومًا

المهمة 9: إنشاء سجل الخدمة في CRM

```
- name: Add Service via API
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/service/"
    method: PUT
    body_format: json
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      {
        "customer_id": "{{ customer_id }}",
        "product_id": "{{ product_id }}",
        "service_name": "Mobile - {{ phone_number }}",
        "service_type": "mobile",
        "service_uuid": "{{ service_uuid }}",
        "service_status": "Active",
        "service_provisioned_date": "{{ provision_datetime }}",
        "retail_cost": "{{ monthly_cost }}",
        "wholesale_cost": "{{
api_response_product.json.wholesale_cost }}",
        "icon": "fa-solid fa-sim-card"
      }
  register: service_creation_response
```

ما الذي ينشئه هذا:

- سجل الخدمة مرتبط بالعميل
- روابط إلى OCS عبر `service_uuid`
- يخزن تكاليف التجزئة والجملة
- يضبط الحالة على "نشطة"
- يعيد `service_id` للعمليات اللاحقة

المهمة 10: تخصيص المخزون للخدمة

```
- name: Assign SIM Card to Service
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ hostvars[inventory_hostname]['SIM
Card'] }}"
    method: PATCH
    body_format: json
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      {
        "service_id": "{{
service_creation_response.json.service_id }}",
        "customer_id": "{{ customer_id }}",
        "item_state": "Assigned"
      }
```

ما الذي تفعله:

- يقوم بتحديث سجل مخزون بطاقة SIM
- يضبط `service_id` لربط SIM بالخدمة
- يغير الحالة من "في المخزون" إلى "مخصص"
- يتكرر ذلك لرقم الهاتف المحمول

لماذا:

- يتتبع أي SIM مخصص لأي عميل
- يمنع التخصيص المزدوج للمخزون
- يمكّن من تقارير المخزون والتدقيق

المهمة 11: تسجيل معاملة تكلفة الإعداد

```
- name: Add Setup Cost Transaction
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/transaction/"
    method: PUT
    body_format: json
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      {
        "customer_id": "{{ customer_id }}",
        "service_id": "{{
service_creation_response.json.service_id }}",
        "title": "{{ package_name }} - Setup",
        "description": "Activation fee",
        "retail_cost": "{{ setup_cost }}",
        "wholesale_cost": "{{
api_response_product.json.wholesale_setup_cost }}"
      }
```

ما الذي تفعله:

- يسجل تكلفة إعداد قدرها £0.00 للعميل (تجزئة)
- يسجل تكلفة جملة قدرها £1.00
- ينشئ سجل معاملة للفوترة

المهمة 12: كتلة الإنقاذ (معالجة الأخطاء)

```
rescue:
  - name: Remove account in OCS on failure
    uri:
      url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
      method: POST
      body:
        {
          "method": "ApierV2.RemoveAccount",
          "params": [{
            "Account": "{{ service_uuid }}"
          }]
        }

  - name: Fail the provisioning
    fail:
      msg: "Provisioning failed, rolled back OCS account"
```

ما الذي تفعله:

- إذا فشلت أي مهمة، يتم تنفيذ كتلة الإنقاذ
- تحذف حساب OCS الذي تم إنشاؤه جزئيًا
- تطلق المخزون مرة أخرى إلى "في المخزون"
- تفشل وظيفة التوفير مع رسالة خطأ

لماذا:

- يمنع وجود حسابات يتيمة في OCS
- يضمن التراجع النظيف عند حدوث أخطاء
- يحافظ على اتساق البيانات

التوفير مكتمل: ما تم إنشاؤه

بعد التوفير الناجح، يحتوي النظام على:

1. حساب OCS (CGRateS):

- معرف الحساب: `Local_Mobile_SIM_a3f2c1d8`
- الأرصدة:
  - 20 جيجابايت من البيانات (تنتهي صلاحيتها في 30 يومًا)

    - صوت غير محدود (999M يومًا في 30 يومًا) (ثانية، تنتهي صلاحيتها
    - رسائل نصية غير محدودة (999M يومًا في 30 يومًا) (رسالة، تنتهي صلاحيتها

2. سجل خدمة CRM:

- معرف الخدمة: 1234
- العميل: جون دو (customer_id: 123)
- المنتج: Prepaid Mobile 20GB (product_id: 42)
- اسم الخدمة: "Mobile - +44 7700 900123"
- UUID الخدمة: `Local_Mobile_SIM_a3f2c1d8`
- الحالة: نشطة
- التكلفة الشهرية: 15.00£ (تجزئة)، 5.00£ (جملة)
- الربح: 10.00£/شهر

3. تخصيصات المخزون:

- بطاقة SIM 5001: 123 العميل، 1234 مخصصة للخدمة
- رقم الهاتف المحمول 5002: مخصص للخدمة 1234، العميل 123

4. سجلات المعاملات:

- تم إنشاء سجل معاملة تكلفة الإعداد
- تم تسجيل رسوم الشهر الأول

5. يمكن للعميل الآن:

- عرض الخدمة في بوابة الخدمة الذاتية
- رؤية رصيد البيانات 20 جيجابايت
- إجراء المكالمات وإرسال الرسائل النصية
- إعادة الشحن أو إضافة الملحقات
- عرض الاستخدام في الوقت الحقيقي

المرحلة 3: إضافة الملحقات والمدفوعات الإضافية

بعد أن تصبح الخدمة نشطة، يمكن للعملاء شراء الملحقات لتعزيز خدمتهم.

تدفق توفير الملحقات

افترض أن العميل قد استخدم 18 جيجابايت من سعة 20 جيجابايت ويريد شراء ملحق "5GB Data Boost".

1. يتنقل العميل إلى الخدمة

- يفتح صفحة "Mobile - +44 7700 900123"
- يرى الاستخدام الحالي: 18 جيجابايت من 20 جيجابايت المستخدمة (90%)
- ينقر على "إضافة ملحق" أو "إعادة شحن"

2. يقوم النظام بتصفية الملحقات المتاحة

يعرض فقط الملحقات حيث:

- `category = "addon"`
- `service_type = "mobile"` (يتطابق مع نوع الخدمة)
- `residential = true` (إذا كان العميل سكنيًا)
- `enabled = true`

يرى العميل: "5GB Data Boost - £5.00"

3. يختار العميل الملحق

- ينقر على "5GB Data Boost"
- يؤكد الشراء مقابل £5.00
- يلتقط النظام تفويض الدفع

4. يتم بدء توفير الملحق

يستدعي النظام `play_topup_charge_then_action.yaml` مع المتغيرات:

```
{
    "product_id": 43,  # 5 منتجGB Data Boost
    "customer_id": 123,
    "service_id": 1234,  # الخدمة الموجودة
    "access_token": "eyJ...",
    "data_gb": 5,  # من provisioning_json_vars
    "days": 7      # من provisioning_json_vars
}
```

الاختلاف الرئيسي عن المستقل:

- يتم تضمين `service_id` (الخدمة الموجودة للتعديل)
- لا يتطلب المخزون
- لا يتم إنشاء الخد��ة (يعدل الموجودة)

استعراض برنامج توفير الملحقات

المهمة 1: استرجاع تفاصيل الخدمة

```
- name: Get Service information from CRM API
  uri:
    url: "http://localhost:5000/crm/service/service_id/{{
service_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: api_response_service

- name: Set service facts
  set_fact:
    service_uuid: "{{ api_response_service.json.service_uuid }}"
    customer_id: "{{ api_response_service.json.customer_id }}"
```

لماذا:

- تحتاج إلى `service_uuid` لإضافة الرصيد إلى حساب OCS الصحيح
- يتحقق من وجود الخدمة وأنها نشطة
- يضمن أن الخدمة تعود للعميل

المهمة 2: فرض رسوم على العميل

```
- name: Get Customer's Default Payment Method
  uri:
    url: "http://localhost:5000/api/payments/methods/default?
customer_id={{ customer_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: api_response_payment_method

- name: Get default payment method ID
  set_fact:
    payment_method_id: "{{
api_response_payment_method.json.data.payment_method_id }}"

- name: Charge customer
  uri:
    url: "http://localhost:5000/api/payments/charge"
    method: POST
    body_format: json
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      {
        "customer_id": "{{ customer_id | int }}",
        "amount": 5.00,
        "currency": "USD",
        "payment_method_id": "{{ payment_method_id }}",
        "metadata": {
          "description": "5GB Data Boost",
          "service_id": "{{ service_id | int }}",
          "product_id": "{{ product_id | int }}",
          "invoice": true
        }
      }
  register: charge_response

- name: Assert payment successful
  assert:
    that:
      - charge_response.json.success == true
```

ما الذي تفعله:

- يجد طريقة الدفع الافتراضية للعميل
- يفرض 5.00£ على البطاقة
- يسجل تكلفة الجملة 1.50£ لتتبع الهامش
- ينشئ معاملة مرتبطة بالخدمة
- يضيف إلى الفاتورة التالية
- يفشل التوفير إذا فشل الدفع

لماذا يتم فرض الرسوم أولاً:

- لا يتم تسليم الائتمان حتى يتم تأكيد الدفع
- يمنع الاحتيال
- يطابق الدفع مع توفير الملحق

المهمة 3: إضافة رصيد البيانات إلى OCS

```
- name: Add 5GB Data Balance
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "DATA_5GB_Boost_{{ uuid }}",
            "Value": 5368709120,
            "ExpiryTime": "+168h",
            "Weight": 20
          }
        }]
      }
```

ما الذي تفعله:

- يضيف 5 جيجابايت (5368709120 بايت) إلى الحساب
- تنتهي صلاحيتها في 168 ساعة (7 أيام)
- الوزن 20 (يتم استهلاك الوزن الأعلى أولاً - تعزيز قبل السماح الشهري)

رصيد العميل بعد الملحق:

- الشهري الأصلي: 2 جيجابايت متبقية (تنتهي صلاحيتها في 25 يومًا)
- التعزيز الجديد: 5 جيجابايت (تنتهي صلاحيتها في 7 أيام)
- الإجمالي المتاح: 7 جيجابايت
- ترتيب الاستخدام: يتم استهلاك التعزيز أولاً، ثم الشهري

المهمة 4: تسجيل المعاملة

```
- name: Add Addon Transaction
  uri:
    url: "http://localhost:5000/crm/transaction/"
    method: PUT
    body_format: json
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      {
        "customer_id": "{{ customer_id }}",
        "service_id": "{{ service_id }}",
        "title": "5GB Data Boost",
        "description": "Additional 5GB data valid for 7 days",
        "retail_cost": 5.00,
        "wholesale_cost": 1.50
      }
```

ما الذي تفعله:

- يسجل رسوم 5.00£ على العميل
- يسجل تكلفة جملة 1.50£
- يربط المعاملة بالخدمة للتقارير

ملخص تدفق الملحقات الكامل

1. يختار العميل الملحق من القائمة المصفاة

2. يتم تفويض الدفع وفرض الرسوم
3. يتم إضافة رصيد البيانات إلى حساب OCS
4. يتم تسجيل المعاملة في CRM
5. يرى العميل على الفور الرصيد المحدث: 7 جيجابايت متاحة

تتبع مالي:

- رسوم الخدمة الشهرية: 15£ تجزئة، 5£ جملة
- شراء الملحق: 5£ تجزئة، 1.50£ جملة

التجديد التلقائي: الملحقات المتكررة

يمكن تعيين بعض الملحقات للتجديد التلقائي (خطط البيانات الشهرية، الاشتراكات، إلخ).

تكوين المنتج:

```
{
  "product_name": "Monthly 10GB Data Plan",
  "category": "addon",
  "retail_cost": 10.00,
  "contract_days": 30,
  "auto_renew": "true",
  "provisioning_play": "play_topup_charge_then_action"
}
```

ينشئ التوفير خطة العمل:

```
- name: Create ActionPlan for Auto-Renewal
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body:
      {
        "method": "ApierV1.SetActionPlan",
        "params": [{
          "Id": "ServiceID_{{ service_uuid }}__ProductID_{{
product_id }}__MonthlyRenewal",
          "ActionPlan": [{
            "ActionsId": "Action_{{ product_slug }}",
            "Years": "*any",
            "Months": "*any",
            "MonthDays": "*any",
            "WeekDays": "*any",
            "Time": "00:00:00",
            "Weight": 10
          }],
          "Overwrite": false,
          "ReloadScheduler": true
        }]
      }
```

ما الذي تفعله:

- ينشئ مهمة مجدولة في OCS
- ينفذ `Action_{{ product_slug }}` كل 30 يومًا
- يفرض رسومًا على العميل ويعيد تطبيق رصيد البيانات
- يستمر حتى يلغي العميل

إدارة العملاء:

- يرى العميل "التجديد التالي: 1 فبراير 2025 - 10.00£" في عرض الخدمة
- يمكنه النقر على "إلغاء التجديد التلقائي" لإيقاف الرسوم المستقبلية
- يمكنه النقر على "تجديد الآن" لتطبيق رصيد الشهر التالي على الفور

المرحلة 4: إلغاء توفير الخدمات

عندما يلغي العميل الخدمة، يجب على النظام إزالة جميع الموارد بشكل نظيف.

محفزات إلغاء التوفير

يمكن أن يتم إلغاء التوفير بواسطة:

1. **إلغاء العميل** - ينقر العميل على "إلغاء الخدمة"
2. **إجراء إداري** - يقوم الموظف بتمييز الخدمة للإلغاء
3. **عدم الدفع** - تنتهي الخدمة بسبب عدم التجديد
4. **نهاية العقد** - انتهاء عقد محدد المدة

تدفق إل��اء التوفير

1. يبدأ العميل الإلغاء

- يتنقل إلى الخدمة
- ينقر على "إلغاء الخدمة"
- يطلب النظام: "هل أنت متأكد؟ سيتم فقدان أي رصيد متبقي."
- يؤكد العميل

2. فترة السماح (اختياري)

تقوم بعض الشركات بتطبيق فترة سماح:

- يتم تمييز الخدمة "قيد الإلغاء"
- تبقى نشطة لمدة 7-30 يومًا
- يمكن للعميل عكس الإلغاء خلال فترة السماح
- إلغاء تلقائي بعد فترة السماح

3. يتم إنشاء وظيفة إلغاء التوفير

ينشئ النظام وظيفة التوفير مع:

```
{
    "action": "deprovision",
    "service_id": 1234,
    "customer_id": 123,
    "service_uuid": "Local_Mobile_SIM_a3f2c1d8"
}
```

أو كتلة الإنقاذ من البرنامج `service.deprovisioning_play` يستدعي البرنامج المحدد في الأصلي.

4. برنامج إلغاء التوفير Ansible

```
- name: Deprovision Mobile Service
  hosts: localhost
  tasks:
    - name: Disable OCS Account
      uri:
        url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
        method: POST
        body:
          {
            "method": "ApierV2.SetAccount",
            "params": [{
              "Account": "{{ service_uuid }}",
              "ExtraOptions": { "Disabled": true }
            }]
          }

    - name: Remove ActionPlans (stop auto-renewals)
      uri:
        url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
        method: POST
        body:
          {
            "method": "ApierV1.RemoveActionPlan",
            "params": [{
              "Id": "ServiceID_{{ service_uuid }}__*"
            }]
          }

    - name: Update Service Status in CRM
      uri:
        url: "http://localhost:5000/crm/service/{{ service_id }}"
        method: PATCH
        body:
          {
            "service_status": "Deactivated",
            "service_deactivate_date": "{{ current_datetime }}"
          }

    - name: Release Inventory to Stock
      uri:
        url: "http://localhost:5000/crm/inventory/inventory_id/{{
sim_card_id }}"
        method: PATCH
```

```
        body:
          {
            "service_id": null,
            "customer_id": null,
            "item_state": "Decommissioned"
          }
```

ما الذي تفعله:

1. **تعطيل حساب OCS** - توقف جميع الشحنات، يتم حظر الاستخدام
2. **إزالة خطط العمل** - تلغي التجديدات التلقائية
3. **تحديث خدمة CRM** - الحالة "معطلة"، يتم تسجيل التاريخ
4. **إطلاق المخزون** - يتم وضع علامة على SIM كـ "غير مفعل"، متاحة لإعادة الاستخدام (بعد إعادة التأهيل)

5. بعد إلغاء التوفير

يقوم النظام بأداء التنظيف:

- لم يعد العميل يرى الخدمة في بوابة الخدمة الذاتية
- تبقى الخدمة في CRM للتقارير التاريخية
- يتم الاحتفاظ بالسجلات والمعاملات للفوترة
- يمكن إعادة تأهيل المخزون وإعادة استخدامه
- يمكن أرشفة حساب OCS بعد فترة الاحتفاظ

إلغاء التوفير الجزئي مقابل الكامل

إلغاء التوفير الجزئي (التعليق):

- يستخدم لعدم الدفع أو التعليق المؤقت
- يتم تعطيل حساب OCS ولكن لا يتم حذفه
- يتم الحفاظ على الأرصدة
- يمكن إعادة تمكينه عند استلام الدفع

```
- name: Suspend Service
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body:
      {
        "method": "ApierV2.SetAccount",
        "params": [{
          "Account": "{{ service_uuid }}",
          "ExtraOptions": { "Disabled": true }
        }]
      }
```

إلغاء التوفير الكامل (الإلغاء الدائم):

- يستخدم للإلغاء الدائم
- يتم حذف حساب OCS بالكامل
- يتم فقدان الأرصدة
- لا يمكن إعادة تمكينه

```
- name: Remove OCS Account
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body:
      {
        "method": "ApierV2.RemoveAccount",
        "params": [{
          "Account": "{{ service_uuid }}"
        }]
      }
```

أفضل الممارسات لإدارة المنتجات

إدارة دورة حياة المنتج

حالات المنتج:

- `enabled: true` - المنتج متاح للطلبات الجديدة
- `enabled: false` - تم تعطيل المنتج، تستمر الخدمات الحالية

تعطيل المنتجات:

- ضع علامة على المنتج كـ `enabled: false` لمنع الطلبات الجديدة
- تبقى الخدمات الحالية نشطة
- لا يزال بإمكان العملاء تجديد/تعديل الخدمات الحالية
- مفيد لإنهاء المنتجات القديمة

إدارة المخزون

حالات المخزون:

- `New` - مخزون جديد، جاهز للتخصيص
- `In Stock` - متاح للتوفير
- `Assigned` - مرتبط بخدمة العميل
- `Decommissioned` - يمكن إعادة تأهيله وإعادة استخدامه
- `Damaged` - يحتاج إلى إصلاح أو التخلص

إعادة استخدام المخزون:

بعد إلغاء التوفير:

- بطاقات SIM: وضع علامة "في المخزون" ووضع علامة "إعادة تأهيل"
- أرقام الهواتف: إطلاق بعد فترة النقل (30 يومًا)
- المعدات: اختبار، إعادة تأهيل، وضع علامة "مستخدمة"

مقاييس التوفير

مراقبة:

- معدل نجاح التوفير
- متوسط وقت التوفير
- نقاط الفشل الشائعة
- دوران المخزون

تكامل Mailjet مع OmniCRM

يتكامل OmniCRM مع **Mailjet** لإدارة جميع الاتصالات البريدية مع العملاء والموظفين، مما يضمن تسليم البريد الإلكتروني بشكل احترافي ومميز وموثوق لكل من رسائل البريد الإلكتروني المعاملات وحملات التسويق.

نظرة عامة

يوفر تكامل Mailjet:

- **رسائل البريد الإلكتروني المعاملات الآلية** - إعادة تعيين كلمة المرور، الفواتير، رسائل الترحيب، الإشعارات
- **موافقة التسويق (الاختيار المزدوج)** - إمكانية تدقيق للاشتراك/إلغاء الاشتراك في رسائل البريد الإلكتروني التسويقية، بما في ذلك تدفقات التأكيد وإعادة الإذن
- **قوالب البريد الإلكتروني** - أكثر من 10 أنواع بريد إلكتروني مُعدة مسبقًا مع قوالب Mailjet القابلة للتخصيص
- **حملات التسويق** - حملات بريد إلكتروني مجزأة بناءً على بيانات العملاء (يمكن بناء الشرائح من بيانات CRM أو تحميل CSV)
- **تسليم موثوق** - بنية تحتية احترافية للبريد الإلكتروني مع تتبع وتحليلات

لا يحتفظ OmniCRM بمزامنة قائمة جهات اتصال Mailjet مباشرة. استخدم **العملاء → جميع** (أو `GET /crm/contact/export`) لتصدير جهات الاتصال في واجهة الإدارة **جهات الاتصال** لإدارة القوائم الخارجية عند الحاجة.

التكوين

يتم تكوين Mailjet في `OmniCRM-API/crm_config.yaml` تحت قسم `mailjet`.

التكوين الأساسي

```
mailjet:
  api_key: your_mailjet_api_key
  api_secret: your_mailjet_api_secret
```

الحصول على بيانات اعتماد API:

1. إنشاء حساب في <https://www.mailjet.com>
2. الانتقال إلى **API** → **إعدادات الحساب** → **مفاتيح**
3. نسخ **المفتاح السري API** و **مفتاح**
4. لصقها في `crm_config.yaml`

تكوين قالب البريد الإلكتروني

يستخدم OmniCRM **10** يتم للاتصالات الآلية. **أنواع مختلفة من قوالب البريد الإلكتروني** تكوين ل قالب مع:

- **from_email** - عنوان البريد الإلكتروني للمرسل
- **from_name** - اسم العرض للمرسل
- **template_id** - معرف قالب Mailjet (رقمي)
- **subject** - سطر موضوع البريد الإلكتروني

أنواع القوالب والتكوين

رسالة ترحيب العميل

ترسل عند إنشاء حساب عميل جديد.

```
api_crmCommunicationCustomerWelcome:
  from_email: "support@yourcompany.com"
  from_name: "Your Company Support"
  template_id: 5977509
  subject: "Welcome to Your Company"
```

متى يتم الإرسال:

- يقوم عميل جديد بالتسجيل عبر بوابة الخدمة الذاتية
- يقوم الموظف بإنشاء حساب عميل جديد
- يقوم العميل بتفعيل الخدمة لأول مرة

المتغيرات المتاحة في القالب:

- `{{ var:customer_name }}` - الاسم الكامل للعميل
- `{{ var:email }}` - عنوان البريد الإلكتروني للعميل
- `{{ var:company_name }}` - اسم شركتك
- `{{ var:login_url }}` - رابط إلى بوابة الخدمة الذاتية
- `{{ var:support_url }}` - رابط إلى صفحة الدعم

رسالة فاتورة العميل

ترسل عند إنش��ء فاتورة وجاهزيتها للدفع.

```
api_crmCommunicationCustomerInvoice:
  from_email: "billing@yourcompany.com"
  from_name: "Your Company Billing"
  template_id: 6759851
  subject: "Your Invoice - "
```

متى يتم الإرسال:

- تم إنشاء فاتورة تلقائيًا لفترة الفوترة
- تم إنشاء فاتورة يدوية بواسطة الموظف
- يطلب العميل نسخة من الفاتورة

المتغيرات المتاحة في القالب:

- `{{ var:customer_name }}` - الاسم الكامل للعميل
- `{{ var:invoice_number }}` - معرف/رقم الفاتورة
- `{{ var:invoice_date }}` - تاريخ إصدار الفاتورة
- `{{ var:due_date }}` - تاريخ استحقاق الدفع
- `{{ var:total_amount }}` - المبلغ الإجمالي المستحق
- `{{ var:invoice_url }}` - رابط لعرض/تنزيل الفاتورة PDF
- `{{ var:pay_url }}` - رابط لدفع الفاتورة عبر الإنترنت

مرفق الفاتورة:

يتم إرفاق PDF الفاتورة تلقائيًا بالبريد الإلكتروني.

تذكير فاتورة العميل

ترسل لتذكير العملاء بالفواتير المتأخرة.

```
api_crmCommunicationCustomerInvoiceReminder:
  from_email: "billing@yourcompany.com"
  from_name: "Your Company Billing"
  template_id: 6759852
  subject: "Payment Reminder - Invoice Overdue"
```

متى يتم الإرسال:

- يوم (يمكن تكوينها) X الفاتورة متأخرة بمقدار
- تم تفعيل تذكير يدوي بواسطة الموظف
- سير العمل التلقائي للتذكير (إذا تم تكوينه)

المتغيرات المتاحة في القالب:

- `{{ var:customer_name }}`
- `{{ var:invoice_number }}`
- `{{ var:due_date }}`
- `{{ var:days_overdue }}`
- `{{ var:total_amount }}`
- `{{ var:pay_url }}`

رسالة ترحيب مستخدم الموظف

ترسل عند إنشاء حساب مستخدم موظف جديد.

```
api_crmCommunicationUserWelcome:
  from_email: "admin@yourcompany.com"
  from_name: "Your Company Admin"
  template_id: 5977510
  subject: "Welcome to the Team"
```

متى يتم الإرسال:

- يقوم المسؤول بإنشاء مستخدم موظف جديد
- يتم النقر على زر "إرسال رسالة الترحيب" في إدارة المستخدمين

المتغيرات المتاحة في القالب:

- `{{ var:user_name }}` - الاسم الكامل لمستخدم الموظف
- `{{ var:email }}` - بريد المستخدم
- `{{ var:role }}` - الدور (الأدوار) المعينة
- `{{ var:login_url }}` - رابط لتسجيل الدخول إلى بوابة الإدارة
- `{{ var:temp_password }}` - كلمة مرور مؤقتة (إذا كانت قابلة للتطبيق)
- `{{ var:support_email }}` - جهة الاتصال للدعم الفني

إعادة تعيين كلمة مرور المستخدم

ترسل عندما يطلب المستخدم إعادة تعيين كلمة مروره.

```
api_crmCommunicationUserPasswordReset:
  from_email: "noreply@yourcompany.com"
  from_name: "Your Company Security"
  template_id: 5977511
  subject: "Password Reset Request"
```

متى يتم الإرسال:

- ينقر المستخدم على "نسيت كلمة المرور" في صفحة تسجيل الدخول
- يقدم المستخدم طلب إعادة تعيين كلمة المرور

المتغيرات المتاحة في القالب:

- `{{ var:user_name }}`
- `{{ var:reset_url }}` - رابط إعادة تعيين كلمة المرور المحدود بالوقت (عادةً 1 ساعة)
- `{{ var:expiry_time }}` - متى تنتهي صلاحية رابط إعادة التعيين

ملاحظة أمان:

تنتهي صلاحية روابط إعادة التعيين بعد فترة زمنية محددة (الافتراضي 1 ساعة).

نجاح إعادة تعيين كلمة مرور المستخدم

ترسل لتأكيد أنه تم تغيير كلمة المرور بنجاح.

```
api_crmCommunicationUserPasswordResetSuccess:
  from_email: "noreply@yourcompany.com"
  from_name: "Your Company Security"
  template_id: 5977512
  subject: "Password Changed Successfully"
```

متى يتم الإرسال:

- يكمل المستخدم بنجاح عملية إعادة تعيين كلمة المرور
- مباشرة بعد تعيين كلمة المرور الجديدة

المتغيرات المتاحة في القالب:

- `{{ var:user_name }}`
- `{{ var:change_date }}` - تاريخ/وقت تغيير كلمة المرور
- `{{ var:ip_address }}` - عنوان IP (اختياري) للتغيير
- `{{ var:support_email }}` - جهة الاتصال إذا كان التغيير غير مصرح به

تغيير كلمة مرور المستخدم

ترسل عندما يغير المستخدم كلمة مروره من الإعدادات.

```
api_crmCommunicationUserPasswordChange:
  from_email: "noreply@yourcompany.com"
  from_name: "Your Company Security"
  template_id: 5977513
  subject: "Password Change Notification"
```

متى يتم الإرسال:

- يغير المستخدم كلمة المرور من الملف الشخصي/الإعدادات
- يقوم المسؤول بإعادة تعيين كلمة مرور المستخدم

المتغيرات المتاحة في القالب:

- `{{ var:user_name }}`
- `{{ var:change_date }}`
- `{{ var:changed_by }}` - "نفسه" أو اسم المسؤول
- `{{ var:support_email }}`

التحقق من البريد الإلكتروني

ترسل للتحقق من عنوان البريد الإلكتروني للمستخدم.

```
api_crmCommunicationEmailVerification:
  from_email: "noreply@yourcompany.com"
  from_name: "Your Company"
  template_id: 5977514
  subject: "Verify Your Email Address"
```

متى يتم الإرسال:

- تم إنشاء حساب جديد (عميل أو موظف)
- يغير المستخدم عنوان بريده الإلكتروني
- مطلوب التحقق من البريد الإلكتروني لأغراض الأمان

المتغيرات المتاحة في القالب:

- `{{ var:user_name }}`
- `{{ var:verification_url }}` - رابط للتحقق من البريد الإلكتروني
- `{{ var:verification_code }}` - رمز للدخول يدويًا (بديل للرابط)

إشعار انتهاء الرصيد

ترسل عندما تنتهي صلاحية رصيد ��دمة العميل/الحد المسموح به.

```
api_crmCommunicationsBalanceExpired:
  from_email: "support@yourcompany.com"
  from_name: "Your Company Support"
  template_id: 5977515
  subject: "Your Service Balance Has Expired"
```

متى يتم الإرسال:

- انتهاء صلاحية الرصيد المدفوع مسبقًا
- تجاوز تاريخ تجديد الحد الشهري
- الوصول إلى تاريخ انتهاء الخدمة

المتغيرات المتاحة في القالب:

- `{{ var:customer_name }}`
- `{{ var:service_name }}` - اسم الخدمة المنتهية
- `{{ var:expiry_date }}`
- `{{ var:balance_type }}` - "بيانات"، "صوت"، "نقدي"، إلخ.
- `{{ var:renewal_url }}` - رابط للتجديد/إعادة الشحن

تنبيه انخفاض الرصيد

ترسل عندما ينخفض رصيد العميل تحت الحد المحدد.

```
api_crmCommunicationsBalanceLow:
  from_email: "support@yourcompany.com"
  from_name: "Your Company Support"
  template_id: 5977516
  subject: "Low Balance Alert"
```

متى يتم الإرسال:

- ينخفض الرصيد تحت الحد (على سبيل المثال، 20% المتبقية)
- تم تكوينه في خطة الخدمة أو OCS
- يؤدي المرا��بة في الوقت الحقيقي إلى تنبيه

المتغيرات المتاحة في القالب:

- `{{ var:customer_name }}`
- `{{ var:service_name }}`
- `{{ var:current_balance }}`
- `{{ var:threshold }}`
- `{{ var:balance_type }}`

- `{{ var:topup_url }}` - رابط لإضافة رصيد

رسالة تسليم eSIM

ترسل عندما يتم توفير eSIM وجاهز للتفعيل.

```
api_crmCommunicationEsimDelivery:
  from_email: "support@yourcompany.com"
  from_name: "Your Company Support"
  template_id: 0
  subject: "Your eSIM is Ready"
```

متى يتم الإرسال:

- أثناء التسجيل الذاتي eSIM تم توفير
- لخدمة العميل eSIM تم تعيين

المتغيرات المتاحة في القالب:

- `{{ var:customer_name }}` - الاسم الكامل للعميل
- `{{ var:iccid }}` - ICCID لـ eSIM
- `{{ var:activation_code }}` - سلسلة رمز تفعيل LPA
- `{{ var:has_qr_code }}` - Boolean، صحيح إذا كانت صورة رمز QR مرفقة
- `{{ var:has_activation_code }}` - Boolean، صحيح إذا كان رمز التفعيل متاحًا

النسخة الاحتياطية المدمجة:

عندما يكون `template_id` هو `0` (أو مفتاح القالب مفقود من التكوين)، يست��دم OmniCRM بريدًا إلكترونيًا HTML مميزًا مدمجًا مع رمز QR لـ eSIM. يتم مُعتمد بشكل مباشر. التحكم في الألوان واسم العلامة التجارية بواسطة قسم `branding` في `crm_config.yaml`. راجع دليل التكوين للحصول على التفاصيل.

إنشاء قوالب بريد إلكتروني Mailjet

لكل نوع بريد إلكتروني، تحتاج إلى إنشاء قالب مطابق في Mailjet.

الخطوة 1: إنشاء قالب في Mailjet

1. تسجيل الدخول إلى لوحة معلومات Mailjet
2. الانتقال إلى **قوالب البريد الإلكتروني** ← **القوالب المعاملات**
3. النقر على **إنشاء قالب جديد**
4. اختيار **برمجة قالبك الخاص** (للمستخدمين المتقدمين) أو **استخدام منشئ القوالب**

الخطوة 2: تصميم القالب

استخدم منشئ السحب والإفلات أو محرر HTML الخاص بـ Mailjet لتصميم بريدك الإلكتروني.

العناصر الأساسية:

- **رأس** - شعار الشركة والعلامة التجارية
- **تحية** - مخصصة باستخدام `{{ var:customer_name }}` أو `{{ var:user_name }}`
- **المحتوى** - نص الرسالة الرئيسي
- **المتغيرات** - إدراج متغيرات القالب من القائمة أعلاه
- **دعوة للعمل** - أزرار/روابط لإجراءات المستخدم
- **تذييل** - رابط إلغاء الاشتراك، عنوان الشركة، جهة الاتصال للدعم

مثال على القالب (إعادة تعيين كلمة المرور):

```
<!DOCTYPE html>
<html>
<head>
    <style>
        body { font-family: Arial, sans-serif; }
        .button { background-color: #4CAF50; color: white;
padding: 14px 28px; }
    </style>
</head>
<body>
    ![Logo](https://yourcompany.com/logo.png)

    <h2>Password Reset Request</h2>

    <p>Hello {{ var:user_name }},</p>

    <p>We received a request to reset your password. Click the
button below to create a new password:</p>

    <a href="{{ var:reset_url }}" class="button">Reset
Password</a>

    <p>This link expires in {{ var:expiry_time }}.</p>

    <p>If you didn't request this, please ignore this email.</p>

    <hr>
    <p style="font-size: 12px; color: #666;">
        Your Company | support@yourcompany.com<br/>
        123 Business St, City, Country
    </p>
</body>
</html>
```

الخطوة 3: الحصول على معرف ��لقالب

1. حفظ القالب في Mailjet
2. ملاحظة **معرف القالب** (رقمي، على سبيل المثال، `5977509`)
3. نسخ هذا المعرف إلى `crm_config.yaml`

الخطوة 4: اختبار القالب

1. في Mailjet، استخدم ميزة **البريد الإلكتروني التجريبي**
2. قدم قيمًا نموذجية لجميع المتغيرات
3. أرسل بريدًا إلكترونيًا تجريبيًا لنفسك
4. تحقق من التنسيق والروابط والعلامة التجارية

الخطوة 5: التكوين في OmniCRM

أضف تكوين القالب إلى `crm_config.yaml`:

```
mailjet:
  api_key: your_api_key
  api_secret: your_secret

  api_crmCommunicationUserPasswordReset:
    from_email: "noreply@yourcompany.com"
    from_name: "Your Company Security"
    template_id: 5977511
    subject: "Password Reset Request"
```

أعد تشغيل واجهة برمجة تطبيقات OmniCRM لتصبح التغييرات سارية المفعول:

```
cd OmniCRM-API
sudo systemctl restart omnicrm-api
```

تصدير جهات الاتصال (CRM)

يمكن للمسؤولين تصفح وتصدير جهات الاتصال من CRM دون مزامنة قائمة Mailjet مباشرة، والامتثال، أو الاستيراد اليدوي إلى Mailjet (أو ESP آخر).

واجهة الإدارة

توفر جدولًا مقسمًا مع بحث، ومرشحات (`/contacts`) **العملاء → جميع جهات الاتصال**، وتصدير **CSV**. يستخدم التصدير (`marketing_consent_status`، `customer_status`) نفس المرشحات والبحث مثل الجدول.

واجهة برمجة التطبيقات

تتطلب كلا نقطتي النهاية إذن `VIEW_CUSTOMER_CONTACT`.

الطريقة	المسار	الغرض
`GET`	`/crm/contact/search`	قائمة جهات الاتصال المقسمة مع `search`, `page`, `per_page`, `sort`, `order`, `filters` (JSON)
`GET`	`/crm/contact/export`	تنزيل CSV بنفس معلمات الاستعلام (حد أقصى 50,000 صف)

أعمدة CSV: `contact_id`, `customer_id`, `customer_name`, `customer_status`, الاسم، البريد الإلكتروني، الهاتف، المسمى الوظيفي، `contact_types`, `marketing_consent_status`, `marketing_consent_granted_at`, `created`, `last_modified`.

عند إعداد قائمة للاستخدام التسويقي. تصفية على `marketing_consent_status=granted` باستخدام سير عمل استيراد جهات الاتصال الخاصة (الخاص بك ESP أو Mailjet) استورد CSV إلى بهم.

التنفيذ: `OmniCRM-API/services/contact_service.py` (`search_contacts`, `export_contacts_csv`).

موافقة التسويق (الاختيار المزدوج)

ترفض Mailjet الموافقة على استيراد قائمة جهات الاتصال ما لم يتمكن المرسل من إثبات أن كل مستلم قد وافق صراحةً على تلقي البريد الإلكتروني التسويقي. لتلبية هذا المطلب، يقوم OmniCRM بتخزين موافقة التسويق لكل جهة اتصال ويعرض تدفق اختيار مزدوج ("البريد الإلكتروني للتأكيد") الذي ينتج مسار موافقة يمكن تدقيقه.

توجد التنفيذات في `OmniCRM-API/services/marketing_consent_service.py`. يجب فقط استخدام جهات الاتصال التي لديها `marketing_consent_status == 'granted'` للإرسال التسويقي أو استيرادات القوائم الخارجية.

نموذج البيانات

تشمل الجدولين المعنيين:

- `contact.marketing_consent_status` (`Enum: none | pending | granted | revoked`) و `contact.marketing_consent_granted_at` - الحالة الحالية غير المهيكلة المستخدمة للتصفية السريعة في جدول جهات الاتصال والتصدير.
- `marketing_consent` - **سجل تدقيق إضافي فقط**. كل انتقال حالة (نقر الاشتراك، نقر التأكيد، تبديل الخدمة الذاتية، إلغاء عبر الويب، إعادة الإذن من المسؤول) يكتب صفًا جديدًا يلتقط معرف جهة الاتصال، البريد الإلكتروني في وقت الحدث، وكيل المستخدم، رمز موقّع، وتواريخ التأكيد/ IP، المصدر، إصدار نص الموافقة، عنوان الإلغاء. السجل هو مصدر الحقيقة إذا طلبت Mailjet يومًا ما إثبات الموافقة لمستلم معين.

يتم تكوين `marketing_consent.consent_text_version` في "إصدار نص الموافقة" تحت `crm_config.yaml` ويتم ختمه على كل صف تدقيق، لذا فإن تغيير صياغة نص الموافقة في المستقبل يكون غير تدميري: تظل الصفوف الموجودة تشير إلى الإصدار الذي رآه المستخدم بالفعل.

التكوين

أضف كتلة `marketing_consent` إلى `crm_config.yaml`:

```yaml
marketing_consent:
  consent_text_version: "v1-2026-05"
  confirmation_url_path: "/marketing/confirm"
  token_expiry_days: 30
  re_permission_batch_size: 200
  re_permission_rate_per_second: 5
  re_permission_skip_pending_within_hours: 72
```

يمكنك أيضًا تكوين قالب Mailjet للبريد الإلكتروني للتأكيد (نص/HTML وإلا سيتم استخدام نص ثابت):

```yaml
mailjet:
  api_crmCommunicationMarketingConfirm:
    from_email: "support@example.com"
    from_name: "Example Mobile Support"
    template_id: 0       # 0 = استخدم نص HTML المدمج
    subject: "Please confirm your subscription"
```

متغيرات قالب Mailjet المعروضة لقالب مخصص: `firstname`، `confirmation_link`، `consent_text_version`.

التدفق A: الاشتراكات الجديدة

يعرض نموذج الاشتراك الذاتي مربع اختيار "الاشتراك في أخبار Example Mobile للحصول على أحدث الأخبار، والفعاليات، والعروض الترويجية". عند تحديده:

1. `routes/auth.py /self-signup-partial` يستخرج `customer.subscribe_to_newsletter` من جسم الطلب.
2. بعد إنشاء جهة الاتصال، يتم استدعاء `marketing_consent_service.request_consent` مع `source='signup'`، وعنوان IP ووكيل المستخدم للطلب.
3. يتم كتابة صف تدقيق `marketing_consent` مع `granted=True`، `confirmed_at=NULL`، ورمز موقّع آمن. يتم تعيين `marketing_consent_status` لجهة الاتصال إلى `pending`.
4. يتم إرسال بريد إلكتروني معاملات Mailjet يحتوي على `${public_url}${confirmation_url_path}?token=...`.
5. عندما ينقر المستخدم على الرابط، يقوم `GET /crm/marketing-consent/confirm` باسترداد الرمز: يتم ختم `confirmed_at` على صف التدقيق ويتحول `marketing_consent_status` إلى `granted`.

التدفق B: الخدمة الذاتية (المستخدمون المسجلون)

تظهر الخدمة الذاتية صفحة "التفضيلات" (`/self-care/preferences`) التي تستدعي:

الطريقة	المسار	الغرض
`GET`	`/crm/marketing-consent/me`	الحالة الحالية + تاريخ التدقيق الكامل
`POST`	`/crm/marketing-consent/me/subscribe`	بدء تدفق الاختيار (إعادة) المزدوج
`POST`	`/crm/marketing-consent/me/unsubscribe`	إلغاء الموافقة على التسويق على الفور

يجب إزالة جهة الاتصال. `revoked` ويعين الحالة إلى `granted=False` يكتب الإلغاء صف تدقيق من أي قوائم تسويقية خارجية يتم الاحتفاظ بها خارج OmniCRM.

C: التدفق إعادة الإذن لقاعدة البيانات الحالية (المسؤول)

لا تحتوي جهات الاتصال القديمة التي تم إنشاؤها قبل إصدار هذه الميزة على سجل موافقة وبالتالي فهي في `marketing_consent_status='none'`. يمكن للمسؤولين أن يطلبوا منهم الاشتراك عبر حملة إعادة الإذن:

الطريقة	المسار	الغرض
`GET`	`/crm/oam/marketing-consent/stats`	عدادات لكل حالة + قائمة المرشحين + عينات
`POST`	`/crm/oam/marketing-consent/re-permission?dry_run=true`	التخطيط (لم يتم إر��ال بريد إلكتروني)
`POST`	`/crm/oam/marketing-consent/re-permission?dry_run=false`	التنفيذ. يتم التحكم فيه وفقًا للتكوين.

التي تجاوزت `pending` أو ، `revoked` ، `none` تقوم الوظيفة بمسح كل جهة اتصال حالتها ولديها بريد إلكتروني صالح، ولم يتم إلغاء ، `re_permission_skip_pending_within_hours` اشتراكها، وحالة العميل ليست `request_consent` مغلقة. لكل مرشح، يتم استدعاء نفس

المستخدم في تدفق الاشتراك، لذا فإن تنسيق صف التدقيق ودلالات الرمز متطابقة بغض النظر عن الأصل.

يتم تحديد معدل الإرسال عبر دلو الرموز بحجم `re_permission_rate_per_second`، مما يبقينا تحت حد الإنتاجية المعاملات لـ Mailjet. يحدد معلمة الاستعلام الاختيارية `limit` عدد الإرسال لكل تشغيل بحيث يمكن أن يكون التنفيذ الأول اختبار دخان صغير.

التوصيات للإطلاق الأول

1. تأكيد أن ترحيل قاعدة البيانات لـ `marketing_consent` قد تم.
2. زيارة **التسويق عبر البريد الإلكتروني → م��افقة التسويق** في واجهة الإدارة (`/email-marketing/marketing-consent`). ستكون مراجعة تفصيل الحالة. معظم جهات الاتصال القديمة في حالة `none`.
3. تشغيل **إعادة الإذن الجافة** وفحص العدادات.
4. إلى دفعة صغيرة (على سبيل `limit` **إرسال رسائل تأكيد** مع تعيين معلمة الاستعلام المثال، 50) للتحقق من إمكانية التسليم ودعم حجم البريد الوارد.
5. لتغطية بقية القاعدة. الوظيفة غير مدمرة: يتم تخطي `limit` تشغيل مرة أخرى بدون جهات الاتصال التي لديها طلب معلق حديث.
6. بعد أن تستقر الاشتراكات، استخدم **العملاء → جميع جهات الاتصال**، تصفية `marketing_consent_status = granted`، **وتصدير** **CSV** للاستيراد إلى Mailjet (أو ESP الخاص بك) إذا كنت تحتفظ بقائمة تسويقية هناك.

مقاييس Prometheus

المقياس	التسميات	لاحظات
`omnicrm_marketing_consent_events_total`	`event`	`confirmation_sent` `confirmation_fail` `revoked`, `token_re` `token_invalid`, `to`
`omnicrm_marketing_consent_status_total`	`status`	مقياس لكل من `none|pending|gran` يتم تحديثه بواسطة `get_global_stats`.

تتم توثيق مقاييس واجهة برمجة التطبيقات Mailjet المعاملات (`omnicrm_mailjet_api_*`, `omnicrm_mailjet_emails_sent_*`) في المراقبة والمقاييس.

استكشاف الأخطاء وإصلاحها

البريد الإلكتروني لا يُرسل

- **السبب:** بيانات اعتماد API غير صحيحة، تم تعليق حساب Mailjet، أو معرف القالب غير صحيح
- **الإصلاح:**
  - تحقق من `api_key` و `api_secret` في `crm_config.yaml`
  - تحقق من حالة حساب Mailjet والفوترة
  - تحقق من أن معرف القالب موجود في Mailjet
  - تحقق من سجلات API بحثًا عن الأخطاء

المتغيرات القالب لا يتم استبدالها

- **السبب:** عدم تطابق اسم المتغير أو بيانات مفقودة في OmniCRM
- **الإصلاح:**
  - تحقق من أن أسماء المتغيرات تتطابق تمامًا (حساسة لحالة الأحرف)
  - استخدم تنسيق `{{ var:variable_name }}`
  - تحقق من أن OmniCRM تمرر بيانات المتغير في استدعاء API
  - اختبر باستخدام بيانات نموذجية في Mailjet

الفاتورة لا يتم إرفاقه PDF

- **السبب:** فشل في إنشاء PDF أو حجم الملف كبير جدًا
- **الإصلاح:**
  - تحقق من سجلات إنشاء الفاتورة
  - تحقق من أن قالب الفاتورة يتم عرضه بشكل صحيح
  - تأكد من أن PDF أقل من 15 ميغابايت (حد Mailjet)
  - اختبر إنشاء PDF الفاتورة بشكل منفصل

Mailjet تحتاج إلى قائمة جهات اتصال تسويقية في

- **السبب:** لم يعد OmniCRM يدفع جهات الاتصال إلى Mailjet تلقائيًا

- **الإصلاح:**
  - جمع الموافقات عبر تدفق موافقة التسويق (`granted` status)
  - تصدير من **العملاء → جميع جهات الاتصال** مع فلتر الموافقة، أو `GET /crm/contact/export?filters={"marketing_consent_status":["granted"]}`
  - استيراد CSV إلى Mailjet باستخدام واجهة استير��د جهات الاتصال الخاصة بهم

الوثائق ذات الصلة

- `administration_configuration` - مرجع تكوين Mailjet الكامل
- `payments_invoices` - إنشاء الفواتير وتسليم البريد الإلكتروني
- `authentication_flows` - رسائل إعادة تعيين كلمة المرور والتحقق
- `customer_care` - رسائل الترحيب في بوابة الخدمة الذاتية
- المراقبة والمقاييس - مقاييس واجهة برمجة التطبيقات Mailjet، تتبع تسليم البريد الإلكتروني، ومراقبة الأداء

قراءة إضافية

- وثائق Mailjet: <https://dev.mailjet.com/>
- مرجع واجهة برمجة التطبيقات Mailjet: <https://dev.mailjet.com/email/reference/>

واجهة برمجة التطبيقات للإشعارات للأنظمة الخارجية

تتيح واجهة برمجة التطبيقات للإشعارات في OmniCRM للأنظمة الخارجية والتكاملات من الطرف الثالث إرسال إشعارات داخل التطبيق إلى المستخدمين. وهذا يمكّن من إرسال تنبيهات آلية، وتحديثات الحالة، ومعلومات مهمة مباشرة إلى المستخدمين داخل واجهة OmniCRM.

انظر أيضًا: استخدام واجهة برمجة التطبيقات لطرق المصادقة، نظام تذاكر الدعم للإشعارات المتعلقة بالتذاكر، المراقبة والقياسات لمراقبة النظام.

نظرة عامة

توفر واجهة برمجة التطبيقات للإشعارات طريقة مرنة لأي نظام لإرسال الإشعارات إلى مستخدمي OmniCRM. تظهر الإشعارات في قائمة الإشعارات المنسدلة للمستخدم في واجهة OmniCRM ويمكن أن ترتبط اختياريًا بصفحات أو موارد محددة.

الميزات الرئيسية

1. **تكامل مرن**: يمكن لأي نظام إرسال إشعارات عبر واجهة برمجة التطبيقات
2. **أنواع إشعارات متعددة**: دعم لمجموعة متنوعة من أنواع الأحداث (التوفير، الفوترة، التذاكر، مخصص)
3. **بيانات وصفية غنية**: تضمين بيانات منظمة مع الإشعارات
4. **تنقل عبر الروابط**: يمكن أن ترتبط الإشعارات بصفحات محددة
5. **مصادقة مفتاح واجهة برمجة التطبيقات**: تكامل آمن من خادم إلى خادم
6. **تسليم في الوقت الحقيقي**: تظهر الإشعارات على الفور في واجهة المستخدم
7. **تخزين دائم**: يقوم النظام بتخزين الإشعارات ويجعلها قابلة للاسترجاع

حالات الاستخدام الشائعة

أنظمة التوفير:

- إعلام المستخدمين عند اكتمال تفعيل الخدمة
- تنبيه عن فشل التوفير
- تحديث المستخدمين حول مهام التوفير الطويلة

أنظمة الفوترة والدفع:

- إعلام المستخدمين عند إنشاء فواتير جديدة
- تأكيد معالجة الدفع
- تحذير بشأن انتهاء الخدمات قريبًا

المراقبة والتنبيه:

- إرسال تنبيهات صحة النظام إلى المسؤولين
- إعلام عن تجاوز الحدود
- تنبيه عن الأحداث الحرجة

التكاملات المخصصة:

- أنظمة سير العمل الخارجية
- تكاملات خدمات الطرف الثالث
- أنظمة التقارير الآلية

المصادقة

تدعم واجهة برمجة التطبيقات للإشعارات طريقتين للمصادقة:

مصادقة مفتاح واجهة برمجة التطبيقات (موصى بها للأنظمة الخارجية)

تقدم مفاتيح واجهة برمجة التطبيقات مصادقة آمنة وطويلة الأمد لتكاملات الخادم إلى الخادم. هذه هي الطريقة الموصى بها للأنظمة الخارجية.

تكوين مفتاح واجهة برمجة التطبيقات:

أضف مفتاح واجهة برمجة التطبيقات الخاص بك إلى `crm_config.yaml`:

```yaml
api_keys:
  your-secure-api-key-minimum-32-characters:
    roles:
      - admin
    description: "نظام إشعارات التوفير"
```

استخدام مفتاح واجهة برمجة التطبيقات:

قم بتضمين مفتاح واجهة برمجة التطبيقات في رأس `X-API-KEY`:

```
curl -X POST "https://yourcrm.com/crm/notification/send" \
  -H "X-API-KEY: your-secure-api-key-minimum-32-characters" \
  -H "Content-Type: application/json" \
  -d '{
    "user_id": 123,
    "type": "provision_complete",
    "title": "تم تفعيل الخدمة",
    "message": "تم تفعيل خدمتك الهاتفية بنجاح",
    "link": "/services/167"
  }'
```

مصادقة رمز الحامل

لأنظمة التفاعل أو الإشعارات التي يبدأها المستخدم، استخدم مصادقة رمز الحامل.

انظر استخدام واجهة برمجة التطبيقات للحصول على تفاصيل حول الحصول على رموز الحامل.

نقطة نهاية واجهة برمجة التطبيقات

إرسال إشعار

نقطة النهاية: `POST /crm/notification/send`

المصادقة: مفتاح واجهة برمجة التطبيقات (X-API-KEY) أو رمز الحام��

الوصف: إرسال إشعار إلى مستخدم معين

جسم الطلب


```
{
  "user_id": 123,
  "type": "provision_complete",
  "title": "تم تفعيل الخدمة",
  "message": "تم تفعيل خدمتك الهاتفية بنجاح",
  "link": "/services/167",
  "metadata": {
    "service_id": 167,
    "provision_job_id": "prov-2026-00042",
    "duration_seconds": 45
  },
  "related_ticket_id": 42,
  "related_user_id": 5
}
```


معلمات الطلب

الحقل	النوع	مطلوب	الوصف
`user_id`	عدد صحيح	**نعم**	معرف المستخدم الذي سيتم إخطاره
`type`	سلسلة	**نعم**	نوع الإشعار (انظر الأنواع أدناه)
`title`	سلسلة	**نعم**	عنوان الإشعار (حد أقصى 255 حرف)
`message`	سلسلة	لا	رسالة الإشعار التفصيلية
`link`	سلسلة	لا	عنوان URL/المسار للتنقل عند النقر
`metadata`	كائن	لا	بيانات منظمة إضافية (JSON)
`related_ticket_id`	عدد صحيح	لا	معرف التذكرة المرتبطة
`related_user_id`	عدد صحيح	لا	معرف المستخد�ً المرتبط (على سبيل المثال، من تسبب في الحدث)

أنواع الإشعارات

حالة الاستخدام الشائعة	الوصف	النوع
انتهى برنامج Ansible	اكتمل التوفير بنجاح	`provision_complete`
خطأ في تفعيل الخدمة	فشل التوفير	`provision_failed`
تنبيه التعاون	تم ذكر المستخدم @ في التذكرة	`ticket_mention`
سير عمل الدعم	تم تعيين التذكرة للمستخدم	`ticket_assigned`
إشعار مشترك التذكرة	تم تحديث التذكرة	`ticket_updated`
تنبيه الدعم	رد العميل على التذكرة	`customer_reply`
حدث نظام حرج	تنبيه على مستوى النظام	`system_alert`
تكامل نظام الفوترة	تم إنشاء فاتورة جديدة	`billing_invoice`
ويب هوك بوابة الدفع	تمت معالجة الدفع	`billing_payment`
مراقبة آلية	الخدمة تنتهي قريبًا	`service_expiring`
أي غرض آخر	نوع إشعار مخصص	`custom`

استجابة النجاح

الحالة: `201 تم الإنشاء`

```
{
  "result": "نجاح",
  "message": "تم إرسال الإشعار بنجاح",
  "notification": {
    "notification_id": 1234,
    "user_id": 123,
    "type": "provision_complete",
    "title": "تم تفعيل الخدمة",
    "message": "تم تفعيل خدمتك الهاتفية بنجاح",
    "link": "/services/167",
    "metadata": {
      "service_id": 167,
      "provision_job_id": "prov-2026-00042",
      "duration_seconds": 45
    },
    "is_read": false,
    "created_at": "2026-01-11T14:30:00Z"
  }
}
```


استجابات الخطأ

الحقل المطلوب مفقود (400 طلب غير صالح):


```
{
  "result": "فشل",
  "reason": "user_id مطلوب"
}
```


المستخدم غير موجود (404 غير موجود):


```
{
  "result": "فشل",
  "reason": "المستخدم 999 غير موجود"
}
```


فشل المصادقة (401 غير مصرح):

```
{
  "message": "مفتاح واجهة برمجة التطبيقات غير صالح"
}
```

أمثلة على التعليمات البرمجية

مثال بايثون

```python
import requests

# التكوين
CRM_URL = 'https://yourcrm.com'
API_KEY = 'your-secure-api-key-minimum-32-characters'

def send_notification(user_id, notification_type, title,
message=None, link=None, metadata=None):
    """إرسال إشعار إلى مستخدم عبر واجهة برمجة التطبيقات OmniCRM"""

    headers = {
        "Content-Type": "application/json",
        "X-API-KEY": API_KEY
    }

    payload = {
        "user_id": user_id,
        "type": notification_type,
        "title": title
    }

    # إضافة الحقول الاختيارية
    if message:
        payload["message"] = message
    if link:
        payload["link"] = link
    if metadata:
        payload["metadata"] = metadata

    response = requests.post(
        f"{CRM_URL}/crm/notification/send",
        headers=headers,
        json=payload
    )

    if response.status_code == 201:
        print(f"✓ تم إرسال الإشعار بنجاح: {response.json()}")
        return response.json()
```

```
    else:
        print(f"✗ فشل في إرسال الإشعار: {response.text}")
        response.raise_for_status()

# مثال على الاستخدام
if __name__ == "__main__":
    # إعلام المستخدم بإكمال التوفير
    send_notification(
        user_id=123,
        notification_type="provision_complete",
        title="تم تفعيل الخدمة الهاتفية",
        message="تم تفع��ل الخدمة ID 167 بنجاح",
        link="/services/167",
        metadata={
            "service_id": 167,
            "provision_job_id": "prov-2026-00042",
            "duration_seconds": 45,
            "playbook": "mobile_service_activation"
        }
    )
```

مثال Bash/cURL

```
#!/bin/bash

# التكوين
CRM_URL="https://yourcrm.com"
API_KEY="your-secure-api-key-minimum-32-characters"

# إرسال الإشعار
curl -X POST "${CRM_URL}/crm/notification/send" \
  -H "X-API-KEY: ${API_KEY}" \
  -H "Content-Type: application/json" \
  -d '{
    "user_id": 123,
    "type": "provision_complete",
    "title": "تم تفعيل الخدمة",
    "message": "تم تفعيل خدمتك الهاتفية بنجاح",
    "link": "/services/167",
    "metadata": {
      "service_id": 167,
      "provision_job_id": "prov-2026-00042",
      "duration_seconds": 45
    }
  }'
```

مثال Node.js

```
const axios = require('axios');

const CRM_URL = 'https://yourcrm.com';
const API_KEY = 'your-secure-api-key-minimum-32-characters';

async function sendNotification(userId, type, title, options = {})
{
    try {
        const response = await axios.post(
            `${CRM_URL}/crm/notification/send`,
            {
                user_id: userId,
                type: type,
                title: title,
                message: options.message,
                link: options.link,
                metadata: options.metadata,
                related_ticket_id: options.relatedTicketId,
                related_user_id: options.relatedUserId
            },
            {
                headers: {
                    'X-API-KEY': API_KEY,
                    'Content-Type': 'application/json'
                }
            }
        );

        console.log('✓ تم إرسال الإشعار:', response.data);
        return response.data;
    } catch (error) {
        console.error('✗ فشل في إرسال الإشعار:',
error.response?.data || error.message);
        throw error;
    }
}

// مثال على الاستخدام
sendNotification(
    123,
    'provision_complete',
```

```
    'تم تفعيل الخدمة',
    {
        message: 'تم تفعيل خدمتك الهاتفية بنجاح',
        link: '/services/167',
        metadata: {
            service_id: 167,
            provision_job_id: 'prov-2026-00042',
            duration_seconds: 45
        }
    }
);
```

سيناريوهات التكامل

لتوفير الخدمات Ansible برنامج

إعلام المستخدمين عند ��نتهاء برامج Ansible:

```python
# أو البرنامج النصي بعد التشغيل Ansible في نهاية مكون الإرجاع الخاص بـ
import requests

def notify_provisioning_complete(service_id, user_id,
success=True):
    """إعلام المستخدم بنتيجة التوفير"""

    notification_type = "provision_complete" if success else
"provision_failed"
    title = "تم تفعيل الخدمة" if success else "فشل تفعيل الخدمة"
    message = f"تم تفعيل الخدمة {service_id} بنجاح" if success
else f"حدث خطأ في تفعيل الخدمة {service_id}"

    requests.post(
        "https://yourcrm.com/crm/notification/send",
        headers={"X-API-KEY": "your-api-key"},
        json={
            "user_id": user_id,
            "type": notification_type,
            "title": title,
            "message": message,
            "link": f"/services/{service_id}",
            "metadata": {
                "service_id": service_id,
                "playbook": "mobile_service_activation",
                "success": success
            }
        }
    )
```

تكامل نظام الفوترة

إرسال إشعارات الفواتير من نظام الفوترة:

```
def notify_invoice_generated(customer_id, invoice_id, amount,
due_date):
    """إعلام جهة الاتصال الخاصة بالعميل بالفاتورة الجديدة"""

    # العثور على المستخدم المرتبط بالعميل
    user_id = get_customer_primary_contact_user_id(customer_id)

    if user_id:
        requests.post(
            "https://yourcrm.com/crm/notification/send",
            headers={"X-API-KEY": "your-billing-api-key"},
            json={
                "user_id": user_id,
                "type": "billing_invoice",
                "title": f"فاتورة جديدة #{invoice_id}",
                "message": f"الفاتورة بمبلغ ${amount} متاحة الآن.
تاريخ الاستحقاق: {due_date}",
                "link": f"/invoices/{invoice_id}",
                "metadata": {
                    "invoice_id": invoice_id,
                    "amount": amount,
                    "due_date": due_date,
                    "customer_id": customer_id
                }
            }
        )
```

تكامل نظام المراقبة

إرسال تنبيهات من أنظمة المراقبة:

```python
def send_monitoring_alert(admin_user_ids, alert_title,
alert_message, severity="warning"):
    """إرسال تنبيه النظام إلى المسؤولين"""

    for user_id in admin_user_ids:
        requests.post(
            "https://yourcrm.com/crm/notification/send",
            headers={"X-API-KEY": "your-monitoring-api-key"},
            json={
                "user_id": user_id,
                "type": "system_alert",
                "title": alert_title,
                "message": alert_message,
                "metadata": {
                    "severity": severity,
                    "alert_time": datetime.now().isoformat(),
                    "source": "prometheus_alertmanager"
                }
            }
        )

# مثال: تنبيه مساحة القرص
send_monitoring_alert(
    admin_user_ids=[1, 2, 3],
    alert_title="⚠ تحذير من مساحة القرص",
    alert_message="85 استخدام قرص خادم قاعدة البيانات تجاوز%",
    severity="warning"
)
```

ويب هوك بوابة الدفع

إعلام المستخدمين عند معالجة المدفوعات:

```python
from flask import Flask, request

app = Flask(__name__)

@app.route('/webhook/payment', methods=['POST'])
def payment_webhook():
    """معالجة ويب هوك بوابة الدفع"""

    payment_data = request.json

    if payment_data['status'] == 'succeeded':
        # إعلام المستخدم بالدفع الناجح
        requests.post(
            "https://yourcrm.com/crm/notification/send",
            headers={"X-API-KEY": "your-payment-webhook-api-key"},
            json={
                "user_id": payment_data['user_id'],
                "type": "billing_payment",
                "title": "تم استلام الدفع",
                "message": f"تمت معالجة الدفع بمبلغ
${payment_data['amount']} بنجاح",
                "link": f"/payments/{payment_data['payment_id']}",
                "metadata": {
                    "payment_id": payment_data['payment_id'],
                    "amount": payment_data['amount'],
                    "payment_method":
payment_data['payment_method']
                }
            }
        )

    return {"status": "ok"}
```

مراقبة انتهاء الخدمة

برنامج نصي آلي لإعلام المستخدمين بالخدمات التي ستنتهي:

```
import datetime
from sqlalchemy import create_engine, text

def check_expiring_services():
    """التحقق من الخدمات التي ستنتهي في الأيام السبعة القادمة وإعلام المستخدمين"""

    engine = create_engine("postgresql://...")

    # استعلام عن الخدمات التي ستنتهي قريبًا
    query = text("""
        SELECT s.service_id, s.end_date, u.id as user_id,
s.service_name
        FROM services s
        JOIN customer_contacts cc ON s.customer_id =
cc.customer_id
        JOIN users u ON cc.contact_id = u.contact_id
        WHERE s.end_date BETWEEN CURRENT_DATE AND CURRENT_DATE +
INTERVAL '7 days'
        AND s.deleted = false
    """)

    with engine.connect() as conn:
        results = conn.execute(query)

        for row in results:
            days_until_expiry = (row.end_date -
datetime.date.today()).days

            requests.post(
                "https://yourcrm.com/crm/notification/send",
                headers={"X-API-KEY": "your-automation-api-key"},
                json={
                    "user_id": row.user_id,
                    "type": "service_expiring",
                    "title": f"الخدمة ستنتهي في
{days_until_expiry} يوم",
                    "message": f"ستنتهي خدمتك {row.service_name}
في {row.end_date}",
                    "link": f"/services/{row.service_id}",
                    "metadata": {
                        "service_id": row.service_id,
                        "days_remaining": days_until_expiry,
```

```
                        "expiry_date": str(row.end_date)
                    }
                }
            )

# تشغيل يوميًا عبر cron
if __name__ == "__main__":
    check_expiring_services()
```

أفضل الممارسات

أمان مفتاح واجهة برمجة التطبيقات

1. **إنشاء مفاتيح قوية**: استخدم مولدات عشوائية آمنة من الناحية التشفيرية:

   ```
   openssl rand -base64 48
   ```

2. **مفاتيح منفصلة لكل نظام**: استخدم مفاتيح واجهة برمجة التطبيقات مختلفة للأنظمة الخارجية المختلفة

3. **توثيق غرض المفتاح**: دائمًا ما يتضمن وصفًا في `crm_config.yaml`

4. **تدوير المفاتيح بانتظام**: تغيير مفاتيح واجهة برمجة التطبيقات بشكل دوري (على سبيل المثال، ربع سنوي)

5. **عدم الالتزام بالمفاتيح**: ابقِ مفاتيح واجهة برمجة التطبيقات خارج التحكم في الإصدار

6. **أذونات الحد الأدنى**: منح الأدوار الضرورية فقط لكل مفتاح واجهة برمجة التطبيقات

تصميم الإشعارات

1. **عناوين واضحة**: اجعل العناوين وصفية وقابلة للتنفيذ (مثل "تم تفعيل الخدمة" وليس "تحديث")

2. **رسائل مختصرة**: احتفظ بالرسائل قصيرة ولكن معلوماتية

3. **تضمين الروابط**: قدم دائمًا رابطًا للمستخدمين للحصول على مزيد من التفاصيل
4. **استخدام البيانات الوصفية**: تخزين البيانات المنظمة للرجوع إليها في المستقبل وتصحيح الأخطاء
5. **اختيار النوع الصحيح**: استخدام أنواع الإشعارات المناسبة للتصنيف
6. **اختبار الإشعارات**: اختبار مع مستخدمين حقيقيين قبل نشرها في الإنتاج

معالجة الأخطاء

1. **معالجة الفشل بشكل جيد**: لا تدع فشل الإشعار يكسر سير العمل الرئيسي
2. **تسجيل جميع المحاولات**: تسجيل محاولات الإشعار للتصحيح
3. **إعادة المحاولة عند الأخطاء العابرة**: تنفيذ منطق إعادة المحاولة لمشاكل الشبكة
4. **التحقق من معرفات المستخدمين**: التحقق من وجود المستخدم قبل ��لإرسال
5. **التحقق من استجابات واجهة برمجة التطبيقات**: تحقق دائمًا من رموز حالة الاستجابة

الأداء

1. **التجميع عند الإمكان**: بالنسبة للعديد من المستخدمين، ضع في اعتبارك الإرسال بالتوازي
2. **المعالجة غير المتزامنة**: إرسال الإشعارات بشكل غير متزامن لتجنب الحظر
3. **تحديد معدل الاستخدام**: تنفيذ تحديد معدل الاستخدام لمنع إساءة استخدام واجهة برمجة التطبيقات
4. **تجميع الاتصالات**: إعادة استخدام اتصالات HTTP لتحسين الأداء

الاختبار

اختبار إعداد مفتاح واجهة برمجة التطبيقات

قبل النشر، تحقق من أن مفتاح واجهة برمجة التطبيقات يعمل:

```
# اختبار المصادقة
curl -X POST "https://yourcrm.com/crm/notification/send" \
  -H "X-API-KEY: your-api-key" \
  -H "Content-Type: application/json" \
  -d '{
    "user_id": 1,
    "type": "custom",
    "title": "اختبار الإشعار",
    "message": "هذه إشعار اختبار للتحقق من مصادقة مفتاح واجهة
برمجة التطبيقات"
  }'
```

الاستجابة المتوقعة: `201 تم الإنشاء` مع تفاصيل الإشعار

الاختبار في التطوير

استخدم معرف مستخدم اختبار للتحقق من ظهور الإشعارات:

```
# إرسال إشعار اختبار
response = send_notification(
    user_id=1,  # مستخدم الاختبار الخاص بك
    notification_type="custom",
    title="اختبار التكامل",
    message="اختبار تكامل واجهة برمجة التطبيقات للإشعارات"
)

print(f"معرف الإشعار: {response['notification']
['notification_id']}")
```

كمستخدم ذلك وتحقق من ظهور الإشعار في قائمة OmniCRM ثم قم بتسجيل الدخول إلى الإشعارات المنسدلة.

استكشاف الأخطاء وإصلاحها

المشكلة: 401 غير مصرح

- **السبب:** مفتاح واجهة برمجة التطبيقات غير صالح أو مفقود
- **الحل:** تحقق من مفتاح واجهة برمجة التطبيقات في `crm_config.yaml` ورأس الطلب
- **تحقق:** تأكد من أن مفتاح واجهة برمجة التطبيقات لا يقل عن 32 حرفًا

المشكلة: 404 المستخد�� غير موجود

- **السبب:** user_id غير صالح
- **الحل:** تحقق من وجود المستخدم في قاعدة البيانات
- **تحقق:** استعلام عن جدول المستخدمين: `SELECT id FROM users WHERE id = ?`

المشكلة: 400 طلب غير صالح - "user_id مطلوب"

- **السبب:** حقل مطلوب مفقود في الطلب
- **الحل:** تضمين جميع الحقول المطلوبة: user_id، type، title
- **تحقق:** تحقق من هيكل الحمولة JSON

المشكلة: تم إرسال الإشعار ولكن لا يظهر

- **السبب:** المستخدم غير مسجل الدخول أو ذاكرة التخزين المؤقت للمتصفح
- **الحل:** تحديث الصفحة أو مسح الذاكرة المؤقتة
- **تحقق:** استعلام عن قاعدة البيانات: `SELECT * FROM notifications WHERE user_id = ? ORDER BY created_at DESC`

المشكلة: 500 خطأ داخلي في الخادم

- **السبب:** خطأ من جانب الخادم
- **الحل:** تحقق من سجلات خادم CRM للحصول على التفاصيل
- **تحقق:** مراجعة `/var/log/omnicrm/api.log` للأخطاء

ملخص مرجع واجهة برمجة التطبيقات

نقاط النهاية

الطريقة	نقطة النهاية	الوصف	المصادقة
POST	`/crm/notification/send`	إرسال إشعار إلى المستخدم	مفتاح واجهة برمجة التطبيقات أو حامل
GET	`/crm/notification/`	الحصول على إشعارات المستخدم الحالي	حامل
GET	`/crm/notification/unread-count`	الحصول على عدد الإشعارات غير المقروءة	حامل
POST	`/crm/notification/{id}/mark-read`	وضع علامة على الإشعار كمقروء	حامل
POST	`/crm/notification/mark-all-read`	وضع علامة على الكل كمقروء	حامل
DELETE	`/crm/notification/{id}`	حذف الإشعار	حامل

الأذونات المطلوبة

- **إرسال إشعار:** يتطلب مفتاح واجهة برمجة التطبيقات صالح أو دور مسؤول
- **عرض الإشعارات:** يمكن للمستخدم عرض إشعاراته الخاصة فقط
- **إدارة الإشعارات:** يمكن للمستخدم إدارة إشعاراته الخاصة فقط

الوثائق ذات الصلة

- استخدام واجهة برمجة التطبيقات: دليل كامل لمصادقة واجهة برمجة التطبيقات
- نظام تذاكر الدعم: إشعارات التذاكر و@mentions
- نظام التوفير: تكامل Ansible
- المراقبة والقياسات: مراقبة النظام والتنبيه
- RBAC: الأدوار والأذونات

سجل التغييرات

يحتوي هذا على آخر 50 تغييرًا تم إجراؤه على مجموعة برامج OmniCRM أو اعتماداته.

ملاحظة: هذا لا يتتبع التغييرات في إعدادات العملاء الفردية.

المراقبة والقياسات

يوفر واجهة برمجة التطبيقات OmniCRM مقاييس شاملة تعتمد على Prometheus لمراقبة أداء التطبيق وعمليات الأعمال والتكاملات الخارجية. يتم عرض جميع المقاييس عند نقطة النهاية `/crm/metrics` بتنسيق عرض Prometheus.

نظرة عامة

يتتبع نظام المقاييس:

- **عمليات التوفير** - تنفيذ الوظائف، المدة، معدلات النجاح/الفشل
- **أداء قاعدة البيانات** - أوقات الاستعلام، صحة مجموعة الاتصالات
- **التكاملات الخارجية** - مكالمات واجهة برمجة التطبيقات Stripe، OCS/CGRateS، Mailjet
- **وظائف الخلفية** - تنفيذ المهام غير المتزامنة والأداء
- **طلبات HTTP** - استخدام نقطة نهاية واجهة برمجة التطبيقات وأوقات الاستجابة (تم إن��اؤها تلقائيًا)

نقطة نهاية المقاييس

URL: `http://your-omnicrm-api:5000/crm/metrics`

التنسيق: تنسيق عرض Prometheus

المصادقة: يمكن لـ Prometheus جمع المقاييس من نقطة النهاية، والتي يمكن الوصول إليها علنًا. في الإنتاج، قم بتقييد الوصول باستخدام قواعد جدار الحماية أو مصادقة الوكيل العكسي.

فئات المقاييس

1. مقاييس التوفير

تتبع مقاييس التوفير تنفيذ دفاتر التشغيل Ansible، وتدير المخزون، وتكوين الأنظمة الخارجية. راجع نظام التوفير و دفاتر تشغيل Ansible لمزيد من التفاصيل.

omnicrm_provision_jobs_total

النوع: عداد

التسميات:

- `status` - حالة إكمال الوظيفة: `success`, `failed`, `running`

الوصف: العدد الإجمالي لوظائف التوفير التي تم إنشاؤها. يتم زيادته عند اكتمال وظائف التوفير بحالتها النهائية.

omnicrm_provision_job_duration_seconds

النوع: هيستوجرام

التسميات:

- `playbook` - اسم دفتر التشغيل Ansible المنفذ
- `status` - حالة إكمال الوظيفة: `success`, `failed`

الدلاء: `[1, 5, 10, 30, 60, 120, 180, 300, 600]` ثواني (1s إلى 10min)

الوصف: الوقت المستغرق لإكمال وظائف التوفير. يسجل مدة تنفيذ دفتر التشغيل بالكامل من البداية إلى النهاية.

omnicrm_provision_jobs_active

النوع: مقياس

الوصف: عدد وظائف التوفير التي تعمل حاليًا. يتم زيادته عند بدء وظيفة، ويتم تقليله عند اكتمالها.

omnicrm_provision_tasks_total

النوع: عداد

التسميات:

- `playbook` - اسم دفتر التشغيل Ansible
- `status` - نتيجة المهمة: `ok`, `failed`, `ignored`

المنفذة ضمن دفاتر التشغيل. يتم زيادته لكل مهمة فردية Ansible العدد الإجمالي لمهام **الوصف:** تكتمل (نجاح أو فشل).

omnicrm_provision_errors_total

عداد **النوع:**

التسميات:

- `error_type` - نوع الخطأ: `fatal`, `task_failed`, `timeout`
- `playbook` - اسم دفتر التشغيل Ansible

العدد الإجمالي ��أخطاء التوفير حسب النوع. يتم زيادته عندما تفشل مهام التوفير أو **الوصف:** تحدث أخطاء قاتلة أثناء التنفيذ.

---

2. مقاييس قاعدة البيانات

تراقب مقاييس قاعدة البيانات أداء الاستعلام وصحة مجموعة الاتصالات. يستخدم OmniCRM مع أدوات القياس التلقائية. راجع معمارية النظام للحصول على تفاصيل نموذج SQLAlchemy البيانات.

omnicrm_db_query_duration_seconds

هيستوجرام **النوع:**

التسميات:

- `operation` - نوع عملية SQL: `SELECT`, `INSERT`, `UPDATE`, `DELETE`
- `table` - اسم جدول قاعدة البيانات

ثواني `[10.0 ,5.0 ,1.0 ,0.5 ,0.1 ,0.05 ,0.01 ,0.005 ,0.001]` **الدلاء:**

وقت تنفيذ استعلام قاعدة البيانات. يتم تتبعه تلقائيًا عبر مستمعي أحداث SQLAlchemy **الوصف:** على كل استعلام.

omnicrm_db_pool_size

مقياس **النوع:**

الوصف: الحجم الحالي لمجموعة اتصالات قاعدة البيانات. العدد الإجمالي للاتصالات في المجموعة (المستعارة والمتاحة).

omnicrm_db_pool_overflow

النوع: مقياس

الوصف: تجاوز مجموعة اتصالات قاعدة البيانات الحالية. عدد الاتصالات التي تم إنشاؤها بما يتجاوز الحد الطبيعي لحجم المجموعة.

omnicrm_db_pool_connections_checked_out

النوع: مقياس

الوصف: عدد الاتصالات المستعارة حاليًا من المجموعة والمستخدمة بواسطة كود التطبيق.

omnicrm_db_errors_total

النوع: عداد

التسميات:

- `error_type` - نوع خطأ قاعدة البيانات
- `operation` - العملية التي تسببت في الخطأ: `connection_error`, `query_error`, إلخ.

الوصف: العدد الإجمالي لأخطاء قاعدة البيانات.

الحالة: محددة ولكن لا تستخدم بنشاط في الكود الحالي (محجوزة للاستخدام في المستقبل)

---

3. مقاييس OCS/CGRateS

تتبع مقاييس OCS (نظام الشحن عبر الإنترنت) التفاعلات مع محرك الفوترة CGRateS. راجع نظرة عامة على الفوترة CGRateS و تكامل لمزيد من التفاصيل.

omnicrm_ocs_api_calls_total

النوع: عداد

التسميات:

- `method` - طريقة واجهة برمجة التطبيقات OCS: `GetBalance`, `SetBalance`, `SetAccount`, `RemoveAccount`, إلخ.
- `status` - نتيجة المكالمة: `success`, `failed`

الوصف: العدد الإجمالي لمكالمات واجهة برمجة التطبيقات OCS/CGRateS. يتم زيادته لكل مكالمة واجهة برمجة التطبيقات غير المتزامنة إلى نظام OCS.

omnicrm_ocs_api_duration_seconds

النوع: هيستوجرام

التسميات:

- `method` - طريقة واجهة برمجة التطبيقات OCS: `GetBalance`, `SetBalance`, `GetAccount`, إلخ.

الدلاء: `[0.1, 0.25, 0.5, 1.0, 2.0, 5.0, 10.0, 30.0]` ثواني

الوصف: مدة مكالمة واجهة برمجة التطبيقات OCS/CGRateS. يسجل الوقت المستغرق لكل مكالمة واجهة برمجة التطبيقات بما في ذلك زمن انتقال الشبكة.

omnicrm_ocs_api_errors_total

النوع: عداد

التسميات:

- `method` - طريقة واجهة برمجة التطبيقات OCS التي فشلت
- `error_type` - فئة الخطأ: `timeout`, `connection_error`, `json_error`, `http_error`, إلخ.

الوصف: العدد الإجمالي لأخطاء واجهة برمجة التطبيقات OCS/CGRateS. يتم زيادته عندما تفشل مكالمات واجهة برمجة التطبيقات OCS مع أنواع أخطاء محددة.

omnicrm_ocs_balance_queries_total

النوع: عداد

التسميات:

- `query_type` - نوع استعلام الرصيد: `single_account`, `multiple_accounts`

يستخدم لتتبع استخدام عمليات استعلام OCS. **الوصف:** العدد الإجمالي لاستعلامات الرصيد إلى الرصيد.

omnicrm_ocs_action_plan_operations_total

النوع: عداد

التسميات:

- `operation` - عملية خطة العمل: `create`، `remove`، `query`

الوصف: العدد الإجمالي لعمليات خطة العمل. تتبع إنشاء وإزالة واستعلام خطط العمل CGRateS للرسوم المتكررة.

---

4. مقاييس مدفوعات Stripe

تتبع مقاييس Stripe عمليات معالجة المدفوعات. راجع دليل نظام المدفوعات و طرق الدفع للحصول على تفاصيل التكامل.

omnicrm_stripe_api_calls_total

النوع: عداد

التسميات:

- `operation` - عملية Stripe: `create_customer`، `charge`، `refund`، `update_payment_method`، إلخ.
- `status` - نتيجة العملية: `success`، `failed`

الوصف: العدد الإجمالي لمكالمات واجهة برمجة التطبيقات Stripe. يتم زيادته لكل عملية معالجة مدفوعات.

omnicrm_stripe_api_duration_seconds

النوع: هيستوجرام

التسميات:

- `operation` - نوع عملية Stripe

الدلاء: `[0.1, 0.5, 1.0, 2.0, 5.0, 10.0, 30.0]` ثواني

الوصف: مدة مكالمة واجهة برمجة التطبيقات Stripe بما في ذلك زمن انتقال الشبكة.

omnicrm_stripe_api_errors_total

النوع: عداد

التسميات:

- `operation` - عملية Stripe التي فشلت
- `error_type` - فئة الخطأ: `card_declined`, `network_error`, `api_error`, إلخ.

الوصف: العدد الإجمالي لأخطاء واجهة برمجة التطبيقات Stripe. يتم زيادته عندما تفشل عمليات الدفع.

omnicrm_stripe_payment_amount_cents

النوع: عداد

التسميات:

- `payment_type` - اتجاه الدفع: `charge`, `refund`

الوصف: إجمالي مبلغ الدفع المعالج عبر Stripe بالسنتات. مفيد لتتبع حجم المعاملات والإيرادات.

omnicrm_stripe_payment_failures_total

النوع: عداد

التسميات:

- `reason` - سبب الفشل: `card_declined`, `insufficient_funds`, `expired_card`, إلخ.

الوصف: العدد الإجمالي لفشل المدفوعات عبر Stripe مصنفة حسب رمز الرفض.

---

5. مقاييس بريد Mailjet

تتبع مقاييس Mailjet تكامل Mailjet عمليات تسليم البريد الإلكتروني. راجع للحصول على تفاصيل التكوين.

omnicrm_mailjet_api_calls_total

النوع: عداد

التسميات:

- `email_type` - نوع قالب البريد الإلكتروني: `welcome`, `user_welcome`, `invoice`, `notification`
- `status` - نتيجة التسليم: `success`, `failed`

الوصف: العدد الإجمالي لمكالمات واجهة برمجة التطبيقات Mailjet. يتم تتبعه عبر الزخرفة `@track_mailjet_call`.

omnicrm_mailjet_api_duration_seconds

النوع: هيستوجرام

التسميات:

- `email_type` - نوع قالب البريد الإلكتروني

الدلاء: `[0.1, 0.5, 1.0, 2.0, 5.0, 10.0, 30.0]` ثواني

الوصف: مدة مكالمة واجهة برمجة التطبيقات Mailjet. الوقت المستغرق لتقديم البريد الإلكتروني عبر واجهة برمجة التطبيقات Mailjet (ليس وقت التسليم).

omnicrm_mailjet_api_errors_total

النوع: عداد

التسميات:

- `email_type` - نوع قالب البريد الإلكتروني
- `error_type` - فئة الخطأ

الوصف: العدد الإجمالي لأخطاء واجهة برمجة التطبيقات Mailjet. يتم زيادته عندما تفشل إرسال البريد الإلكتروني.

omnicrm_mailjet_emails_sent_total

النوع: عداد

التسميات:

- `email_type` - نوع قالب البريد الإلكتروني
- `template_id` - معرف قالب Mailjet

الوصف: العدد الإجمالي للبريد الإلكتروني الذي تم إرساله بنجاح عبر Mailjet. يختلف عن مكالمات واجهة برمجة التطبيقات حيث يمكن أن ترسل مكالمة واحدة إلى عدة مستلمين.

omnicrm_mailjet_email_recipients_total

النوع: عداد

التسميات:

- `email_type` - نوع قالب البريد الإلكتروني

الوصف: العدد الإجمالي لمستلمي البريد الإلكتروني عبر جميع رسائل البريد الإلكتروني المرسلة.

---

6. مقاييس وظائف الخلفية

تتبع مقاييس وظائف الخلفية العمليات غير المتزامنة مثل سلاسل دفاتر التشغيل والمهام المجدولة. راجع نظام التوفير للحصول على تفاصيل وظائف الخلفية.

omnicrm_background_jobs_total

النوع: عداد

التسميات:

- `job_type` - فئة الوظيفة: `playbook_single`, `playbook_chain`, `periodic_task`

الوصف: العدد الإجمالي لوظائف الخلفية التي بدأت. يتم تتبعه عبر مدير السياق `BackgroundJobTimer`.

omnicrm_background_jobs_active

النوع: مقياس

التسميات:

- `job_type` - فئة الوظيفة

الوصف: عدد وظائف الخلفية التي تعمل حاليًا. يتم زيادته عند بدء الوظيفة، ويتم تقليله عند اكتمال الوظيفة.

omnicrm_background_job_duration_seconds

النوع: هيستوجرام

التسميات:

- `job_type` - فئة الوظيفة
- `status` - نتيجة الوظيفة: `success`, `failed`

الدلاء: `[1, 5, 10, 30, 60, 120, 180, 300, 600, 1800, 3600]` ثواني (1s إلى 1hr)

الوصف: وقت تنفيذ وظيفة الخلفية. يشمل المدة الكاملة للعمليات متعددة الخطوات.

omnicrm_background_job_errors_total

النوع: عداد

التسميات:

- `job_type` - فئة الوظيفة
- `error_type` - فئة الخطأ

الوصف: العدد الإجمالي لأخطاء وظائف الخلفية. يتم زيادته عندما تفشل وظائف الخلفية مع استثناءات.

---

7. مقاييس HTTP لـ Flask (تم إنشاؤها تلقائيًا)

يتم إنشاء هذه المقاييس تلقائيًا بواسطة مكتبة `prometheus-flask-exporter` وتتعقب جميع طلبات HTTP إلى واجهة برمجة التطبيقات.

flask_http_request_duration_seconds

النوع: هيستوجرام

التسميات:

- `method` - طريقة HTTP: `GET`، `POST`، `PUT`، `DELETE`، إلخ.
- `endpoint` - اسم مسار Flask
- `status` - رمز الحالة HTTP

الوصف: مدة طلب HTTP لجميع نقاط نهاية واجهة برمجة التطبيقات. يتم قياسها تلقائيًا.

flask_http_request_total

النوع: عداد

التسميات:

- `method` - طريقة HTTP
- `endpoint` - اسم مسار Flask
- `status` - رمز الحالة HTTP

الوصف: إجمالي طلبات HTTP حسب نقطة النهاية والطريقة ورمز الحالة.

flask_http_request_exceptions_total

النوع: عداد

التسميات:

- `method` - طريقة HTTP
- `endpoint` - اسم مسار Flask

الوصف: العدد الإجمالي للاستثناءات غير المعالجة في طلبات HTTP. يشير إلى الأخطاء أو الأخطاء غير المتوقعة.

---

8. مقاييس أخطاء واجهة برمجة التطبيقات (محجوزة)

تم تعريف هذه المقاييس ولكن لم يتم قياسها حاليًا. تم حجزها للاستخدام في المستقبل.

omnicrm_api_errors_total

النوع: عداد

التسميات:

- `endpoint` - نقطة نهاية واجهة برمجة التطبيقات
- `error_type` - فئة الخطأ
- `status_code` - رمز الحالة HTTP

الحالة: محددة ولكن لا تستخدم بنشاط

omnicrm_api_auth_failures_total

النوع: عداد

التسميات:

- `auth_method` - طريقة المصادقة: `jwt`, `api_key`, `ip_whitelist`
- `reason` - سبب الفشل

الحالة: محددة ولكن لا تستخدم بنشاط

---

9. مقياس معلومات التطبيق

omnicrm_api

النوع: معلومات

التسميات:

- `version` - سلسلة إصدار واجهة برمجة التطبيقات
- `environment` - اسم البيئة: `production`, `staging`, `development`

يتم تعيينه مرة واحدة عند بدء OmniCRM. **الوصف:** مقياس معلومات واجهة برمجة التطبيقات التطبيق مع معلومات الإصدار والبيئة.

التحديثات الدورية

update_db_pool_metrics(engine)

يتم استدعاؤه تلقائيًا كل 30 ثانية لتحديث مقاييس مجموعة اتصالات قاعدة البيانات.

تكوين Prometheus

تكوين جمع البيانات

أضف إلى ملف OmniCRM `prometheus.yml` الخاص بك:

```yaml
scrape_configs:
  - job_name: 'omnicrm-api'
    scrape_interval: 15s
    scrape_timeout: 10s
    static_configs:
      - targets: ['omnicrm-api:5000']
    metrics_path: '/crm/metrics'
```

أمثلة على التنبيهات

معدل فشل التوفير العالي

```yaml
- alert: HighProvisionFailureRate
  expr: |
    (
      rate(omnicrm_provision_jobs_total{status="failed"}[5m]) /
      rate(omnicrm_provision_jobs_total[5m])
    ) > 0.1
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "معدل فشل وظائف التوفير مرتفع"
    description: "{{ $value | humanizePercentage }} من وظائف التوفير تفشل"
```

استنفاد مجموعة اتصالات قاعدة البيانات

```yaml
- alert: DatabasePoolExhausted
  expr: omnicrm_db_pool_overflow > 0
  for: 2m
  labels:
    severity: critical
  annotations:
    summary: "تم الكشف عن تجاوز مجموعة اتصالات قاعدة البيانات"
    description: "تستخدم مجموعة الاتصالات اتصالات تجاوز، قد يشير إلى أن حجم المجموعة صغير جدًا"
```

استعلامات قاعدة بيانات بطيئة

```yaml
- alert: SlowDatabaseQueries
  expr: |
    histogram_quantile(0.99,
      rate(omnicrm_db_query_duration_seconds_bucket[5m])
    ) > 1.0
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "تم الكشف عن استعلامات قاعدة بيانات بطيئة"
    description: "وقت استعلام النسبة المئوية 99 هو {{ $value }}s"
```

معطلة OCS واجهة برمجة التطبيقات

```
- alert: OCSAPIDown
  expr: |
    (
      rate(omnicrm_ocs_api_calls_total{status="failed"}[5m]) /
      rate(omnicrm_ocs_api_calls_total[5m])
    ) > 0.5
  for: 2m
  labels:
    severity: critical
  annotations:
    summary: "حرج OCS معدل فشل واجهة برمجة التطبيقات"
    description: "{{ $value | humanizePercentage }} من مكالمات
تفشل OCS واجهة برمجة التطبيقات"
```

مشاكل مدفوعات Stripe

```
- alert: StripePaymentFailures
  expr: rate(omnicrm_stripe_payment_failures_total[5m]) > 5
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "ارتفاع فشل مدفوعات Stripe"
    description: "{{ $value }} فشل في الدفع في الثانية"
```

أفضل الممارسات

استراتيجية المراقبة

1. **إعداد تنبيهات أساسية** - تكوين تنبيهات للمقاييس الحرجة:
    - معدل فشل التوفير > 10%
    - استنفاد مجموعة اتصالات قاعدة البيانات
    - فشل واجهة برمجة التطبيقات OCS/CGRateS

    - أخطاء معالجة مدفوعات Stripe

2. **تتبع المقاييس التجارية** - مراقبة مؤشرات الأداء الرئيسية التشغيلية:
    - الإيرادات المعالجة عبر Stripe
    - إنتاجية التو��ير
    - معدلات تسليم البريد الإلكتروني للعملاء
3. **مراقبة الأداء** - مراقبة التدهور:
    - النسب المئوية لوقت استجابة واجهة برمجة التطبيقات
    - أداء استعلام قاعدة البيانات
    - زمن انتقال واجهة برمجة التطبيقات الخارجية
4. **تخطيط السعة** - استخدام المقاييس للتنبؤ:
    - حجم مجموعة اتصالات قاعدة البيانات
    - توسيع عمال وظائف الخلفية
    - سعة خادم واجهة برمجة التطبيقات

احتفاظ بالمقاييس

توصيات احتفاظ Prometheus:

- **15 يومًا** - مقاييس عالية الدقة (فترة جمع 15 ثانية)
- **90 يومًا** - مقاييس منخفضة الدقة (تجميع 5 دقائق)
- **سنة واحدة** - مقاييس مجمعة على المدى الطويل (تجميع ساعة واحدة)

استخدم Thanos أو Cortex أو VictoriaMetrics للتخزين على المدى الطويل والاستعلام العالمي.

استكشاف الأخطاء وإصلاحها

المقاييس غير الظاهرة

يتم عرض نقطة النهاية `/metrics` بشكل افتراضي، فقط للمصادر الداخلية (غير العامة). يمكنك تغيير ذلك في تكوين nginx الخاص بك إذا لزم الأمر.

تحقق من نقطة نهاية المقاييس:

```
curl http://localhost:5000/crm/metrics
```

تحقق مما إذا كان Prometheus يمكنه الجمع:

```
# تحقق من صفحة أهداف Prometheus
http://prometheus:9090/targets
```

المقاييس المحددة المفقودة

بعض المقاييس لا يتم إنشاؤها إلا بعد استخدامها لأول مرة:

- تظهر مقاييس التوفير بعد أول وظيفة توفير
- تظهر مقاييس Stripe بعد أول دفعة
- تظهر مقاييس OCS بعد أول عملية فوترة

مشاكل الكاردينالية العالية

إذا كان Prometheus العالية الكاردينالية ذات التسميات من تحقق، زائدة ذاكرة يستهلك أو بطيئًا:

```
# احسب عدد السلاسل الزمنية الفريدة لكل مقياس
count by (__name__) ({__name__=~".+"})
```

المقاييس التي تحتوي على أكثر من 10,000 سلسلة قد تشير إلى مشكلة كاردينالية.

ملخص المقاييس

إجمالي المقاييس: 34 (تم إنشاؤها تلقائيًا Flask 31 مخصص + 3 مقاي��س)

حسب النوع:

- **عدادات:** 20
- **مقاييس:** 5

- **هيستوجرامات:** 8
- **معلومات:** 1

حسب الفئة:

- التوفير: 5 مقاييس
- قاعدة البيانات: 5 مقاييس
- OCS/CGRateS: 5 مقاييس
- Stripe: 5 مقاييس
- Mailjet: 5 مقاييس
- وظائف الخلفية: 4 مقاييس
- HTTP/Flask: 3 مقاييس
- معلومات التطبيق: مقياس واحد
- محجوز (للاستخدام في المستقبل): 2 مقياس

الوثائق ذات الصلة

- معمارية النظام - تصميم النظام العام وعلاقات المكونات
- نظام التوفير - سير عمل التوفير وتنفيذ الوظائف
- دفاتر تشغيل Ansible - هيكل دفتر التشغيل والتنفيذ
- وثائق واجهة برمجة التطبيقات - نقاط نهاية واجهة برمجة التطبيقات والمصادقة
- نظرة عامة على الفوترة - نظام الفوترة والشحن
- دليل نظام المدفوعات Stripe ومعالجة المدفوعات - تكامل
- تكامل Mailjet - تكوين خدمة البريد الإلكتروني
- تكوين الإدارة - خيارات تكوين النظام

إدارة طرق الدفع

يتيح نظام طرق الدفع في OmniCRM إدارة بطاقات الدفع بأمان باستخدام للعملاء والموظفين **معالجة الدفع متعددة البائعين** (Stripe، PayPal، إلخ). تتيح طرق الدفع الفوترة التلقائية للخدمات، والمدفوعات لمرة واحدة، والرسوم المتكررة دون تخزين بيانات البطاقة الحساسة في OmniCRM.

انظر أيضًا: دليل نظام الدفع، نظرة عامة على الفواتير، معالجة الدفع، الفواتير.

نظرة عامة

يوفر نظام طرق الدفع:

- **تخزين آمن للبطاقات** (Stripe، PayPal)، ولا يتم تخزينها أبدًا في OmniCRM - يتم تحويل بطاقات الدفع بواسطة البائعين
- **دعم متعدد البائعين** - دعم Stripe وPayPal من ورق الدفع ��
- **بطاقات متعددة** - يمكن للعملاء تخزين عدة طرق دفع
- **اختيار افتراضي** - تعيين طريقة الدفع المفضلة للرسوم التلقائية
- **تتبع انتهاء الصلاحية** - مراقبة وتحديث البطاقات التي على وشك الانتهاء
- **خدمة ذاتية** - يمكن للعملاء إدارة بطاقاتهم الخاصة عبر بوابة الرعاية الذاتية
- **إدارة الموظفين** - يمكن لموظفي الدعم إضافة/إزالة بطاقات نيابة عن العملاء

طرق الدفع المدعومة:

- **البطاقات** (عبر Stripe أو PayPal)
  - بطاقات الائتمان (Visa، Mastercard، American Express، Discover)
  - بطاقات الخصم
  - بطاقات مسبقة الدفع (إذا كانت مدعومة من شبكة البطاقة)
- **حسابات PayPal** (عبر تكامل PayPal)

غير مخزنة في OmniCRM:

تخزن OmniCRM فقط تفاصيل البطاقة يتم تحويلها بواسطة بائعي الدفع وتخزينها بأمان:

- بائع الدفع (stripe، paypal)

- علامة البطاقة (Visa، Mastercard، إلخ)
- آخر 4 أرقام
- شهر/سنة انتهاء الصلاحية
- اسم/لقب حامل البطاقة
- رمز طريقة الدفع المحدد من البائع

الوصول إلى طرق الدفع

من صفحة العميل:

1. انتقل إلى **العملاء → [اختر العميل]**
2. انقر على علامة التبويب **الفوترة**
3. انتقل إلى قسم **طرق الدفع**

أو مباشرة:

من لوحة معلومات البطاقات التي على وشك الانتهاء:

عرض جميع العملاء الذين لديهم بطاقات على وشك الانتهاء:

هذا يعرض قائمة شاملة للبطاقات التي ستنتهي صلاحيتها خلال الـ 60 يومًا القادمة.

قائمة طرق الدفع

تعرض جدول طرق الدفع جميع البطاقات المخزنة لعميل:

وصف الأعمدة:

- **اللقب** - الاسم الودي للبطاقة (مثل "بطاقة شخصية"، "فيزا العمل")
- **المصدر** - علامة البطاقة وآخر 4 أرقام
- **الانتهاء** - شهر/سنة انتهاء الصلاحية (بتنسيق MM/YYYY)
- **تم الإضافة** - تاريخ إضافة البطاقة إلى الحساب
- **افتراضي** - علامة تشير إلى طريقة الدفع الافتراضية للرسوم التلقائية

الإجراءات لكل بطاقة:

كل صف يحتوي على قائمة إجراءات (⋮) مع الخيارات:

- **تعيين كافتراضي** - جعل هذه هي طريقة الدفع الافتراضية
- **حذف** - إزالة البطاقة من الحساب

إضافة طريقة دفع

انقر على **"إضافة طريقة دفع"** لفتح نافذة الدفع الآمنة.

الخطوة 1: إدخال تفاصيل البطاقة

تظهر نموذج الدفع الآمن (مدعوم من Stripe Elements أو PayPal SDK):

الحقول المطلوبة:

- **معلومات البطاقة**، CVC (تم التحقق منه بواسطة Stripe) - رقم البطاقة، تاريخ الانتهاء
- **اسم حامل البطاقة** - الاسم الموجود على البطاقة
- **الدولة/المنطقة** - دولة الفوترة

الحقول الاختيارية:

- **لقب البطاقة** - تسمية ودية لتمييز البطاقات

الأمان:

- آمن م��تضاف من قبل البائع iframe يتم إدخال تفاصيل البطاقة مباشرة في (Stripe Elements / PayPal SDK)
- أو تخزن أرقام بطاقات كاملة OmniCRM لا ترى
- بواسطة بائع الدفع PCI DSS يتم التعامل مع الامتثال لمعايير
- يمنع التحقق الفوري من الأرقام غير الصحيحة

الخطوة 2: الإرسال والتحويل إلى رمز

عند النقر على **"إضافة طريقة دفع"**:

1. **التحقق من جانب العميل:**
    - يتحقق بائع الدفع من تنسيق رقم البطاقة

- يتحقق من أن تاريخ الانتهاء في المستقبل
- يتحقق من تنسيق CVC

2. **التحويل إلى رمز:**
   - يتم إرسال تفاصيل البطاقة مباشرة إلى بائع الدفع (ليس إلى OmniCRM)
   - ينشئ البائع رمزًا آمنًا (مثل `pm_1A2B3C4D` لـ Stripe)
   - يتم إرجاع الرمز إلى OmniCRM
3. **معالجة الخادم:**
   - تقوم OmniCRM بحفظ الرمز في سجل العميل مع معرف البائع
   - تخزن آخر 4 أرقام، العلامة، تاريخ الانتهاء، واسم البائع للعرض
   - لا تلمس أرقام البطاقة الكاملة خوادم OmniCRM أبدًا

الخطوة 3: التأك��د

تظهر رسالة النجاح:

> تم إضافة بطاقتك Visa إلى حسابك بـ 1234 التي تنتهي.

تظهر البطاقة الجديدة في جدول طرق الدفع.

اختيار افتراضي تلقائي:

- إذا كانت هذه هي البطاقة الأولى للعميل، يقوم النظام تلقائيًا بتعيينها كافتراضية
- إذا كان لدى العميل بطاقات بالفعل، تتم إضافة البطاقة الجديدة كغير افتراضية
- يمكن للعميل تغيير الافتراضي بعد الإضافة

تعيين طريقة الدفع الافتراضية

تستخدم طريقة الدفع الافتراضية لـ:

- رسوم الخدمة المتكررة التلقائية
- مدفوعات الفواتير
- الشحنات وإعادة الشحن
- المعاملات لمرة واحدة (ما لم يُحدد خلاف ذلك)

لتغيير الافتراضي:

1. حدد البطاقة التي تريد تعيينها كافتراضية في جدول طرق الدفع
2. انقر على **قائمة الإجراءات (⋮)** بجوار البطاقة
3. اختر **"تعيين كافتراضي"**
4. تظهر رسالة التأكيد

   التي تنتهي بـ 5678 هي الآن طريقة الدفع الافتراضية الخاصة بك Visa.

تتحرك علامة الاختيار إلى البطاقة المحددة حديثًا.

مؤشر بصري:

تظهر البطاقات الافتراضية:

في عمود الافتراضي، عادةً مع شارة علامة اختيار خضراء.

حذف طريقة الدفع

إزالة البطاقات التي انتهت صلاحيتها، أو فقدت، أو لم تعد مطلوبة.

الخطوة 1: بدء الحذف

1. ابحث عن البطاقة التي تريد حذفها في جدول طرق الدفع
2. انقر على أيقونة **سلة المهملات** في ذلك الصف

إذا كان لدى العميل طريقة دفع واحدة نشطة فقط، فإن أيقونة سلة المهملات **معطلة**. عند التمرير فوقها، تظهر رسالة توضيحية:

> لا يمكن حذف طريقة الدفع الوحيدة لديك. أضف طريقة دفع أخرى أولاً.

أضف طريقة دفع ثانية قبل أن تتمكن من إزالة الطريقة الحالية.

الخطوة 2: تأكيد الحذف

عندما يُسمح بالحذف، تظهر نافذة تأكيد:

> هل أنت متأكد أنك تريد حذف هذه الطريقة للدفع؟

> [حذف طريقة الدفع] [إلغاء]

إذا كانت الطريقة التي يتم حذفها هي الافتراضية الحالية وتوجد طرق نشطة أخرى، تشير النافذة إلى أن طريقة نشطة أخرى سيتم ترقيتها إلى الافتراضي بعد الحذف.

انقر على **"حذف طريقة الدفع"** للتأكيد.

الخطوة 3: اكتمال الحذف

رسالة النجاح:

تمت إزالة البطاقة من الجدول وحذفها من بائع الدفع.

قيود مهمة:

- **لا يمكن حذف الطريقة النشطة الوحيدة** - يتم تعطيل التحكم في الحذف (مع رسالة توضيحية). كما ترفض واجهة برمجة التطبيقات الطلب. أضف طريقة دفع أخرى أولاً.
- **حذف الافتراضي عندما توجد طرق أخرى** - يتم ترقية طريقة دفع نشطة أخرى قابلة للاستخدام إلى الافتراضي تلقائيًا.
- **لا يمكن التراجع** - الحذف دائم؛ يجب على العميل إعادة إضافة الطريقة إذا لزم الأمر

إدارة البطاقات التي على وشك الانتهاء

يتتبع OmniCRM تواريخ انتهاء صلاحية البطاقات ويوفر أدوات لتحديث البطاقات التي على وشك الانتهاء بشكل استباقي.

لوحة معلومات البطاقات التي على وشك الانتهاء

انتقل إلى **الفوترة → البطاقات التي على وشك الانتهاء** لرؤية قائمة شاملة:

> أيام انتهاء صلاحية بطاقة العميل حتى الإجراء جون سميث فيزا **1234 02/2025 12**
> **يومًا تحديث جين دو 45 03/2025 MC5678 يومًا تحديث أكمي كورب**
> **أميكس**9012 01/2025 منتهية الصلاحية تحديث

الفلاتر:

- **نطاق الانتهاء** - الأيام الـ 30/60/90 القادمة أو المنتهية بالفعل
- **نوع العميل** - فرد مقابل عمل
- **نوع الخدمة** - تصفية حسب الخدمة التي تتطلب طريقة دفع

الإجراءات:

- **تحديث** - يفتح صفحة طرق الدفع الخاصة بالعميل لإضافة بطاقة جديدة
- **إخطار** - إرسال تذكير عبر البريد الإلكتروني للعميل (Mailjet إذا تم تكوين)

إشعارات انتهاء الصلاحية

إذا تم تكوين Mailjet، يتم إرسال رسائل بريد إلكتروني تلقائية:

- التذكير الأول - **قبل 60 يومًا من انتهاء الصلاحية**
- التذكير الثاني - **قبل 30 يومًا من انتهاء الصلاحية**
- التحذير النهائي - **قبل 7 أيام من انتهاء الصلاحية**
- إشعار انتهاء صلاحية البطاقة - **عند انتهاء الصلاحية**

يمكن للعملاء النقر على رابط في البريد الإلكتروني لتحديث طريقة الدفع الخاصة بهم عبر بوابة الرعاية الذاتية.

متغيرات قالب البريد الإلكتروني:

تتلقى قوالب Mailjet:

- اسم العميل
- علامة البطاقة وآخر 4 أرقام
- تاريخ انتهاء الصلاحية
- رابط إلى صفحة طرق الدفع في الرعاية الذاتية

انظر `integrations_mailjet` لتكوين قالب البريد الإلكتروني.

تحديث بطاقة على وشك الانتهاء

سير العمل الموصى به:

1. يتلقى العميل إشعار انتهاء الصلاحية عبر البريد الإلكتروني
2. يقوم العميل بتسجيل الدخول إلى بوابة الرعاية الذاتية
3. ينتقل إلى **الفوترة → طرق الدفع**
4. ينقر على **"إضافة طريقة دفع"**
5. يدخل تفاصيل البطاقة الجديدة (نفس البطاقة مع تاريخ انتهاء محدث، أو بطاقة بديلة)
6. يعين البطاقة الجديدة كافتراضية
7. يحذف البطاقة القديمة/المنتهية

سير العمل للموظفين:

إذا اتصل العميل بالدعم:

1. يفتح الموظف حساب العميل
2. ينتقل إلى **الفوترة → طرق الدفع**
3. يضيف بطاقة جديدة نيابة عن العميل (يقدم العميل التفاصيل عبر الهاتف)
4. يعين البطاقة الجديدة كافتراضية
5. يحذف البطاقة المنتهية
6. يؤكد مع العميل

> ⚠ **WARNING**

لا تطلب أبدًا من العملاء إرسال تفاصيل البطاقة عبر البريد الإلكتروني أو الرسائل النصية.
استخدم دائمًا:

- بوابة الرعاية الذاتية الآمنة للخدمة الذاتية
- الهاتف مع الموظف الذي يدخل التفاصيل مباشرة في النظام
- شخصيًا في موقع البيع بالتجزئة

م��ذا يحدث عند انتهاء صلاحية البطاقات

عندما تصل بطاقة الدفع إلى تاريخ انتهاء صلاحيتها ولم يتم تحديثها:

التأثيرات الفورية:

1. **تفشل المدفوعات التلقائية**
   - يرفض بائع الدفع المعاملات مع البطاقات المنتهية
   - تفشل تجديدات الخدمة الشهرية في المعالجة
   - تفشل الشحنات التلقائية
   - تفشل المدفوعات التلقائية للفواتير
2. **إشعارات العملاء**
   - يحاول النظام فرض رسوم على البطاقة
   - يتم إرسال إشعار بفشل الدفع
   - يتم إرسال بريد إلكتروني "تحديث طريقة الدفع" مع رابط إلى بوابة الرعاية الذاتية
3. **تغييرات حالة الخدمة**
   - **الخدمات المدفوعة لاحقًا** - قد تستمر مؤقتًا مع رصيد مستحق
   - **الخدمات المدفوعة مسبقًا** - تعليق الخدمة عند نفاد الرصيد
   - **الخدمات المتجددة تلقائيًا** - تفشل التجديد، قد تنتهي الخدمة

الإجراءات اللاحقة:

اليوم 1-3 (فترة السماح):

- تستمر الخدمة بشكل طبيعي
- يتلقى العميل أول إشعار بفشل الدفع
- يحاول النظام إعادة المحاولة (اعتمادًا على التكوين)

اليوم 4-7:

- محاولة دفع ثانية (إذا تم تكوينها)
- يتم إرسال بريد إلكتروني تحذيري
- قد يتصل خدمة العملاء بالعميل

اليوم 8-14:

- قد يتم تعليق الخدمة لعدم الدفع
- يمنع الحالة المعلقة الاستخدام ولكن يحفظ الحساب
- يمكن للعميل الاستعادة عن طريق تحديث طريقة الدفع ودفع الرصيد المستحق

اليوم 15+:

- قد يتم إنهاء الخدمة لعدم الدفع
- يتم وضع المخزون (المعدات، SIM بطاقات) للعودة
- يتم إرسال إشعار نهائي
- يتم إحالة الحساب إلى التحصيل (إذا كان ذلك مناسبًا)

منع انقطاع الخدمة:

لتجنب انقطاع الخدمة:

- تحديث البطاقات **قبل 30 يومًا من انتهاء الصلاحية**
- إضافة طرق دفع متعددة للحد من المخاطر
- تمكين تنبيهات فشل الدفع
- مراقبة لوحة معلومات البطاقات التي على وشك الانتهاء أسبوعيًا

استعادة الخدمة بعد انتهاء الصلاحية:

إذا تم تعليق الخدمة بسبب بطاقة منتهية:

1. أضف طريقة دفع جديدة صالحة
2. عيّنها كافتراضية
3. ادفع الرصيد المستحق (إن وجد)
4. اتصل بالدعم لإعادة تفعيل الخدمة
5. يتم استعادة الخدمة خلال دقائق إلى ساعات

أمان طريقة الدفع

التحويل إلى رمز

يستخدم OmniCRM تحويل الرموز من البائع لضمان الأمان:

1. **يدخل العميل البطاقة** → يتم إرسالها مباشرة إلى خوادم بائع الدفع
2. **يتحقق البائع ويحول إلى رمز** → ينشئ رمزًا فريدًا
3. **يتم تخزين الرمز في OmniCRM** → لا يتم تخزين رقم البطاقة الكامل أبدًا
4. **معالجة الدفع** → يتم إرسال الرمز إلى البائع، يقوم البائع بفرض رسوم على البطاقة

ما تخزنه OmniCRM:

```
{
  "vendor": "stripe",
  "vendor_payment_method_id": "pm_1A2B3C4D5E6F",
  "payment_type": "card",
  "brand": "visa",
  "last4": "1234",
  "exp_month": 12,
  "exp_year": 2026,
  "name": "جون سميث",
  "nickname": "بطاقة شخصية",
  "is_default": true
}
```

ما لا تخزنه OmniCRM:

- رقم البطاقة الكامل
- رمز CVV/CVC
- بيانات الشريط المغناطيسي
- أرقام التعريف الشخصي

الامتثال لمعايير PCI

من خلال استخدام نماذج الدفع المستضافة من قبل البائع:

- أبدًا OmniCRM **مخفض** - بيانات البطاقة لا تلمس خوادم **نطاق PCI**

- الآمن للبائع iframe - يتم إدخال البطاقة في **حقول مستضافة من قبل البائع**
- **لا تخزين للبطاقة** - يتم استخدام الرموز بدلاً من بيانات البطاقة الخام
- HTTPS/TLS - جميع الاتصالات عبر **نقل آمن**

انظر دليل نظام الدفع للحصول على تفاصيل أمان بائع الدفع.

سير العمل الشائع

سير العمل 1: العميل يضيف أول طريقة دفع

السيناريو: عميل جديد يسجل للحصول على الخدمة

1. يقوم العميل بإنشاء حساب
2. يختار خطة الخدمة
3. يتم تحفيزه لإضافة طريق�ة دفع أثناء الخروج
4. يدخل تفاصيل البطاقة في نافذة Stripe
5. يتم تحويل البطاقة وحفظها
6. يتم تعيينها تلقائيًا كافتراضية
7. يتم توفير الخدمة
8. تتم معالجة أول رسوم

سير العمل 2: العميل يحدث بطاقة على وشك الانتهاء

السيناريو: بطاقة الائتمان على وشك الانتهاء

1. يتلقى العميل إشعارًا عبر البريد الإلكتروني (قبل 60 يومًا من انتهاء الصلاحية)
2. يسجل الدخول إلى بوابة الرعاية الذاتية
3. ينتقل إلى **الفوترة → طرق الدفع**
4. يراجع البطاقة الحالية التي ستنتهي في 12/2025
5. ينقر على **"إضافة طريقة دفع"**
6. يدخل بطاقة بديلة بتاريخ انتهاء 12/2028
7. يعين البطاقة الجديدة كافتراضية
8. يحذف البطاقة القديمة
9. يتم إرسال بريد إلكتروني للتأكيد

سير العمل 3: موظف يساعد العميل عبر الهاتف

السيناريو: يتصل العميل: "تم رفض بطاقتي"

1. يتصل العميل بالدعم
2. يتحقق الموظف من الهوية (أسئلة الأمان)
3. يتحقق الموظف من طرق الدفع: البطاقة منتهية في 01/2025
4. الموظف: "لقد انتهت صلاحية بطاقتك. هل لديك بطاقة جديدة؟"
5. يقدم العميل تفاصيل البطاقة الجديدة عبر الهاتف
6. ينتقل الموظف إلى **العملاء → [العميل] → الفوترة**
7. ينقر على **"إضافة طريقة دفع"**
8. يدخل تفاصيل البطاقة كما يقرأها العميل
9. يعين البطاقة الجديدة كافتراضية
10. يحذف البطاقة المنتهية
11. يعيد محاولة الدفع الفاشل
12. يؤكد مع العميل: "تم الدفع بنجاح، وتم استعادة الخدمة"

سير العمل 4: عميل تجاري لديه بطاقات متعددة

السيناريو: تريد الشركة بطاقات مختلفة لأغراض مختلفة

1. يضيف العميل التجاري بطاقة رئيسية (Visa 1111 تنتهي بـ)
2. يتم تعيينها كافتراضية لرسوم الخدمة الشهرية
3. يضيف بطاقة احتياطية (Mastercard 2222 تنتهي بـ) للشحنات
4. يضيف بطاقة شراء (Amex 3333 تنتهي بـ) لشراء المعدات
5. يدويًا عند الخروج Mastercard عند إجراء الشحن، يختار
6. الافتراضية تستخدم للفوترة الشهرية التلقائية Visa لا تزال

سير العمل 5: إدارة البطاقات التي على وشك الانتهاء (الإدارة)

السيناريو: إدارة انتهاء الصلاحية بشكل استباقي

1. ينتقل المدير إلى **الفوترة → البطاقات التي على وشك الانتهاء**
2. الفلاتر: "الأيام الـ 30 القادمة"

3. يرى 15 عميلًا لديهم بطاقات على وشك الانتهاء
4. **"إرسال رسائل تذكير عبر البريد الإلكتروني"** ← يحدد الكل
5. يرسل Mailjet رسائل بريد إلكتروني مخصصة لكل عميل
6. يقوم العملاء بتحديث بطاقاتهم عبر الرعاية الذاتية
7. يراجع المدير القائمة بعد أسبوع
8. يتصل بالعملاء المتبقيين الذين لم يقوموا بالتحديث
9. يساعد في تحديث البطاقات عبر الهاتف

استكشاف الأخطاء وإصلاحها

"تم رفض البطاقة" عند إضافة ط��يقة الدفع

- **السبب:** رفض بائع الدفع البطاقة (نقص في الأموال، منع الاحتيال، رفض المصدر)
- **الإصلاح:**
  - جرب بطاقة مختلفة
  - اتصل بمصدر البطاقة لتفويض المعاملة
  - تأكد من أن البطاقة تدعم المشتريات عبر الإنترنت
  - تحقق من أن عنوان الفوترة يتطابق مع البطاقة المسجلة

"خطأ في إضافة طريقة الدفع" (خطأ عام)

- **السبب:** خطأ في واجهة برمجة تطبيقات بائع الدفع أو مشكلة في الشبكة
- **الإصلاح:**
  - قم بتحديث الصفحة وحاول مرة أخرى
  - تحقق من اتصال الإنترنت
  - تحقق من أن تكوين بائع الدفع صحيح في إعدادات النظام
  - تحقق من وحدة تحكم المتصفح للحصول على رسالة خطأ محددة
  - جرب متصفحًا مختلفًا (قم بتعطيل مانعات الإعلانات)

لا يمكن حذف طريقة الدفع (زر معطل)

- **السبب:** إنها الطريقة النشطة الوحيدة للعميل
- **الإصلاح:**
  - أضف طريقة دفع جديدة أولاً، ثم احذف القديمة
  - مرر فوق أيقونة سلة المهملات المعطلة للحصول على شرح الرسالة

تظهر البطاقة على أنها منتهية ولكن ليست في قائمة "البطاقات التي على وشك الانتهاء"

- **السبب:** انتهت صلاحية البطاقة مؤخرًا، لم يتم تحديث التخزين المؤقت
- **الإصلاح:**
  - قم بتحديث الصفحة
  - تحقق من الفلاتر على لوحة معلومات البطاقات التي على وشك الانتهاء
  - قد تنتقل البطاقات المنتهية إلى عرض مختلف

البطاقة الجديدة لا تظهر على الفور

- **السبب:** لم يتم تحديث الصفحة بعد إضافة البطاقة
- **الإصلاح:**
  - يجب أن يتم تحديث جدول طرق الدفع تلقائيًا
  - إذا لم يكن، قم بتحديث المتصفح يدويًا
  - تحقق مما إذا كانت هناك مشكلة أثناء عملية الإضافة

نافذة الدفع لن يتم تحميلها

- **السبب:** عدم تحميل SDK بائع الدفع، مشكلة في مفتاح API، أو ملحق المتصفح يمنع التحميل
- **الإصلاح:**
  - تحقق من وحدة تحكم المتصفح للحصول على أخطاء
  - قم بتعطيل مانعات الإعلانات وحماية التتبع
  - تحقق من تكوين بائع الدفع في إعدادات النظام
  - تأكد من تحميل برنامج SDK (تحقق من علامة التبويب الشبكة) الخاص بالبائع
  - جرب وضع التصفح المتخفي/الخاص

العميل لا يتلقى إشعارات انتهاء الصلاحية

- **السبب:** لم يتم تكوين Mailjet أو القالب مفقود
- **الإصلاح:**
  - تحقق من بيانات اعتماد Mailjet في *crm_config.yaml*
  - تحقق من وجود قالب البريد الإلكتروني لبطاقة انتهاء الصلاحية
  - تأكد من أن عنوان البريد الإلكتروني للعميل صالح
  - تحقق من سجلات Mailjet لفشل التسليم

أفضل الممارسات

للعملاء:

- أضف طريقة الدفع قبل تفعيل الخدمة لتجنب التأخيرات
- احتفظ على الأقل بـ 2 بطاقات في الملف للحد من المخاطر
- تحديث البطاقات المنتهية بعد 30 يومًا أو أكثر من انتهاء الصلاحية
- احذف البطاقات القديمة/المنتهية لتجنب الارتباك
- استخدم ألقاب وصفية ("فيزا شخصية"، "أميكس عمل")
- تحقق من أن طريقة الدفع الافتراضية صحيحة للفوترة التلقائية

لموظفي الدعم:

- تحقق من هوية العميل قبل الوصول إلى طرق الدفع
- لا تطلب أبدًا من العملاء إرسال تفاصيل البطاقة عبر البريد الإلكتروني/SMS/chat
- قم بمعالجة إضافات البطاقة على الفور أثناء المكالمات (لا تؤجل)
- تأكد من تعيين البطاقة الجديدة كافتراضية بعد الإضافة
- احذف البطاقات القديمة فقط بعد التأكد من أن البطاقة الجديدة تعمل
- اختبر الدفع بعد تحديث البطاقة المنتهية (قم بمعالجة تفويض بقيمة 0.01 جنيه إسترليني)

للمسؤولين:

- راقب لوحة معلومات البطاقات التي على وشك الانتهاء أسبوعيًا
- أرسل رسائل تذكير عبر البريد الإلكتروني قبل 60/30/7 يومًا من انتهاء الصلاحية
- احتفظ بمفاتيح اختبار/حياة بائع الدفع منفصلة للتطوير مقابل الإنتاج
- تأكد من تكوين قوالب Mailjet لإشعارات انتهاء الصلاحية
- راجع تقارير الدفع الفاشلة لتحديد البطاقات المنتهية
- درب الموظفين على إجراءات التعامل الآمن مع البطاقات

الوثائق ذات الصلة

- دليل نظام الدفع - مرجع كامل لواجهة برمجة تطبيقات الدفع والهندسة المعمارية
- `payments_process` - معالجة المدفوعات باستخدام طرق الدفع المخزنة
- `payments_invoices` - الدفع التلقائي للفواتير باستخدام البطاقة الافتراضية

- `features_topup_recharge` - نظام الشحن باستخدام طرق الدفع
- `basics_payment` - مفاهيم الدفع والفوترة العامة
- `customer_care` - بوابة الرعاية الذاتية للعملاء لإدارة بطاقاتهم الخاصة

دليل واجهة برمجة تطبيقات نظام الدفع OmniCRM

نظرة عامة

يوفر نظام الدفع OmniCRM بنية تحتية شاملة لمعالجة المدفوعات **غير المرتبطة بالبائع**. يدعم اليوم كل من Stripe وPayPal، لكن الهيكلية المودولارية تسمح بالتكامل مع أي مزود دفع (Square، Adyen، Braintree، إلخ) دون تغيير كود التطبيق.

يغطي هذا المستند جميع واجهات برمجة التطبيقات الخاصة بالدفع وسير العمل المتاحة في النظام.

> 🔧 **تكامل Playbook Ansible**: لتنفيذ هذه واجهات برمجة التطبيقات للدفع في Playbooks، راجع **التحصيل والمدفوعات من Playbooks** التزويد

جدول المحتويات

1. الهيكلية المودولارية
2. المفاهيم الأساسية
3. المستندات المالية
4. واجهات برمجة التطبيقات لطرق الدفع
5. واجهات برمجة التطبيقات لتدفقات الدفع
6. واجهات برمجة التطبيقات للمحفظة
7. ملخص مرجع واجهة برمجة التطبيقات
8. حالات الاستخدام الشائعة

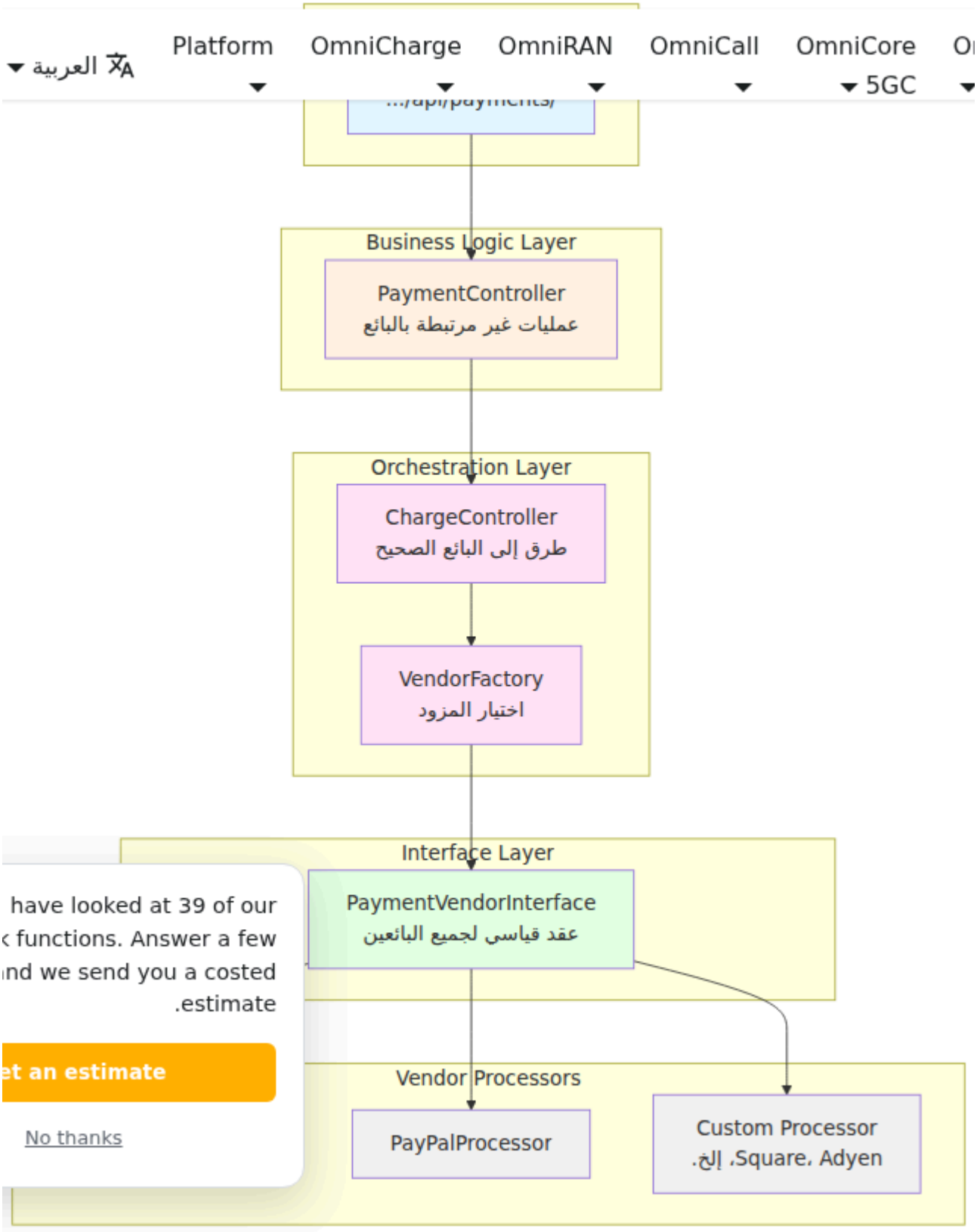
الهيكلية المودولارية

لماذا غير مرتبطة بالبائع؟

يستخدم النظام **طبقات التجريد** لفصل منطق الأعمال عن تفاصيل مزود الدفع. وهذا يعني:

- إضافة مزودي دفع جدد دون لمس كود التطبيق
- تبديل البائعين دون الحاجة إلى ترحيل قاعدة البيانات
- دعم عدة بائعين في نفس الوقت
- واجهة برمجة تطبيقات متسقة بغض النظر عن مزود الخلفية

طبقات الهيكلية



إضافة مزود دفع جديد

لإضافة مزود جديد (مثل Square، Adyen، Braintree)، اتصل بفريق OmniCRM الفني الخاص بك.

للحصول على تفاصيل التكوين حول المزودين الحاليين (Stripe، PayPal)، راجع **تكوين المزود**.

نظرة عامة على العملية:



الخطوات:

1. **تنفيذ فئة المعالج** مع `PaymentVendorInterface` القياسي (تفويض، التقاط، شحن، استرداد، تحرير)
2. **تسجيل المعالج** في VendorFactory مع اسم المزود
3. **اختبار التكامل** مع بيئة الاختبار الخاصة بالمزود
4. **نشر** - جميع واجهات برمجة التطبيقات الحالية تدعم تلقائيًا المزود الجديد

النتيجة: بمجرد النشر، يعمل المزود الجديد بسلاسة:

```
# إضافة طريقة دفع Square
curl -X POST /api/payments/methods \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "vendor": "square",
    "payment_token": "sq_xxxxx"
  }'
```

تدعم جميع واجهات برمجة التطبيقات للدفع المزود الجديد تلقائيًا دون الحاجة إلى تغييرات في كود التطبيق.

وقت التكامل: عادةً ما يستغرق من 1 إلى 2 يومًا لمعالج مزود جديد.

تكوين المزود

تمكين مزودي الدفع

تتحكم واجهة المستخدم في مزودي الدفع المتاحين للعملاء. يتم تعيين ذلك عبر المتغير `VITE_PAYMENT_VENDORS` في ملف `.env` الخاص بواجهة المستخدم:

```
# تمكين Stripe (افتراضي) فقط
VITE_PAYMENT_VENDORS="stripe"

# تمكين كل من Stripe وPayPal
VITE_PAYMENT_VENDORS="stripe,paypal"

# تمكين PayPal فقط
VITE_PAYMENT_VENDORS="paypal"
```

مهم: `VITE_PAYMENT_VENDORS` هو متغير **وقت البناء**. يتطلب تغييره إعادة بناء وإعادة تشغيل واجهة المستخدم.

تكوين Stripe

Stripe هو المزود الرئيسي للدفع، حيث يوفر معالجة بطاقات مع دعم 3D Secure.

1. الحصول على مفاتيح واجهة برمجة تطبيقات Stripe

- قم بالتسجيل في https://stripe.com
- انتقل إلى المطورين → مفاتيح واجهة برمجة التطبيقات
- انسخ **المفتاح القابل للنشر** الخاص بك (pk_live_... أو pk_test_...)
- انسخ **المفتاح السري** الخاص بك (sk_live_... أو sk_test_...)

مهم:

- للتطوير (pk_test_/sk_test_) استخدم **المفاتيح التجريبية**
- للإ��تاج فقط (pk_live_/sk_live_) استخدم **المفاتيح الحية**
- **لا** تقم بتسجيل مفاتيح واجهة برمجة التطبيقات في التحكم في الإصدارات

2. تكوين الخلفية

أضف إلى `crm_config.yaml` (مستوى الجذر، وليس متداخلاً تحت `payment_vendors`):

```
stripe:
  secret_key: 'sk_live_YOUR_SECRET_KEY_HERE'
  publishable_key: 'pk_live_YOUR_PUBLISHABLE_KEY_HERE'
  currency: 'aud'
  statement_descriptor_suffix: 'YOUR_COMPANY'
```

عند بدء التشغيل. `crm_config.yaml` تقرأ واجهة برمجة التطبيقات بيانات اعتماد Stripe من يتطلب تغيير هذه القيم إعادة تشغيل واجهة برمجة التطبيقات.

3. تكوين الواجهة الأمامية

أضف إلى ملف `.env` الخاص بواجهة المستخدم:

```
VITE_STRIPE_PUBLISHABLE_KEY=pk_live_YOUR_PUBLISHABLE_KEY_HERE
```

ملاحظة أمان: يجب أن يذهب المفتاح القابل للنشر فقط في الواجهة الأمامية. يجب **ألا** يتم الكشف عن المفتاح السري في المتصفح.

تكوين PayPal

يوفر PayPal تخزين بطاقات المدفوعات ومدفوعات حساب PayPal.

1. الحصول على بيانات اعتماد واجهة برمجة تطبيقات PayPal

- أنشئ تطبيق واجهة برمجة تطبيقات REST في https://developer.paypal.com/dashboard/
- انسخ **معرف العميل** و**المفتاح السري**
- استخدم بيانات اعتماد **Sandbox** للاختبار، استخدم بيانات اعتماد **Live** ؛ للإنتاج

2. تكوين الخلفية

أضف إلى `crm_config.yaml` (مستوى الجذر):

```yaml
paypal:
  client_id: 'YOUR_PAYPAL_CLIENT_ID'
  client_secret: 'YOUR_PAYPAL_CLIENT_SECRET'
  mode: 'sandbox'  # 'sandbox' للاختبار، 'live' للإنتاج
```

عند بدء التشغيل. `crm_config.yaml` تقرأ واجهة برمجة التطبيقات بيانات اعتماد PayPal من يتطلب تغيير هذه القيم إعادة تشغيل واجهة برمجة التطبيقات.

3. تكوين الواجهة الأمامية

أضف إلى ملف `.env` الخاص بواجهة المستخدم:

```
VITE_PAYMENT_VENDORS="stripe,paypal"
VITE_PAYPAL_CLIENT_ID=YOUR_PAYPAL_CLIENT_ID
```

يجب أن يتطابق `VITE_PAYPAL_CLIENT_ID` مع قيمة `paypal.client_id` في `crm_config.yaml`.

Sandbox مقابل Live

	Sandbox	Live
واجهة برمجة تطبيقات Stripe	`sk_test_` / `pk_test_`	`sk_live_` / `pk_live_`
واجهة برمجة تطبيقات PayPal	`api-m.sandbox.paypal.com`	`api-m.paypal.com`
وضع PayPal	`sandbox`	`live`
المال	مزيف	حقيقي

عند الانتقال من sandbox إلى live، **كلا** قم بتحديث `crm_config.yaml` (بيانات الاعتماد ثم أعد بناء واجهة المستخدم وأعد ،(المفتاح القابل للنشر / معرف العميل) وملف `.env` (والوضع تشغيل واجهة برمجة التطبيقات.

الامتثال لمعايير PCI

كيف يحافظ OmniCRM على الامتثال لمعايير PCI:

- يتم إدخال بيانات البطاقة مباشرة في iframes المستضافة من قبل البائع (Stripe Elements، حقول بطاقة PayPal)
- OmniCRM أرقام البطاقة الكاملة **لا ترى أو تخزن**
- يتم تخزين طرق الدفع المرمزة فقط في قاعدة البيانات
- تقليل نطاق الامتثال لمعايير PCI لعملك

مقاييس معالجة الدفع

يوفر OmniCRM مقاييس شاملة لمراقبة عمليات معالج�� الدفع. للحصول على تفاصيل كاملة حول مقاييس دفع Stripe، بما في ذلك تتبع استدعاءات واجهة برمجة التطبيقات، وأحجام الدفع، ومعدلات الفشل، وأوقات الاستجابة، راجع المراقبة والمقاييس.

مخطط قاعدة البيانات غير مرتبط بالبائع

يدعم مخطط قاعدة البيانات أي مزود دفع دون الحاجة إلى ترحيلات:



مثال Square - حفظ بطاقة:

```
{
  "vendor": "square",
  "vendor_payment_method_id": "sq_card_abc123",
  "payment_type": "card"
}
```

لا تغييرات في الكود. لا ترحيلات. يعمل فقط.

المفاهيم الأساسية

نماذج البيانات

PaymentMethod

تخزين غير مرتبط بالبائع لطرق دفع العملاء.

```
{
  "payment_method_id": 789,
  "customer_id": 123,
  "vendor": "stripe",                    // 'stripe'، 'paypal'، أو أي بائع مضاف
  "vendor_payment_method_id": "pm_xxx", // معرف داخلي للبائع
  "payment_type": "card",                // 'card'، 'paypal'، 'ach'، إلخ.
  "is_default": true,
  "card_brand": "visa",
  "card_last4": "4242",
  "card_exp_month": 12,
  "card_exp_year": 2025,
  "card_nickname": "My Visa Card",
  "status": "active"
}
```

PaymentAuthorization

سجلات تفويض الالتزام ذو المرحلتين (حجز الأموال).

```
{
  "authorization_id": 301,
  "customer_id": 123,
  "payment_method_id": 789,
  "vendor": "stripe",                    // أي بائع قام بالتفويض
  "vendor_authorization_id": "auth_xxx", // معرف تفويض البائع
  "amount": 200.00,
  "currency": "USD",
  "status": "authorized",                // 'authorized'، 'captured'، 'released'
  "authorized_at": "2025-12-27T10:00:00Z",
  "expires_at": "2026-01-03T10:00:00Z",
  "meta": {}
}
```

PaymentCapture

المدفوعات الملتقطة/المكتملة.

```
{
  "capture_id": 103,
  "authorization_id": 301,
  "customer_id": 123,
  "payment_method_id": 789,
  "vendor": "stripe",
  "vendor_transaction_id": "ch_xxx",     // معرف المعاملة للبائع
  "amount": 200.00,
  "currency": "USD",
  "status": "succeeded",                 // 'succeeded'، 'failed'، 'refunded'
  "captured_at": "2025-12-27T10:30:00Z",
  "vendor_response": {},                 // استجابة كاملة من البائع
  "meta": {}
}
```

WalletAccount

محفظة العميل مع تتبع الرصيد (1 إلى 1 مع العميل).

```
{
  "wallet_account_id": 456,
  "customer_id": 123,
  "balance": 150.50,
  "currency": "USD",
  "auto_recharge_enabled": true,
  "auto_recharge_amount": 100.00,
  "auto_recharge_threshold": 10.00,
  "low_balance_warning_threshold": 10.00
}
```

WalletLedger

سجل تدقيق كامل لجميع معاملات المحفظة.

```
{
  "ledger_id": 501,
  "customer_id": 123,
  "wallet_account_id": 456,
  "transaction_type": "credit",          // 'credit', 'debit',
'refund', 'adjustment'
  "amount": 100.00,
  "balance_before": 150.50,
  "balance_after": 250.50,
  "currency": "USD",
  "description": "Card top-up",
  "reference_type": "payment_capture",   // يرتبط بالكائن ذي الصلة
  "reference_id": 103,
  "meta": {},
  "created_at": "2025-12-27T10:35:00Z"
}
```

المستندات المالية

لإنشاء قوالب الفواتير والتخصيص، راجع **فواتير العملاء**.

الفواتير

التعريف: الفاتورة هي مستند يحتوي على قائمة من **المعاملات المدينة** (الرسوم). عند استدعاء واجهة برمجة التطبيقات لإنشاء فاتورة، يجب تقديم مصفوفة من معرفات المعاملات المدينة. يتم ربط" تلك المعاملات بالفاتورة عن طريق تعيين حقل "`invoice_id` الخاص بها.

للحصول على تفاصيل إدارة المعاملات، راجع **المعاملات**.

الحقول الرئيسية:

```
{
  "invoice_id": 12345,
  "invoice_number": "INV-2025-000001",   // تم إنشاؤه تلقائيًا: INV-YYYY-NNNNNN
  "customer_id": 123,
  "title": "فاتورة خدمات شهرية",
  "paid": true,                          // حالة الدفع
  "void": false,                         // حالة الإلغاء
  "payment_reference": "ch_xxxxx",       // آخر/معرف الدفع الرئيسي
  "payment_type": "stripe_capture",      // آخر نوع دفع
  "payment_time": "2025-12-27T10:30:00",
  "start_date": "2025-12-01",
  "end_date": "2025-12-31",
  "due_date": "2026-01-15",
  "retail_cost": 500.00,                 // إجمالي مبلغ الفاتورة
  "wholesale_cost": 250.00
}
```

إنشاء الفاتورة: يتم إنشاء أرقام الفواتير تلقائيًا بتنسيق `INV-YYYY-NNNNNN`، بالتسلسل ضمن السنة التقويمية، وإعادة تعيينها في 1 يناير من كل عام.

المدفوعات على الفواتير

كيف تعمل المدفوعات: بمجرد إنشاء فاتورة من المعاملات المدينة، يمكن تطبيق **الدفع** عن طريق إنشاء **معاملة ائتمانية** (تكلفة تجزئة سلبية) مرتبطة بالفاتورة.

يجب أن:

- يتم إدراجها في الفاتورة مع وضع علامة واضحة كـ "مدفوعات"
- تظهر تاريخ الدفع ذي الصلة (والذي يمكن أن يختلف عن تاريخ إنشاء الفاتورة)

- تدعم عدة مدفوعات لكل فاتورة
- يتم صافيها ضد الديون لحساب رصيد الفاتورة

حالة الفاتورة:

- **مدفوعة**: تساوي المدفوعات الإجمالية (الائتمانات) أو تتجاوز الديون الإجمالية
- **مدفوعة جزئيًا**: تم تطبيق بعض المدفوعات ولكن لا يزال هناك رصيد
- **مدفوعة زائدة**: لا يتم التعامل معها حاليًا - تتطلب تقسيم الائتمان (ميزة مستقبلية)

المحاسبة ذات القيد المزدوج: ينفذ النظام محاسبة صحيحة حيث كل رسوم لها دفعة معاكسة:

```
// 1. معاملة مدينة (رسوم)
{
  "transaction_id": 7001,
  "invoice_id": 12345,
  "retail_cost": 100.00,                    // إيجابي = العميل مدين
  "title": "رسوم الخدمة"
}

// 2. معاملة ائتمانية (دفع)
{
  "transaction_id": 7002,
  "invoice_id": 12345,
  "retail_cost": -100.00,                   // سلبية = تم استلام الدفع
  "title": "(12345) دفع للفاتورة: رسوم الخدمة",
  "payment_type": "stripe_capture",
  "payment_reference": "ch_xxxxx"
}

// النتيجة الصافية: رصيد 0$ → الفاتورة محددة كمدفوعة
```

بيانات التعريف الإضافية للفواتير: بالإضافة إلى المعاملات الائتمانية، تخزن الفاتورة أيضًا لعمليات البحث (`payment_reference`، `payment_type`، `payment_time`) حقول ملخص السريعة. ومع ذلك:

- **الطريقة الأساسية**: المعاملات الائتمانية مرتبطة عبر `invoice_id` (مواصفات هايدن)
- **بيانات التعريف الثانوية**: تخزن حقول الفاتورة ملخصًا لآخر/معرف الدفع الرئيسي
- **لعدة مدفوعات**: استعلام عن المعاملات الائتمانية للحصول على التاريخ الكامل

البيانات المالية

التعريف: تظهر البيانات المالية جميع **الديون والائتمانات** من معاملات العميل خلال فترة محددة. هذا هو نوع المستند الوحيد الذي يتم فيه عرض كل من الديون والائتمانات كعناصر قائمة معًا، مثل بيان البنك.

ملاحظات الائتمان

الغرض: يتم تطبيقها فقط على الفواتير التي تم تطبيق المدفوعات عليها بالفعل. يجب أن يتم ذلك كجزء من عملية إلغاء الفاتورة.

العملية:

1. يتم إلغاء الفاتورة (مع المعاملات المدينة المرتبطة بها)
2. يتم ربط أي مدفوعات تم تطبيقها بالفعل بملاحظة الائتمان التي تم إنشاؤها
3. يذهب رصيد العميل إلى الائتمان بمقدار المبلغ المعادل
4. يمكن بعد ذلك تطبيق رصيد ملاحظة الائتمان على فاتورة أخرى كدفع، أو استرداده للعميل

منطق الاسترداد: إذا تم اختيار الاسترداد، يتم استدعاؤه بناءً على المعاملات الائتمانية المرتبطة (على سبيل المثال، استرداد إذا كان نوع المعاملة الائتمانية هو Stripe وكيف تم دفعها في الأصل Stripe).

واجهات برمجة التطبيقات لطرق الدفع

تستخدم جميع نقاط النهاية عنوان URL الأساسي: `/api/payments/`

لإدارة طرق الدفع بالتفصيل والتعامل مع البطاقات، راجع **طرق الدفع**.

إضافة طريقة دفع

احفظ طريقة دفع جديدة لعميل.

نقطة النهاية: `POST /api/payments/methods`

الطلب:

```
curl -X POST https://your-domain.com/api/payments/methods \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "vendor": "stripe",
    "payment_token": "pm_xxxxx",
    "is_default": false,
    "card_nickname": "My Visa Card"
  }'
```

جسم الطلب:

```
{
  "customer_id": 123,
  "vendor": "stripe",                    // 'stripe'، 'paypal'، أو أي بائع مضاف
  "payment_token": "pm_xxxxx",           // رمز لمرة واحدة من SDK الواجهة الأمامية
  "is_default": false,                   // تعيين كطريقة دفع افتراضية؟
  "card_nickname": "My Visa Card"        // اسم ودي اختياري
}
```

الاستجابة (201 تم الإنشاء):

```
{
  "success": true,
  "message": "تمت إضافة طريقة الدفع بنجاح",
  "data": {
    "payment_method_id": 789,
    "customer_id": 123,
    "vendor": "stripe",
    "payment_type": "card",
    "card_brand": "visa",
    "card_last4": "4242",
    "card_exp_month": 12,
    "card_exp_year": 2025,
    "card_nickname": "My Visa Card",
    "is_default": false,
    "status": "active"
  }
}
```


الحصول على طرق الدفع

استرجع جميع طرق الدفع لعميل.

نقطة النهاية: `GET /api/payments/methods?customer_id={id}`

الطلب:

```
curl -X GET "https://your-domain.com/api/payments/methods?
customer_id=123" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

الاستجابة (200 حسنًا):

```
{
  "success": true,
  "data": [
    {
      "payment_method_id": 789,
      "customer_id": 123,
      "vendor": "stripe",
      "payment_type": "card",
      "card_brand": "visa",
      "card_last4": "4242",
      "is_default": true
    },
    {
      "payment_method_id": 790,
      "customer_id": 123,
      "vendor": "paypal",
      "payment_type": "paypal",
      "paypal_email": "user@example.com",
      "is_default": false
    }
  ]
}
```


الحصول على طريقة الدفع الافتراضية

احصل على طريقة الدفع الافتراضية لعميل.

نقطة النهاية: `GET /api/payments/methods/default?customer_id={id}`

الطلب:

```
curl -X GET "https://your-domain.com/api/payments/methods/default?
customer_id=123" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

الاستجابة (200 حسنًا):

```
{
  "success": true,
  "data": {
    "payment_method_id": 789,
    "vendor": "stripe",
    "payment_type": "card",
    "card_brand": "visa",
    "card_last4": "4242",
    "is_default": true
  }
}
```

تعيين طريقة الدفع الافتراضية

قم بتعيين طريقة دفع كافتراضية.

نقطة النهاية: `PUT /api/payments/methods/set-default`

الطلب:

```
curl -X PUT https://your-domain.com/api/payments/methods/set-
default \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "payment_method_id": 790
  }'
```

الاستجابة (200 حسنًا):

```
{
  "success": true,
  "message": "تم تحديث طريقة الدفع الافتراضية",
  "data": {
    "payment_method_id": 790,
    "is_default": true
  }
}
```

حذف طريقة الدفع

احذف طريقة دفع محفوظة.

نقطة النهاية: `DELETE /api/payments/methods/{payment_method_id}?customer_id={id}`

الطلب:

```
curl -X DELETE "https://your-domain.com/api/payments/methods/789?customer_id=123" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

الاستجابة (200 حسنًا):

```
{
  "success": true,
  "message": "تم حذف طريقة الدفع بنجاح"
}
```

واجهات برمجة التطبيقات لتدفقات الدفع

يدعم النظام عدة تدفقات دفع اعتمادًا على حالة الاستخدام الخاصة بك.

1. الدفع المباشر (رسوم بسيطة)

حالة الاستخدام: دفع بسيط في خطوة واحدة دون توفير الخدمة.

نقطة النهاية: `POST /api/payments/charge`

الطلب:

```
curl -X POST https://your-domain.com/api/payments/charge \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 50.00,
    "currency": "USD",
    "payment_method_id": 789,
    "metadata": {
      "order_id": "12345"
    }
  }'
```

جسم الطلب:

```
{
  "customer_id": 123,
  "amount": 50.00,
  "currency": "USD",                    // الافتراضي: USD
  "payment_method_id": 789,             // اختياري - يستخدم الافتراضي إذا تم حذفه
  "vendor": "stripe",                   // مطلوب إذا كنت تستخدم payment_token
  "payment_token": "pm_xxxxx",          // اختياري - رمز لمرة واحدة
  "save_method": false,                 // حفظ طريقة الدفع للاستخدام المستقبلي؟
  "metadata": {
    "order_id": "12345"
  }
}
```

الاستجابة (200 حسنًا):

```
{
  "success": true,
  "message": "تم الدفع بنجاح",
  "data": {
    "transaction_id": "ch_xxxxx",
    "capture_id": 101,
    "amount": 50.00,
    "currency": "USD",
    "status": "succeeded"
  }
}
```

2. دفع الفاتورة (توجيه المحفظة أولاً)

حالة الاستخدام: دفع فاتورة باستخدام رصيد المحفظة أولاً، ثم بطاقة للمتبقي.

نقطة النهاية: `POST /api/payments/invoice`

التدفق:

1. تحقق من رصيد المحفظة
2. استخدم أموال المحفظة أولاً
3. شحن البطاقة للقصور (إذا كان هناك)
4. ائتمان المحفظة بمبلغ البطاقة
5. خصم المحفظة لمبلغ الفاتورة الكامل

الطلب:

```
curl -X POST https://your-domain.com/api/payments/invoice \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 200.00,
    "payment_method_id": 789,
    "metadata": {
      "invoice_id": 12345,
      "description": "دفع فاتورة"
    }
  }'
```

(200 حسنًا) **الاستجابة**:

```
{
  "success": true,
  "message": "تم معالجة دفع الفاتورة",
  "data": {
    "customer_id": 123,
    "service_amount": 200.00,
    "routing_mode": "hybrid",
    "initial_balance": 150.00,
    "wallet_portion_used": 150.00,        // غطت المحفظة هذا المقدار
    "card_portion_used": 50.00,           // تم شحن البطاقة للمتبقي
    "charged_amount": 50.00,
    "wallet_credited": 50.00,
    "wallet_debited": 200.00,
    "final_balance": 0.00,
    "payment_method_used": true
  }
}
```

3. حجز التفويض (حجز الأموال)

حالة الاستخدام: حجز الأموال للاحتفاظ بها لاحقًا (مثل حجوزات الفنادق، الإيجارات).

> **تنفيذ Playbook**: لأمثلة Playbook Ansible نمط لتدفقات الدفع ذات المرحلتين، راجع الالتزام ذو المرحلتين

الخطوة 1: إنشاء حجز التفويض

نقطة النهاية: `POST /api/payments/authorize/hold`

تدفق المحفظة أولاً:

1. تحقق من رصيد المحفظة
2. احسب القصور (المبلغ - wallet_balance)
3. تفويض البطاقة للقصور فقط
4. خصم المحفظة يحدث عند وقت الالتقاط، وليس هنا

الطلب:

```
curl -X POST https://your-domain.com/api/payments/authorize/hold \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 200.00,
    "payment_method_id": 789,
    "use_wallet": true,
    "metadata": {
      "reservation_id": "RES-001"
    }
  }'
```

جسم الطلب:

```
{
  "customer_id": 123,
  "amount": 200.00,
  "currency": "USD",
  "payment_method_id": 789,              // اختياري - يستخدم الافتراضي إذا تم حذفه
  "vendor": "stripe",                    // مطلوب إذا كنت تستخدم payment_token
  "payment_token": "pm_xxxxx",           // اختياري - رمز لمرة واحدة
  "save_method": false,
  "use_wallet": true,                    // تمكين توجيه المحفظة أولاً (الافتراضي: true)
  "metadata": {
    "reservation_id": "RES-001"
  }
}
```

الاستجابة (200 حسنًا):

```
{
  "success": true,
  "message": "تم تفويض الدفع (تم إنشاء الحجز)",
  "data": {
    "authorization_id": 301,
    "vendor_authorization_id": "auth_xxxxx",
    "amount": 200.00,
    "currency": "USD",
    "status": "authorized",
    "wallet_balance": 150.00,
    "wallet_to_use": 150.00,             // ستغطي المحفظة هذا المقدار
    "card_amount": 50.00,                // تم تفويض البطاقة لهذا المقدار
    "message": "تم تفويض البطاقة بمبلغ 50 دولار (تعبئة المحفظة).
سيتم خصم المحفظة بمبلغ 200 دولار عند الالتقاط."
  }
}
```

الخطوة 2: التقاط التفويض

نقطة النهاية: `POST /api/payments/capture/{authorization_id}`

تدفق الالتقاط أولاً:

1. التقاط البطاقة (إذا تم تفويضها) - تعبئة المحفظة
2. ائتمان المحفظة بمبلغ البطاقة الملتقطة
3. خصم المحفظة لمبلغ الخدمة الكامل
4. إنشاء الفواتير/المعاملات (إذا تم الطلب)

الطلب:

```
curl -X POST https://your-domain.com/api/payments/capture/301 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "amount": 200.00,
    "metadata": {
      "invoice": true,
      "title": "حجز الفندق",
      "description": "إقامة لمدة 3 ليالٍ",
      "wholesale_cost": 100.00,
      "contract_days": 3,
      "send_email": true
    }
  }'
```

جسم الطلب:

```
{
  "amount": 200.00,                          // اختياري - يلتقط المبلغ الكامل إذا تم حذفه
  "metadata": {
    "invoice": true,                         // إنشاء فاتورة ومعاملات؟
    "create_transaction": true,              // إنشاء سجل المعاملة؟
    "title": "حجز الفندق",
    "description": "إقامة لمدة 3 ليالٍ",
    "wholesale_cost": 100.00,
    "contract_days": 3,
    "send_email": true                       // إرسال بريد إلكتروني للفاتورة؟
  }
}
```

الاستجابة (200 حسنًا):

```
{
  "success": true,
  "message": "تم التقاط التفويض",
  "data": {
    "capture_id": 103,
    "transaction_id": "ch_xxxxx",
    "authorization_id": 301,
    "amount": 200.00,
    "currency": "USD",
    "status": "succeeded",
    "wallet_credit": {                    // تم تعبئة المحفظة بالبطاقة
      "ledger_id": 401,
      "amount": 50.00
    },
    "wallet_debit": {                     // تم شحن الخدمة من المحفظة
      "ledger_id": 402,
      "amount": 200.00
    },
    "transaction": {                      // تم إنشاؤه إذا كان
invoice=true
      "transaction_id": 7001
    },
    "invoice": {                          // تم إنشاؤه إذا كان
invoice=true
      "invoice_id": 12345,
      "invoice_number": "INV-2025-000001"
    }
  }
}
```

الخطوة 3: تحرير التفويض (إلغاء)

نقطة النهاية: `POST /api/payments/release/{authorization_id}`

حالة الاستخدام: إلغاء الحجز، فشل التوفير، أو تغيير رأي العميل.

الطلب:

```
curl -X POST https://your-domain.com/api/payments/release/301 \
  -H "Authorization: Bearer YOUR_API_KEY"
```

الاستجابة (200 حسنًا):

```
{
  "success": true,
  "message": "تم تحرير التفويض",
  "data": {
    "authorization_id": 301,
    "vendor_authorization_id": "auth_xxxxx",
    "status": "released",
    "released_at": "2025-12-27T10:45:00Z"
  }
}
```

ملاحظة: مع تدفق المحفظة أولاً، لا حاجة لاسترداد المحفظة حيث لم يتم خصم المحف��ة حتى وقت الالتقاط.

4. دفع التعبئة (مرحلتين مع توفير)

حالة الاستخدام: معالجة الدفع لتعبئة الخدمة التي تتطلب توفير (مثل تفعيل نقطة اتصال/دونجل). إذا فشل التوفير، يتم تحرير التفويض.

نقطة النهاية: `POST /api/payments/topup`

التدفق: تفويض → توفير الخدمة → التقاط

الطلب:

```
curl -X POST https://your-domain.com/api/payments/topup \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 30.00,
    "payment_method_id": 789,
    "service_uuid": "svc-uuid-123",
    "imsi": "123456789012345",
    "days": 30,
    "metadata": {
      "is_rental": false
    }
  }'
```

(طريقة الدفع غير محفوظة) **لإيجار/نقطة اتصال مجهولة**:

```
curl -X POST https://your-domain.com/api/payments/topup \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 1,
    "amount": 5.00,
    "vendor": "stripe",
    "payment_token": "pm_xxxxx",
    "service_uuid": "hotspot-uuid",
    "imsi": "123456789012345",
    "days": 1,
    "metadata": {
      "is_rental": true,
      "billing_email": "user@example.com"
    }
  }'
```

(200 حسنًا) **الاستجابة**:

```
{
  "success": true,
  "message": "تمت معالجة دفع التعبئة بنجاح",
  "data": {
    "transaction_id": "ch_xxxxx",
    "authorization_id": 302,
    "capture_id": 104,
    "amount": 30.00,
    "status": "succeeded",
    "provision_result": {
      "success": true,
      "topup_result": {...},
      "service_uuid": "svc-uuid-123",
      "imsi": "123456789012345",
      "days": 30
    },
    "payment_method_saved": false        // خطأ للإيجار/المجهول
  }
}
```

5. دفع الإيجار (طرف ثالث)

حالة الاستخدام: شحن بطاقة عميل واحد لدفع خدمة عميل آخر.

نقطة النهاية: `POST /api/payments/rental`

الطلب:

```
curl -X POST https://your-domain.com/api/payments/rental \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "beneficiary_customer_id": 456,
    "charge_customer_id": 123,
    "amount": 75.00,
    "payment_method_id": 789,
    "service_description": "دفع خدمة الإيجار",
    "metadata": {
      "rental_agreement_id": "RA-001"
    }
  }'
```

(بطاقة المستأجر غير محفوظة) **لإيجار مجهول**:

```
curl -X POST https://your-domain.com/api/payments/rental \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "beneficiary_customer_id": 456,
    "amount": 75.00,
    "vendor": "stripe",
    "payment_token": "pm_xxxxx",
    "service_description": "دفع إيجار مجهول",
    "metadata": {
      "billing_email": "renter@example.com"
    }
  }'
```

(200 حسنًا) **الاستجابة**:

```
{
  "success": true,
  "message": "تمت معالجة دفع الإيجار",
  "data": {
    "transaction_id": "ch_xxxxx",
    "amount": 75.00,
    "payment_method_saved": false        // مجهول: لا توجد طريقة محفوظة
  }
}
```

6. استرداد الدفع

نقطة النهاية: `POST /api/payments/refund`

الطلب:

```
curl -X POST https://your-domain.com/api/payments/refund \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "transaction_id": "ch_xxxxx",
    "vendor": "stripe",
    "amount": 50.00,
    "reason": "customer_request"
  }'
```

جسم الط��ب:

```
{
  "transaction_id": "ch_xxxxx",          // معرف المعاملة للبائع
  "vendor": "stripe",                    // 'stripe'، 'paypal'، إلخ.
  "amount": 50.00,                       // اختياري - استرداد كامل إذا تم حذفه
  "reason": "customer_request"           // سبب الاسترداد الاختياري
}
```

الاستجابة (200 حسنًا):

```
{
  "success": true,
  "message": "تمت معالجة الاسترداد",
  "data": {
    "refund_id": "re_xxxxx",
    "amount": 50.00,
    "status": "succeeded"
  }
}
```

واجهات برمجة التطبيقات للمحفظة

تستخدم جميع نقاط النهاية الخاصة بالمحفظة عنوان URL الأساسي: `/api/wallet/`

الحصول على رصيد المحفظة

نقطة النهاية: `GET /api/wallet/balance?customer_id={id}`

الطلب:

```
curl -X GET "https://your-domain.com/api/wallet/balance?
customer_id=123" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

الاستجابة (200 حسنًا):

```
{
  "customer_id": 123,
  "balance": 150.50,
  "currency": "USD"
}
```

الحصول على معلومات المحفظة

احصل على معلومات كامل�� عن المحفظة والائتمان بما في ذلك إعدادات التعبئة التلقائية.

نقطة النهاية: `GET /api/wallet/info?customer_id={id}`

الطلب:

```
curl -X GET "https://your-domain.com/api/wallet/info?
customer_id=123" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

الاستجابة (200 حسنًا):

```
{
  "customer_id": 123,
  "wallet": {
    "wallet_account_id": 456,
    "balance": 150.50,
    "currency": "USD",
    "auto_recharge_enabled": true,
    "auto_recharge_amount": 100.00,
    "auto_recharge_threshold": 10.00,
    "low_balance_warning_threshold": 10.00
  }
}
```

تعبئة المحفظة

قم بتعبئة المحفظة عن طريق شحن طريقة الدفع.

نقطة النهاية: `POST /api/wallet/topup`

التدفق: شحن بطاقة ائتمان المحفظة → PayPal/

الطلب:

```
curl -X POST https://your-domain.com/api/wallet/topup \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 100.00,
    "payment_method_id": 789
  }'
```

جسم الطلب:

```
{
  "customer_id": 123,
  "amount": 100.00,
  "currency": "USD",                    // الافتراضي: USD
  "payment_method_id": 789              // اختياري - يستخدم الافتراضي إذا تم حذفه
}
```

الاستجابة (200 حسنًا):

```
{
  "success": true,
  "payment": {
    "transaction_id": "ch_xxxxx",
    "amount": 100.00
  },
  "wallet": {
    "ledger_id": 503,
    "balance_after": 250.50,
    "customer_transaction_id": 7001
  },
  "message": "تم تعبئة المحفظة بمبلغ 100.00 دولار أمريكي"
}
```

خط ائتمان الفواتير ينشئ (`customer_transaction` مع `payment_type=wallet_topup_card`، تكلفة تجزئة سلبية، `wallet_ledger_id` محدد). هذا يختلف عن توفير الهجين، حيث يتم ائتمان دفتر المحفظة بمبالغ البطاقة القصيرة كـ `card_capture_topup` دون خط تعبئة ائتمانية.

إلغاء تعبئة الأموال: `DELETE /crm/transaction/transaction_id/{transaction_id}` يخصم المحفظة ويسترد البطاقة. راجع إلغاء المعاملة أدناه.

الحصول على معاملات المحفظة

احصل على تاريخ معاملات المحفظة (دفتر الأستاذ).

نقطة النهاية: `GET /api/wallet/transactions?customer_id={id}&limit={n}&offset={n}&type={type}`

الطلب:

```
curl -X GET "https://your-domain.com/api/wallet/transactions?
customer_id=123&limit=50&offset=0&type=credit" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

معلمات الاستعلام:

- `customer_id` (مطلوب): معرف العميل
- `limit` (اختياري): (الافتراضي: 50) عدد السجلات
- `offset` (اختياري): (الافتراضي: 0) إزاحة الصفحات
- `type` (اختياري): تصفية حسب نوع المعاملة ('credit'، 'debit'، 'refund'، 'adjustment')

الاستجابة (200 حسنًا):

```
{
  "customer_id": 123,
  "count": 2,
  "transactions": [
    {
      "ledger_id": 501,
      "transaction_type": "credit",
      "amount": 100.00,
      "balance_before": 150.50,
      "balance_after": 250.50,
      "description": "تعبئة البطاقة",
      "reference_type": "payment_capture",
      "reference_id": 103,
      "created_at": "2025-12-27T10:35:00Z"
    },
    {
      "ledger_id": 502,
      "transaction_type": "debit",
      "amount": 50.00,
      "balance_before": 250.50,
      "balance_after": 200.50,
      "description": "رسوم الخدمة",
      "reference_type": "service_charge",
      "reference_id": 789,
      "created_at": "2025-12-27T11:00:00Z"
    }
  ]
}
```

إلغاء المعاملة

قم بإلغاء معاملة عميل غير مفوترة. يتطلب `delete_customer_transaction`.

نقطة النهاية: `DELETE /crm/transaction/transaction_id/{transaction_id}`

الطلب:

```
curl -X DELETE "https://your-
domain.com/api/transaction/transaction_id/7001" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

النجاح (200 حسنًا): رسوم يدوية بسيطة (لا رابط للمحفظة):

```
{
  "result": "نجاح"
}
```

النجاح (200 حسنًا): خط مرتبط بالمحفظة (إلغاء فقط):

```
{
  "result": "نجاح",
  "wallet_reversal": {
    "wallet_reversed": true,
    "wallet_amount": 50.00,
    "wallet_ledger_id": 503
  }
}
```

النجاح (200 حسنًا): تعبئة الأموال الصريحة (`payment_type=wallet_topup_card`):

```
{
  "result": "نجاح",
  "wallet_reversal": {
    "wallet_reversed": true,
    "wallet_amount": 100.00,
    "wallet_ledger_id": 503,
    "refund": {
      "capture_id": 88,
      "vendor": "stripe",
      "vendor_transaction_id": "ch_xxxxx",
      "refund_id": "re_xxxxx",
      "status": "succeeded",
      "amount": 100.00
    }
  }
}
```

سياسة الإلغاء / الاسترداد

مصدر المعاملة	`wallet_ledger_id`	المحفظة عند الإلغاء	استرداد البطاقة عند الإلغاء
رسوم/ائتمان يدوي	لا	لا	لا
تعبئة الأموال (`POST /wallet/topup`)	نعم	خصم المحفظة	**نعم** (`wallet_topup_card`)
نقد / ائتمان يدوي للمحفظة	نعم	خصم المحفظة	لا
فاتورة مدفوعة من المحفظة	نعم	ائتمان المحفظة	لا
خدمة / توفير (التقاط هجين)	عادةً **لا** على رسوم الشحن	لا*	لا

*توجه التدفقات الهجينة البطاقة → المحفظة → خصم الخدمة داخليًا (`card_capture_topup` + `service_charge`). صفوف دفتر الأستاذ هي رسوم الخدمة، وليس خط السطر الفاتورة المرئي. من المخطط أن يتم تقديم إلغاء أغنى لتلك الرسوم في فرع المستندات المالية تعبئة.

الأخطاء (500):

```
{
  "result": "فشل",
  "Reason": "رصيد المحفظة غير كاف. المتاح: 0.00، المطلوب: 20.00"
}
```

- **معاملة مفوترة:** `"لا يمكن حذف معاملة مرتبطة بفاتورة"`
- **رصيد المحفظة غير كاف:** أنفق العميل بالفعل الأموال المعبأة؛ يتم رفض الإلغاء (لا إلغاء جزئي)
- **فشل استرداد البطاقة:** يتم التراجع عن إلغاء المحفظة؛ تبقى المعاملة غير ملغاة

ملاحظة: يشير الإلغاء إلى `void=true` (الإحصائيات/الفوترة). ليس هو نفسه مستبعد من استرداد الفاتورة.

ملخص مرجع واجهة برمجة التطبيقات

نقاط نهاية طريقة الدفع

الطريقة	نقطة النهاية	الوصف
POST	`/api/payments/methods`	إضافة طريقة دفع
GET	`/api/payments/methods?customer_id={id}`	الحصول على جميع طرق الدفع
GET	`/api/payments/methods/default?customer_id={id}`	الحصول على طريقة الدفع الافتراضية
PUT	`/api/payments/methods/set-default`	تعيين طريقة الدفع الافتراضية
DELETE	`/api/payments/methods/{id}?customer_id={id}`	حذف طريقة الدفع

نقاط نهاية تدفق الدفع

الطريقة	نقطة النهاية	الوصف
POST	`/api/payments/charge`	دفع مباشر (خطوة واحدة)
POST	`/api/payments/invoice`	دفع فاتورة (توجيه المحفظة أولاً)
POST	`/api/payments/topup`	تعبئة مع توفير
POST	`/api/payments/authorize/hold`	إنشاء حجز التفويض
POST	`/api/payments/capture/{id}`	التقاط التفويض
POST	`/api/payments/release/{id}`	تحرير التفويض
POST	`/api/payments/rental`	دفع إيجار/طرف ثالث
POST	`/api/payments/refund`	استرداد الدفع

نقاط نهاية المحفظة

الطريقة	نقطة النهاية	الوصف
GET	`/api/wallet/balance?customer_id={id}`	الحصول على رصيد المحفظة
GET	`/api/wallet/info?customer_id={id}`	الحصول على معلومات المحفظة + الإعدادات
POST	`/api/wallet/topup`	تعبئة المحفظة عبر طريقة الدفع (ينشئ خط فواتير `wallet_topup_card`)
GET	`/api/wallet/transactions?customer_id={id}`	الحصول على تاريخ المعاملات

نقاط نهاية المعاملات

الطريقة	نقطة ��لنهاية	الوصف
PUT	`/api/transaction/`	إنشاء معاملة عميل
GET	`/api/transaction/`	قائمة المعاملات (مقيدة)
DELETE	`/api/transaction/transaction_id/{id}`	إلغاء المعاملة؛ إلغاء المحفظة؛ استرداد البطاقة فقط لتعبئة الأموال

نقاط نهاية خاصة بـ PayPal

الطريقة	نقطة النهاية	الوصف
POST	`/api/payments/paypal/vault/setup-token`	إنشاء رمز إعداد PayPal لـ Card Fields SDK
POST	`/api/payments/paypal/vault/finalize`	إنهاء خزانة PayPal وحفظ طريقة الدفع
POST	`/api/payments/paypal/vault/update-setup-token`	تحديث رمز الإعداد لمعالجة 3DS/SCA

حالات الاستخدام الشائعة

حالة الاستخدام 1: شحن العميل مقابل الخدمة (مدفوعة مسبقًا)

السيناريو: عميل يدفع مقابل 30 يومًا من الخدمة باستخدام توجيه المحفظة أولاً.

```
# دفع فاتورة (المحفظة أولاً، بطاقة للقصور)
curl -X POST https://your-domain.com/api/payments/invoice \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 99.99,
    "metadata": {
      "service_id": 456,
      "contract_days": 30,
      "description": "خدمة متميزة لمدة 30 يومًا"
    }
  }'
```

الاستجابة:

```
{
  "success": true,
  "message": "تمت معالجة دفع الفاتورة",
  "data": {
    "service_amount": 99.99,
    "wallet_portion_used": 50.00,        // كانت المحفظة تحتوي على 50 دولار
    "card_portion_used": 49.99,          // تم شحن البطاقة بمبلغ 49.99 دولار
    "final_balance": 0.00
  }
}
```

حالة الاستخدام 2: دفع ذو مرحلتين مع توفير الخدمة

السيناريو: توفير خدمة فقط بعد تفويض الدفع. إذا فشل التوفير، لا يحدث أي شحن.

```
# دفع تعبئة مع توفير تلقائي
curl -X POST https://your-domain.com/api/payments/topup \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 149.99,
    "payment_method_id": 789,
    "service_uuid": "fiber-svc-uuid",
    "imsi": "123456789012345",
    "days": 30,
    "metadata": {
      "service_type": "fiber_internet"
    }
  }'
```

التدفق:

1. تفويض 149.99 دولار على البطاقة
2. توفير خدمة الإنترنت الألياف
3. إذا نجح التوفير → التقاط الدفع
4. إذا فشل التوفير → تحرير التفويض (لا شحن)

حالة الاستخدام 3: دفع مجهول لنقطة اتصال

السيناريو: مستخدم مجهول يدفع مقابل الوصول إلى WiFi لا نقطة اتصال. لا سجل للعميل، طريقة دفع محفوظة.

```
# دفع نقطة اتصال مجهولة
curl -X POST https://your-domain.com/api/payments/topup \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 1,
    "amount": 5.00,
    "vendor": "stripe",
    "payment_token": "pm_xxxxx",
    "service_uuid": "hotspot-downtown",
    "imsi": "999999999999999",
    "days": 1,
    "metadata": {
      "is_rental": true,
      "billing_email": "user@example.com",
      "hotspot_location": "Cafe Downtown"
    }
  }'
```

النتيجة:

- ✅ تمت معالجة الدفع
- ✅ تم تفعيل نقطة الاتصال
- ❌ لم يتم حفظ طريقة الدفع
- ❌ لم يتم إنشاء سجل للعميل

حالة الاستخدام 4: حجز الفندق (حجز التفويض + التقاط)

السيناريو: حجز الأموال لحجز الفندق، التقاط عند تسجيل الوصول، تحرير إذا تم الإلغاء.

```
# الخطوة 1: تسجيل الوصول - حجز الأموال
curl -X POST https://your-domain.com/api/payments/authorize/hold \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 500.00,
    "use_wallet": true,
    "metadata": {
      "reservation_id": "RES-2025-001"
    }
  }'

# الاستجابة: {"authorization_id": 301, "status": "authorized"}

# الخطوة 2a: تسجيل الوصول للعميل - التقاط
curl -X POST https://your-domain.com/api/payments/capture/301 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "metadata": {
      "invoice": true,
      "title": "الإقامة في الفندق - 3 ليالٍ",
      "description": "الغرفة 302، 27-30 ديسمبر",
      "send_email": true
    }
  }'

# الخطوة 2b: إلغاء العميل - تحرير الحجز
curl -X POST https://your-domain.com/api/payments/release/301 \
  -H "Authorization: Bearer YOUR_API_KEY"
```

حالة الاستخدام 5: إضافة طريقة دفع (Stripe)

السيناريو: عميل يضيف بطاقة Visa جديدة إلى حسابه.

```
# الخطوة 1: الواجهة الأمامية تنشئ رمز Stripe عبر Stripe.js
# const {token} = await stripe.createToken(cardElement);

# الخطوة 2: الواجهة الخلفية تحفظ طريقة الدفع
curl -X POST https://your-domain.com/api/payments/methods \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "vendor": "stripe",
    "payment_token": "pm_xxxxx",
    "is_default": true,
    "card_nickname": "Work Visa"
  }'
```

حالة الاستخدام 6: تعبئة المحفظة

السيناريو: عميل يقوم بتعبئة محفظته بمبلغ 100 دولار باستخدام طريقة الدفع المحفوظة.

```
curl -X POST https://your-domain.com/api/wallet/topup \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 100.00
  }'
```

الاستجابة:

```
{
  "success": true,
  "payment": {
    "transaction_id": "ch_xxxxx",
    "amount": 100.00
  },
  "wallet": {
    "balance_after": 250.50
  },
  "message": "تم تعبئة المحفظة بمبلغ 100.00 دولار أمريكي"
}
```

معالجة الأخطاء

استجابات الأخطاء الشائعة

(400 طلب غير صحيح) **الأموال غير الكافية**:

```
{
  "error": "رصيد المحفظة غير كاف. المتاح: 50.00، المطلوب: 200.00"
}
```

(400 طلب غير صحيح) **فشل الدفع**:

```
{
  "error": "فشل الدفع. يرجى المحاولة مرة أخرى."
}
```

(400 طلب غير صحيح) **خطأ في التحقق**:

```
{
  "error": "customer_id و amount مطلوبان"
}
```

(404 غير موجود) **غير موجود**:

```
{
  "error": "التفويض 999 غير موجود"
}
```

(500 خطأ داخلي في الخادم) **خطأ في الخادم**:

```
{
  "error": "حدث خطأ أثناء معالجة الدفع الخاص بك"
}
```

الاستردادات وتوجيه المحفظة أولاً

خيارات الاسترداد

يدعم النظام **نوعين من الاستردادات** اعتمادًا على احتياجات عملك:

1. استرداد إلى مصدر الدفع (Stripe/PayPal)

حالة الاستخدام: يطلب العميل استردادًا كاملًا للخدمة الملغاة أو المنتج المعيب.

التدفق: تعود الأموال إلى طريقة الدفع الأصلية (البطاقة، حساب PayPal، إلخ.)

نقطة النهاية: `POST /api/payments/refund`

المثال:

```
curl -X POST https://your-domain.com/api/payments/refund \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "transaction_id": "ch_xxxxx",
    "vendor": "stripe",
    "amount": 100.00,
    "reason": "customer_request"
  }'
```

النتيجة:

- تتم معالجة استرداد Stripe/PayPal إلى طريقة الدفع الأصلية
- يرى العميل الائتمان على بطاقته/حساب PayPal خلال 5-10 أيام عمل
- لا تذهب الأموال إلى المحفظة
- يتم الحفاظ على سجل تدقيق كامل في جدول `PaymentCapture`

متى يجب استخدامه:

- ✅ ألغى العميل الطلب
- ✅ لم يتم تسليم الخدمة
- ✅ خطأ في الفوترة
- ✅ يطلب العميل صراحةً استردادًا إلى البطاقة

2. ائتمان إلى المحفظة

حالة الاستخدام: استرداد جزئي، ائتمان الخدمة، أو الاحتفاظ بالأموال للشراء في المستقبل.

التدفق: يتم ائتمان الأموال إلى رصيد المحفظة للعميل للاستخدام الفوري.

ملاحظة: يتم عادةً التعامل مع الائتمانات المحفظة داخليًا بواسطة النظام خلال سيناريوهات الخطأ. للحصول على الائتمانات اليدوية للمحفظة، اتصل بالدعم أو استخدم أدوات الإدارة.

النتيجة:

- الأموال متاحة على الفور في المحفظة
- لا فترة انتظار
- يمكن استخدامها للشراء في المستقبل

- يتم الحفاظ على سجل تدقيق كامل في جدول `WalletLedger`

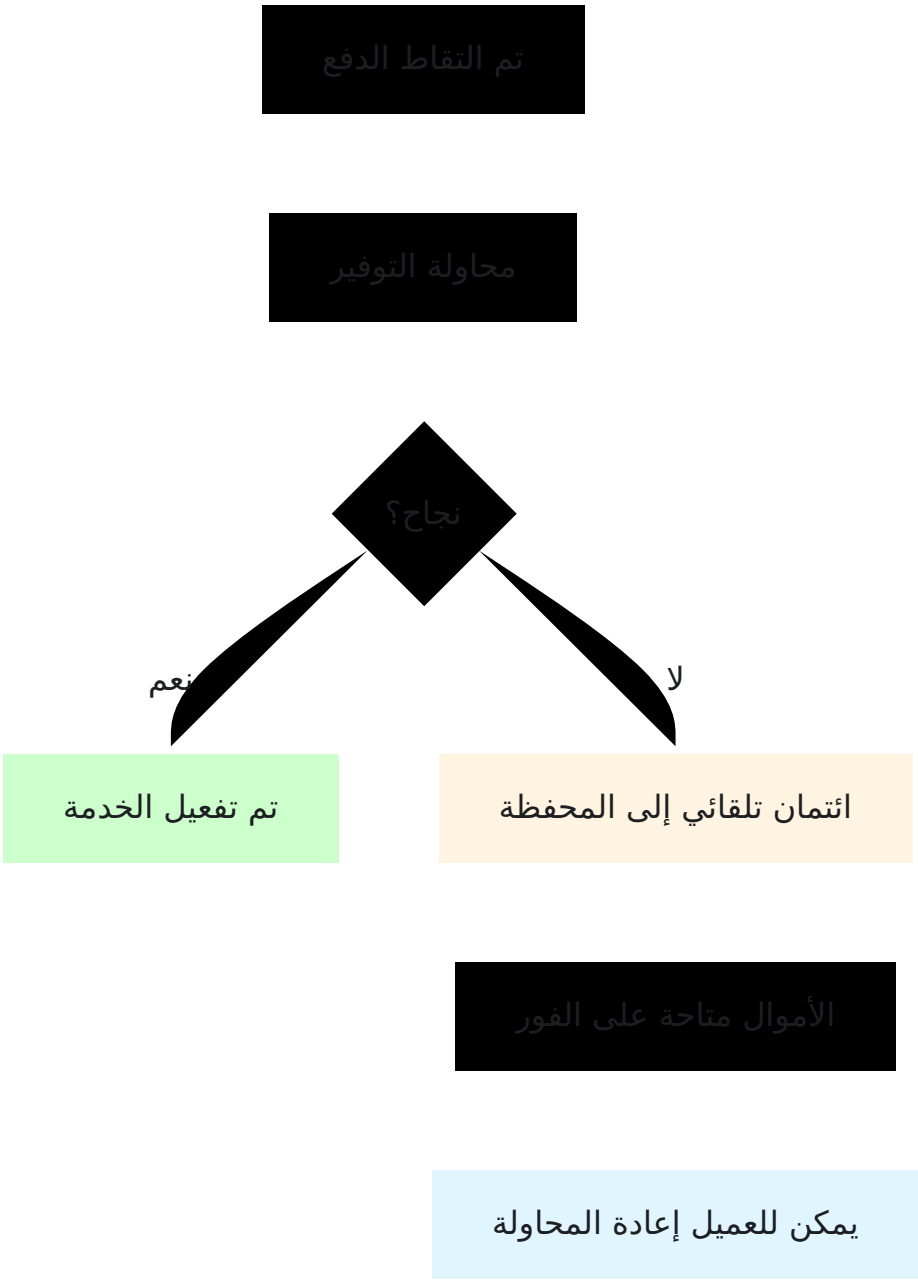
متى يجب استخدامه:

- ✅ ائتمانات الخدمة (مثل التعويض عن التوقف)
- ✅ استردادات جزئية حيث سيعيد العميل الشراء
- ✅ تعويض عن الخطأ (فشل التوفير)
- ❌ ائتمانات ترويجية

استراتيجية الاسترداد الهجينة

أفضل ممارسة: في سيناريوهات الخطأ أثناء تدفق الدفع، يقوم النظام تلقائيًا بائتمان المحفظة بدلاً من استرداد البطاقة.

استرداد الخطأ التلقائي:



سيناريو المثال:

1. يدفع العميل 100 دولار مقابل الخدمة
2. يتم شحن البطاقة بنجاح
3. يفشل التوفير
4. بدلاً من استرداد 100 دولار إلى البطاقة (5-10 أيام + رسوم الاسترداد):
    - ائتمان 100 دولار إلى المحفظة على الفور
    - يمكن للعميل إعادة المحاولة للشراء على الفور
    - لا رسوم استرداد
    - تجربة مستخدم أفضل

توجيه المحفظة أولاً: شحنات البطاقة المحسّنة

يقوم النظام **بشحن بطاقتك فقط للقصور**، وليس المبلغ الكامل، عندما يكون لديك رصيد في المحفظة.

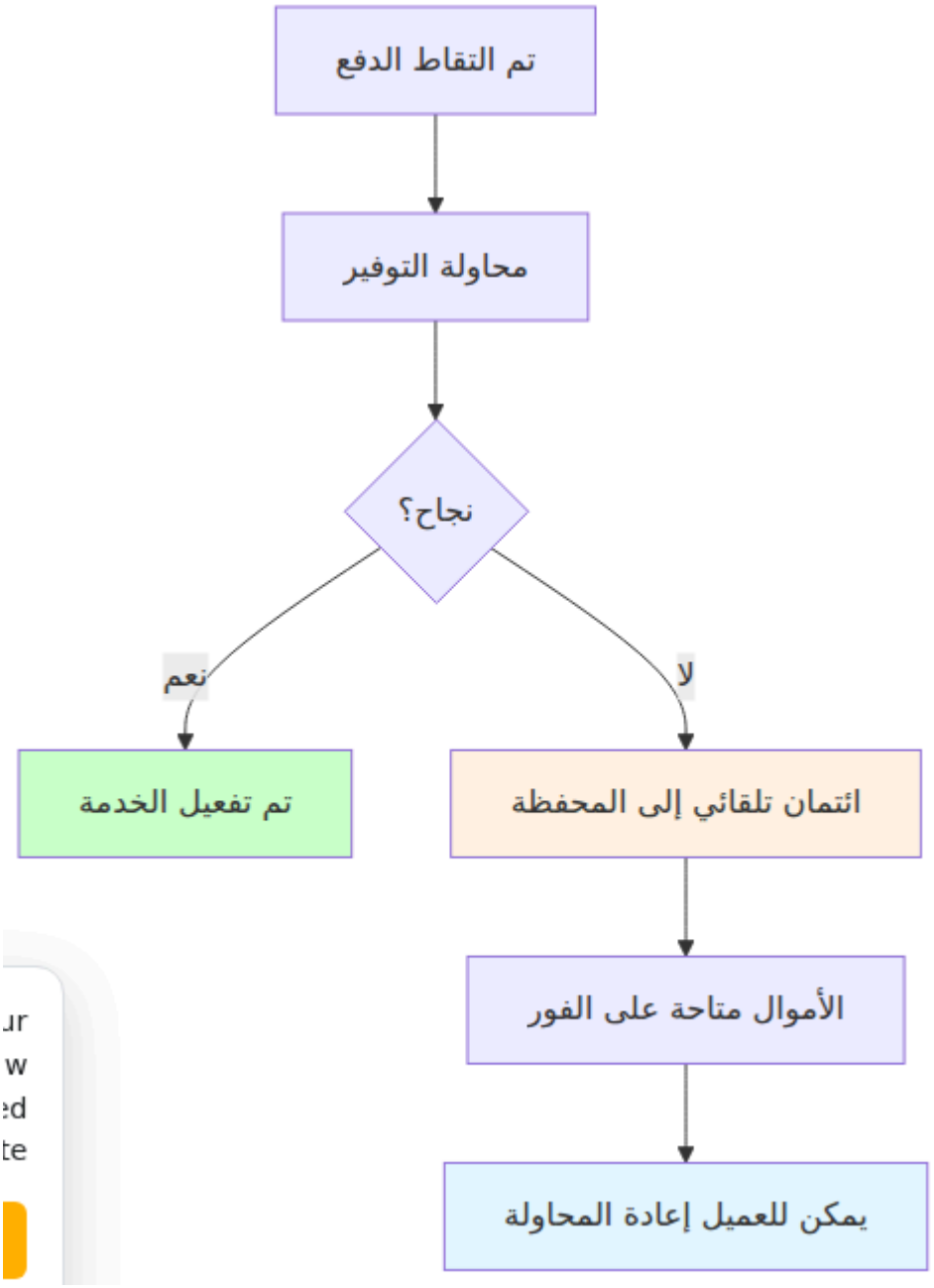
المثال 1: رصيد محفظة 1 دولار + شراء 10 دولارات

السيناريو: لديك 1 دولار في محفظتك وترغب في شراء إضافة بقيمة 10 دولارات.

تدفق الدفع التقليدي (ليس كيف يعمل هذا النظام):



تدفق OmniCRM المحفظة أولاً:



طلب واجهة برمجة التطبيقات:

```
curl -X POST https://your-domain.com/api/payments/invoice \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 10.00,
    "metadata": {
      "addon_id": 456
    }
  }'
```

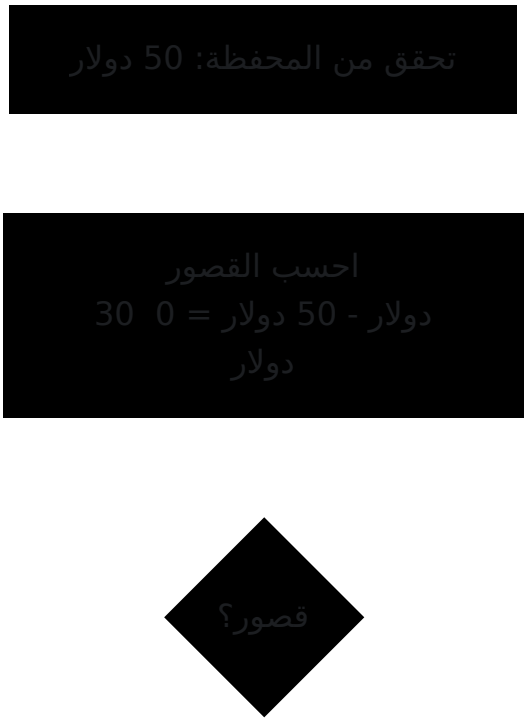
استجابة واجهة برمجة التطبيقات:

```
{
  "success": true,
  "message": "تمت معالجة دفع الفاتورة",
  "data": {
    "customer_id": 123,
    "service_amount": 10.00,
    "routing_mode": "hybrid",
    "initial_balance": 1.00,
    "wallet_portion_used": 1.00,     // استخدمت 1 دولار موجود
    "card_portion_used": 9.00,       // تم شحن 9 دولارات فقط
    "charged_amount": 9.00,          // ← شحن البطاقة
    "wallet_credited": 9.00,         // ← تعبئة المحفظة
    "wallet_debited": 10.00,         // ← رسوم الخدمة
    "final_balance": 0.00
  }
}
```

المثال 2: رصيد محفظة 50 دولار + شراء 30 دولار

السيناريو: لديك 50 دولار في المحفظة وشراء يكلف 30 دولار.

تدفق المحفظة أولاً:



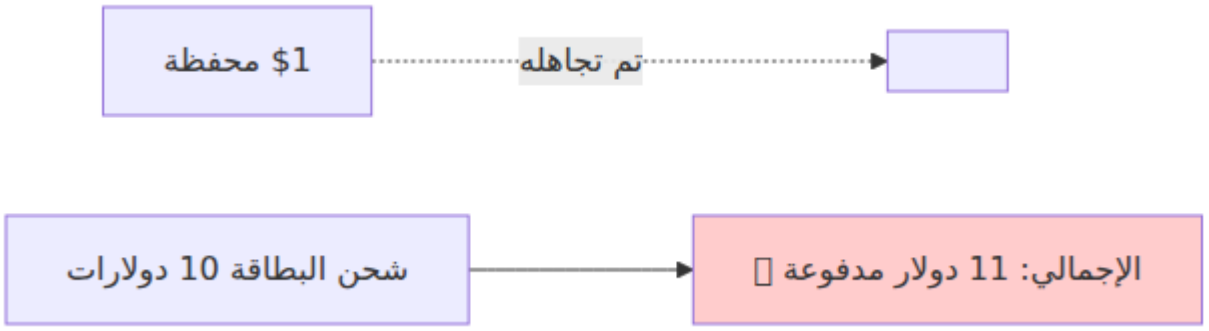
ستجابة واجهة برمجة التطبيقات��:

```
{
  "success": true,
  "data": {
    "service_amount": 30.00,
    "initial_balance": 50.00,
    "charged_amount": 0,              // ← لا شحن للبطاقة
    "wallet_debited": 30.00,
    "final_balance": 20.00,
    "payment_method_used": false      // ← لم يتم استخدام البطاقة
  }
}
```

المثال 3: حجز التفويض مع رصيد المحفظة

السيناريو: حجز 500 دولار للتفويض مع رصيد محفظة 150 دولار.

الخطوة 1: التفويض (`POST /api/payments/authorize/hold`):



استجابة واجهة برمجة التطبيقات:

```
{
  "authorization_id": 301,
  "amount": 500.00,
  "status": "authorized",
  "wallet_balance": 150.00,
  "wallet_to_use": 150.00,            // ستغطي المحفظة هذا المقدار
  "card_amount": 350.00,              // تم تفويض البطاقة لهذا المقدار
  "message": "تم تفويض البطاقة بمبلغ 350 دولار (تعبئة المحفظة).
سيتم خصم المحفظة بمبلغ 500 دولار عند الالتقاط."
}
```

الخطوة 2: الالتقاط (`POST /api/payments/capture/301`):

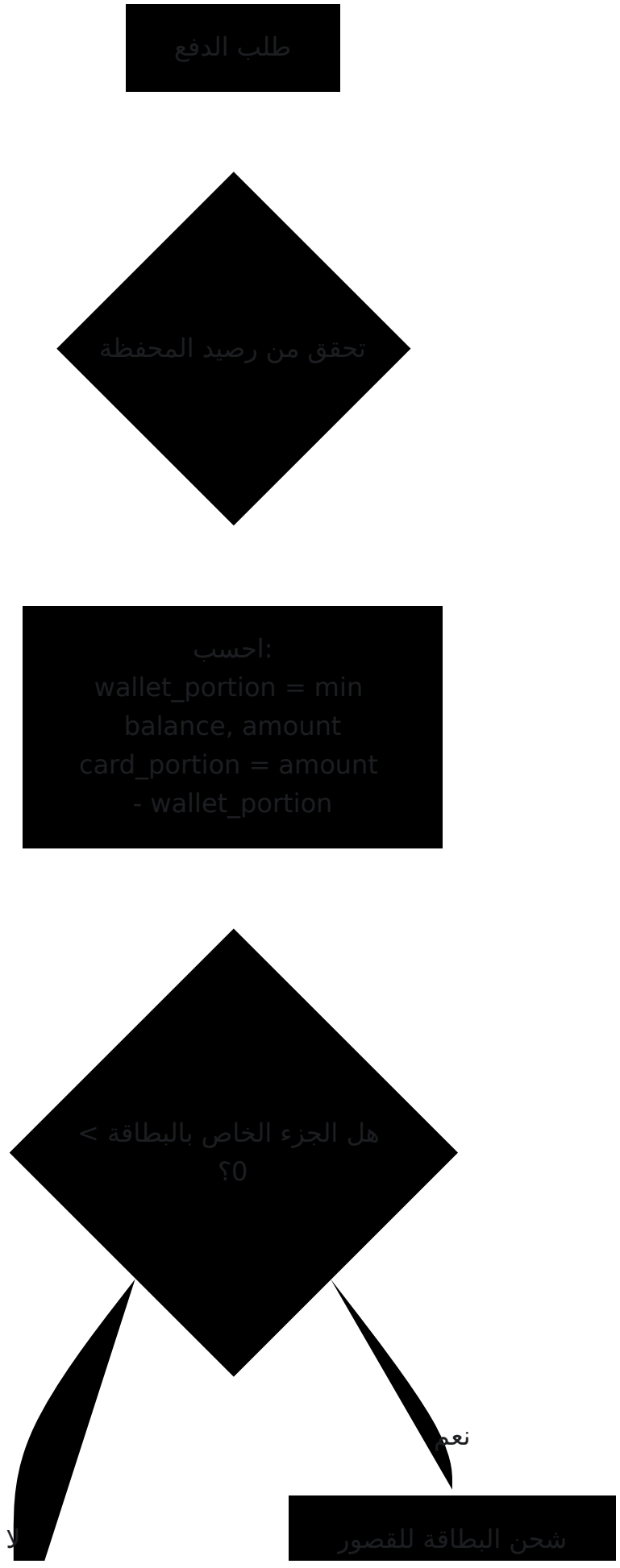


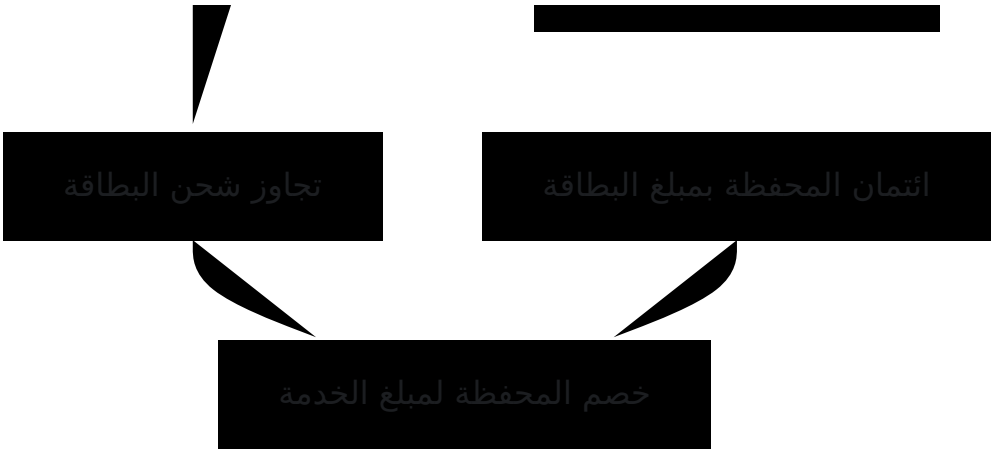
لماذا يهم هذا:

- يتم حجز 350 دولار فقط على بطاقتك، وليس 500 دولار
- يقلل من تأثير الائتمان المتاح للعميل
- مبالغ التفويض أكثر دقة
- تجربة أفضل للعميل

تفاصيل التنفيذ

منطق توجيه المحفظة أولاً:


طلب الدفع
تحقق من رصيد المحفظة
احسب:
wallet_portion = min
balance, amount
card_portion = amount
- wallet_portion
هل الجزء الخاص بالبطاقة >
0؟
نعم
شحن البطاقة للقصور



اكتمال

كيف يعمل:

1. يتحقق النظام من الرصيد الحالي للمحفظة
2. يحسب القصور: `wallet_balance - المبلغ`
3. إذا كان القصور > 0، يتم شحن البطاقة للقصور فقط
4. يتم ائتمان المحفظة بمبلغ الدفع من البطاقة
5. يتم خصم المحفظة لمبلغ الخدمة الكامل
6. النتيجة: يتم شحن العميل فقط لما لم تتمكن المحفظة من تغطيته

تجاوز وضع التوجيه

يمكنك تجاوز سلوك توجيه المحفظة إذا لزم الأمر:

وضع التجاوز: - شحن البطاقة دائمًا حتى لو كان لدى المحفظة أموال

```
curl -X POST https://your-domain.com/api/payments/charge \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 100.00,
    "metadata": {
      "routing_mode": "bypass"
    }
  }'
```

النتيجة: تم شحن البطاقة بمبلغ 100 دولار، وتم ائتمان المحفظة بمبلغ 100 دولار، وتم خصم المحفظة بمبلغ 100 دولار (صافي: 0 دولار)

حالة الاستخدام: يرغب العميل في استخدام البطاقة للحصول على مكافآت/نقاط حتى مع وجود رصيد في المحفظة.

أفضل الممارسات

1. **استخدم دائمًا توجيه المحفظة أولاً** لمدفوعات العملاء لتمكين الوظائف المدفوعة مسبقًا
2. **استخدم المدفوعات ذات المرحلتين** (نقطة النهاية للتعبئة) عند توفير الخدمات لتجنب الشحن إذا فشل التوفير
3. **قم بتعيين بيانات التعريف** لجميع المدفوعات للحفاظ على سجل التدقيق
4. **استخدم المدفوعات المجهولة** فقط للمستخدمين المجهولين حقًا (نقاط الاتصال، الإ

فواتير العملاء

> 🔗 **مرجع API**: لواجهات برمجة تطبيقات دفع الفواتير وسير العمل، راجع **دليل نظام الدفع API**

يتم تجميعها معًا لتشكيل فاتورة، والتي تُرسل إلى العميل للدفع المعاملات.

تحتوي الفواتير على تاريخ بدء وتاريخ انتهاء، وهو الفترة التي تغطيها الفاتورة، وتاريخ استحقاق، وهو التاريخ الذي تكون فيه الفاتورة مستحقة للدفع.

يمكن إنشاء الفواتير تلقائيًا بواسطة النظام، على سبيل المثال، عندما يتم فوتره خدمة، يتم إنشاء فاتورة للتكلفة التجزئة، أو يمكن إنشاؤها يدويًا، على سبيل المثال، إذا طلب العميل نسخة من فاتورة، أو إذا تم فوتره العميل لرسوم لمرة واحدة.

ويمكن تخصيصها لتشمل شعار Mailjet تكون فواتير العمل��ء مُصممة بالكامل باستخدام الشركة، العنوان، وتفاصيل الدفع، ويمكن إرسالها إلى العميل عبر البريد الإلكتروني، أو تنزيلها كملف PDF.

تخصيص قوالب الفواتير

يستخدم OmniCRM قوالب HTML مع Jinja2، لتوليد الفواتير. يمكنك تخصيص تصميم الفاتورة، العلامة التجارية، الألوان، والتخطيط بالكامل.

موقع قالب الفاتورة

تُخزن قوالب الفواتير في `OmniCRM-API/invoice_templates/`

القوالب الافتراضية:

- `sample_invoice_template.html` - قالب فاتورة عينة
- `cifi_invoice_template.html` - مثال لقالب بديل

التكوين:

يتم تحديد قالب الفاتورة النشط في `OmniCRM-API/crm_config.yaml`:

```
invoice:
  template_filename: 'sample_invoice_template.html'
```

المتغيرات المتاحة في القالب

تتمتع قوالب الفواتير بالوصول إلى المتغيرات التالية من Jinja2:

معلومات الفاتورة:

- `{{ invoice_number }}` - معرف الفاتورة الفريد (مثل `INV-2025-001234`)
- `{{ date }}` - تاريخ إصدار الفاتورة (تنسيق ISO: `2025-01-10T12:00:00`)
- `{{ due_date }}` - تاريخ استحقاق الدفع (مثل `10-02-2025`)
- `{{ start_date }}` - تاريخ بدء فترة الفوترة
- `{{ end_date }}` - تاريخ انتهاء فترة الفوترة
- `{{ total_amount }}` - إجمالي مبلغ الفاتورة قبل الضريبة (رقمي)
- `{{ total_tax }}` - إجمالي مبلغ الضريبة المحسوب من جميع المعاملات (رقمي)

معلومات العميل:

- `{{ client.name }}` - الاسم الكامل للعميل أو اسم الشركة
- `{{ client.address.address_line_1 }}` - السطر الأول من العنوان
- `{{ client.address.address_line_2 }}` - السطر الثاني من العنوان
- `{{ client.address.city }}` - المدينة
- `{{ client.address.state }}` - الولاية/المقاطعة
- `{{ client.address.zip_code }}` - الرمز البريدي
- `{{ client.address.country }}` - البلد

عناصر خط المعاملات:

تكرار عبر المعاملات باستخدام:

```
{% for sub_transaction in transactions %}
    <tr>
        <td>{{ sub_transaction.transaction_id }}</td>
        <td>{{ sub_transaction.created.split("T")[0] }}</td>
        <td>{{ sub_transaction.title }}</td>
        <td>{{ sub_transaction.description }}</td>
        <td>${{ "%.2f"|format(sub_transaction.retail_cost) }}</td>
    </tr>
{% endfor %}
```

حقول المعاملة:

- `sub_transaction.transaction_id` - معرف المعاملة
- `sub_transaction.created` - تاريخ/وقت المعاملة
- `sub_transaction.title` - عنوان المعاملة
- `sub_transaction.description` - وصف تفصيلي
- `sub_transaction.retail_cost` - مبلغ عنصر الخط
- `sub_transaction.tax_percentage` - نسبة الضريبة المطبقة (مثل 10 لـ 10%)
- `sub_transaction.tax_amount` - مبلغ الضريبة المحسوب بالدولار

عرض الضريبة في القوالب:

```
<td>
    {% if sub_transaction.tax_amount and
sub_transaction.tax_amount > 0 %}
        ${{ "%.2f"|format(sub_transaction.tax_amount) }} ({{
sub_transaction.tax_percentage }}%)
    {% else %}
        -
    {% endif %}
</td>
```

إنشاء قالب فاتورة مخصص

الخطوة 1: نسخ القالب الحالي

```
cd OmniCRM-API/invoice_templates/
cp sample_invoice_template.html your_company_invoice_template.html
```

الخطوة 2: تخصيص HTML/CSS

قم بتحرير `your_company_invoice_template.html` لتتناسب مع علامتك التجارية:

مجالات التخصيص الرئيسية:

1. **شعار الشركة والعلامة التجارية**

```
<!-- استبدل برابط شعارك -->
![شركتك](https://yourcompany.com/logo.png)

<!-- تحديث اسم الشركة -->
<h1>اسم شركتك</h1>
```

2. **نظام الألوان**

```
<style>
    /* اللون الأساسي للعلامة التجارية */
    .navbar {
        background: linear-gradient(to bottom right,
#your-color-1, #your-color-2);
    }

    /* رؤوس الجداول */
    .table thead th {
        background-color: #your-brand-color !important;
        color: white !important;
    }

    /* الأزرار والروابط */
    .btn-primary {
        background-color: #your-brand-color;
    }
</style>
```

3. **معلومات الشركة في التذييل**

```
<footer>
    <p>اسم شركتك</p>
    <p>123 شارع الأعمال، المدينة، البلد</p>
    <p>الهاتف: +1-555-123-4567 | البريد الإلكتروني:
billing@yourcompany.com</p>
    <p>رقم الضريبة: ABN/12345678900</p>
</footer>
```

4. **تعليمات الدفع**

```
<div class="payment-info">
    <h3>طرق الدفع</h3>
    <p><strong>عبر الإنترنت:</strong> الدفع على
https://yourcompany.com/pay</p>
    <p><strong>التحويل البنكي:</strong></p>
    <ul>
        <li>اسم الحساب: Your Company Ltd</li>
        <li>BSB: 123-456</li>
        <li>987654321 :رقم الحساب</li>
        <li>المرجع: {{ invoice_number }}</li>
    </ul>
</div>
```

5. **الشروط والأحكام**

```
<div class="terms">
    <h4>شروط الدفع</h4>
    <p>يجب الدفع خلال 30 يومًا من تاريخ الفاتورة.</p>
    <p>رسوم التأخير: 2% شهريًا على الأرصدة المتأخرة.</p>
    <p>للاستفسارات المتعلقة بالفوترة:
billing@yourcompany.com</p>
</div>
```

الخطوة 3: تحديث التكوين

قم بتحرير `OmniCRM-API/crm_config.yaml`:

```
invoice:
  template_filename: 'your_company_invoice_template.html'
```

الخطوة 4: إعادة تشغيل API

```
cd OmniCRM-API
sudo systemctl restart omnicrm-api
```

الخطوة 5: اختبار إنشاء الفاتورة

1. انتقل إلى عميل لديه معاملات

2. قم بإنشاء فاتورة اختبار
3. قم بتنزيل PDF للتحقق من التنسيق
4. أرسل الفاتورة إلى نفسك لاختبار تسليم البريد الإلكتروني

تخصيص متقدم

المحتوى الشرطي:

استخدم شروط Jinja2 لإظهار/إخفاء المحتوى:

```
{% if total_amount > 1000 %}
    <div class="high-value-notice">
        <p><strong>ملاحظة:</strong> رصيد كبير - خطة الدفع متاحة عند الطلب.</p>
    </div>
{% endif %}

{% if client.address.country == "Australia" %}
    <p>شامل ضريبة السلع والخدمات: ${{ "%.2f"|format(total_amount * 0.10) }}</p>
{% endif %}
```

دعم متعدد اللغات:

إنشاء قوالب خاصة باللغة:

```
invoice_template_en.html
invoice_template_es.html
invoice_template_fr.html
```

تكوين بناءً على تفضيل لغة العميل.

حسابات مخصصة:

```
<!-- عرض المجموع الفرعي وتفاصيل الضريبة -->
<tr>
    <td colspan="4" class="text-right"><strong>المجموع الفرعي:
</strong></td>
    <td>${{ "%.2f"|format(total_amount) }}</td>
</tr>
<tr>
    <td colspan="4" class="text-right"><strong>الضريبة:</strong>
</td>
    <td>${{ "%.2f"|format(total_tax) }}</td>
</tr>
<tr>
    <td colspan="4" class="text-right"><strong>الإجمالي:</strong>
</td>
    <td>${{ "%.2f"|format(total_amount + total_tax) }}</td>
</tr>
```

ملاحظة: يتم حساب متغير `total_tax` تلقائيًا عن طريق جمع `tax_amount` من جميع المعاملات في الفاتورة. يتم حساب ضريبة كل معاملة بناءً على حقل `tax_percentage` الخاص بها، والذي يكون افتراضيًا نسبة الضريبة الخاصة بالمنتج أو 0% إذا لم يتم تحديده.

رمز QR للدفع:

توليد رمز QR للدفع عبر الهاتف المحمول:

```
<div class="qr-payment">
    ![امسح للدفع](https://api.qrserver.com/v1/create-qr-code/?
size=150x150&data={{ payment_url }})
    <p>امسح باستخدام هاتفك للدفع على الفور</p>
</div>
```

أفضل ممارسات تنسيق PDF

يستخدم OmniCRM **WeasyPrint** لتحويل HTML إلى PDF. اتبع هذه الإرشادات:

CSS المدعوم:

- معظم خصائص CSS 2.1
- CSS3 المحدودة (flexbox، بعض التحولات)

- خطوط الويب عبر `@font-face`

غير مدعوم:

- JavaScript
- CSS Grid (استخدم الجداول بدلاً من ذلك)
- الرسوم المتحركة المعقدة
- بعض خصائص CSS الحديثة

حجم الصفحة والهوامش:

```
@page {
    size: A4;
    margin: 1cm;
}

body {
    font-family: Arial, sans-serif;
    font-size: 10pt;
}
```

تنسيق الطباعة المحدد:

```
@media print {
    .no-print {
        display: none;
    }

    .page-break {
        page-break-after: always;
    }
}
```

تخطيط الجدول:

```
.table {
    table-layout: fixed;
    width: 100%;
}

.table th, .table td {
    word-wrap: break-word;
    padding: 4px;
}
```

تضمين الخط:

للحصول على خطوط مخصصة، استخدم خطوط آمنة على الويب أو قم بتضمينها:

```
@font-face {
    font-family: 'YourFont';
    src: url('https://yourcompany.com/fonts/yourfont.woff2')
format('woff2');
}

body {
    font-family: 'YourFont', Arial, sans-serif;
}
```

اختبار قوالب الفواتير

قائمة التحقق للاختبار:

1. **الفحص البصري:**
    - يظهر الشعار بشكل صحيح
    - تتطابق الألوان مع إرشادات العلامة التجارية
    - النص قابل للقراءة (ليس صغيرًا جدًا)
    - تتماشى الجداول بشكل صحيح
    - جميع الأقسام موجودة
2. **دقة البيانات:**
    - تفاصيل العميل صحيحة
    - مجموعات المعاملات صحيحة

    - تواريخ مُنسقة بشكل صحيح
    - جميع المتغيرات تستبدل بشكل صحيح
3. **جودة PDF:**
    - حجم الملف معقول (<5MB)
    - الصور حادة وواضحة
    - لا يوجد نص مقطوع أو تجاوز
    - تظهر فواصل الصفحات في الأماكن المناسبة
4. **الفواتير متعددة الصفحات:**
    - تتكرر الرؤوس في كل صفحة
    - تظهر أرقام الصفحات
    - قوائم المعاملات الطويلة تتوزع بشكل صحيح
5. **تسليم البريد الإلكتروني:**
    - يتم إرفاق PDF بالبريد الإلكتروني
    - حجم الملف تحت حد Mailjet (15MB)
    - يظهر في Gmail وOutlook وApple Mail

أمر الاختبار (التوليد اليدوي):

يمكنك اختبار إنشاء الفاتورة عبر API:

```
curl -X GET "http://localhost:5000/crm/invoice/{invoice_id}/pdf" \
     -H "Authorization: Bearer YOUR_TOKEN" \
     --output test_invoice.pdf
```

مشكلات القالب الشائعة

المتغيرات لا تستبدل:

- **السبب:** خطأ مطبعي في اسم المتغير أو بيانات مفقودة
- **الإصلاح:** تحقق من التهجئة بدقة (حساسة لحالة الأحرف)، تحقق من وجود البيانات في قاعدة البيانات

تنسيق PDF معطل:

- **السبب:** خاصية CSS غير مدعومة
- **الإصلاح:** استخدم خصائص CSS 2.1، اختبر باستخدام WeasyPrint متوافق مع CSS

الصور لا تظهر:

- **السبب:** روابط نسبية أو موارد خارجية محجوبة
- **الإصلاح:** استخدم روابط HTTPS المطلقة، تأكد من أن الصور متاحة للجمهور

الجداول تتجاوز الصفحة:

- **السبب:** عرض الأعمدة الثابتة واسع جدًا
- **الإصلاح:** استخدم عرض النسب المئوية، `table-layout: fixed`

الخطوط لا تظهر:

- **السبب:** الخط غير مضمن أو غير متاح
- **الإصلاح:** استخدم خطوط آمنة على الويب (Arial، Times New Roman، إلخ) أو قم بتضمين الخطوط المخصصة بشكل صحيح

فشل إنشاء PDF:

- **السبب:** أخطاء في بناء الجملة HTML أو تعطل WeasyPrint
- **الإصلاح:** تحقق من صحة HTML، تحقق من سجلات WeasyPrint، قم بتبسيط التخطيطات المعقدة

تخزين PDF للفواتير

نظام تخزين OmniCRM الزائد، يتضمن PDF لتحسين الأداء وتقليل إنشاء PDF للفواتير. عند إنشاء فاتورة لأول مرة، يتم تخزينه في قاعدة البيانات للطلبات اللاحقة.

كيف يعمل تخزين PDF:

1. **الطلب الأول** - عند طلب فاتورة (تنزيل أو بريد إلكتروني)، يقوم النظام بإنشاء PDF من قالب الفاتورة
   - ترميز PDF كـ Base64
   - حساب تجزئة SHA256 لمحتوى PDF
   - تخزينه في جدول `Invoice_PDF_Cache` مع:
     - مرجع معرف الفاتورة
     - بيانات PDF (مشفرة بـ Base64)
     - اسم الملف

        - تجزئة المحتوى (للتحقق من النزاهة)
        - الطابع الزمني لإنشاء
2. **الطلبات اللاحقة** - عند طلب نفس الفاتورة مرة أخرى:
    - يتحقق النظام من invoice_id المخزن حسب PDF
    - إذا كان التخزين موجودًا وصالحًا، يتم إرجاع PDF المخزن على الفور
    - يتم تحديث الطابع الزمني `last_accessed` لتتبع استخدام التخزين
3. **إبطال التخزين** - يتم إبطال PDFs المخزنة عندما:
    - يتم تعديل الفاتورة (إضافة/إزالة معاملات، تغيير التفاصيل)
    - يتم تحديث قالب الفاتورة
    - يتم تشغيل مسح التخزين اليدوي

الفوائد:

- **الأداء** - تسليم PDF فوري للطلبات المتكررة (بدون تأخير في إعادة التوليد)
- **الاتساق** - نفس PDF لجميع تنزيلات الفاتورة (ما لم يتم تعديل الفاتورة)
- **حمولة الخادم** - تقليل استخدام وحدة المعالجة المركزية من إنشاء PDF
- **تجربة المستخدم** - يظهر مؤشر التحميل أثناء التوليد الأولي، الطلبات اللاحقة تكون فورية

إدارة التخزين:

يتم إدارة تخزين PDF للفواتير تلقائيًا بواسطة النظام. يمكن مسح إدخالات التخزين القديمة أو غير المستخدمة بشكل دوري بناءً على:

- العمر (مثل، إزالة إدخالات التخزين التي تزيد عن 90 يومًا)
- أنماط الوصول (إزالة الإدخالات التي لم يتم الوصول إليها في 30 يومًا)
- حدود التخزين (تنفيذ حدود حجم التخزين إذا لزم الأم��)

سلوك API:

عند تنزيل فاتورة عبر API أو واجهة المستخدم:

- الطلب الأول: يظهر مؤشر التحميل أثناء إنشاء PDF، ثم يتم تخزينه
- الطلبات اللاحقة: تنزيل فوري من التخزين
- نجاح/فشل التخزين شفاف للمستخدم

مهم: عند تحديث قالب الفاتورة، قم بمسح التخزين لضمان استخدام الفواتير الجديدة التصميم المحدث:

```
-- مسح جميع PDFs المخزنة للفواتير (في تشغيل MySQL)
DELETE FROM Invoice_PDF_Cache;
```

لإبطال التخزين تلقائيًا عند تغيير القالب `crm_config.yaml` أو تحديث.

الوصول إلى الفواتير

يمكن عرض الفواتير على مستوى النظام أو لكل عميل:

عرض لكل عميل:

1. انتقل إلى **العملاء** ← **[اختر عميل]**
2. انقر على علامة التبويب **الفوترة**
3. عرض قائمة الفواتير في البطاقة الثالثة

عرض على مستوى النظام:

1. انتقل إلى **الفوترة** ← **الفواتير** (من القائمة الرئيسية)
2. عرض جميع الفواتير عبر جميع العملاء

أدوات إحصائيات الفواتير

في أعلى صفحة الفواتير، تعرض أربع بطاقات إحصائية ملخصات مالية.

وصف الأدوات:

- **إجمالي الفواتير** - مجموع جميع تكاليف الفواتير (على مر الزمن) وعدد الفواتير المرسلة
- **الفواتير غير المدفوعة** - مجموع الفواتير التي لم يتم دفعها بعد وعدد الفواتير غير المدفوعة
- **الفواتير هذا الشهر** - مجموع الفواتير التي تم إنشاؤها في هذا الشهر الميلادي مع العدد
- **الفواتير الشهر الماضي** - مجموع الفواتير التي تم إنشاؤها في الشهر الميلادي الماضي مع العدد

تنسيق القيم:

- القيم التي تزيد عن 1,000 عرض كلاحقة "k" (1.5$ ،مثلk)
- القيم التي تزيد عن 1,000,000 عرض كلاحقة "M" (2.3$ ،مثلM)
- القيم التي تزيد عن 1,000,000,000 عرض كلاحقة "B" (1.1$ ،مثلB)

مؤشرات الاتجاه:

- تعرض الأدوات لـ "هذا الشهر" و "الشهر الماضي" نسبة التغيير
- سهم أخضر لأعلى: زيادة عن الفترة السابقة
- سهم أحمر لأسفل: انخفاض عن الفترة السابقة
- سهم رمادي لليمين: لا تغيير

قائمة الفواتير

تعرض جدول الفواتير جميع الفواتير مع الأعمدة التالية:

وصف الأعمدة:

- **ID** - معرف الفاتورة الفريد
- **العنوان** - عنوان/وصف الفاتورة
- **الفترة** - فترة الفوترة (تاريخ البدء - تاريخ الانتهاء) أو "N/A" للفواتير لمرة واحدة
- **تاريخ الاستحقاق** - تاريخ استحقاق الدفع
- **تاريخ الإنشاء** - تاريخ إنشاء الفاتورة
- **المبلغ** - إجمالي مبلغ الفاتورة (تكلفة التجزئة)
- **الحالة** - مدفوعة، غير مدفوعة، أو مستردة
- **الإجراءات** - الإجراءات المتاحة (تختلف حسب الحالة)

أيقونات الإجراءات:

- ⬇ **(تنزيل)** - تنزيل PDF الفاتورة
- 🗑 **(حذف)** - إلغاء الفاتورة (فقط إذا لم يتم دفعها)
- 💳 **(دفع)** - دفع الفاتورة عبر الإنترنت (فقط إذا كانت غير مدفوعة)
- ✉ **(بريد إلكتروني)** - إرسال بريد إلكتروني بالفاتورة إلى العميل

- **(استرداد)** - استرداد دفعة Stripe (فقط للفواتير المدفوعة عبر Stripe)

إنشاء فاتورة

انقر على **"+ إنشاء فاتورة أولية"** لإنشاء فاتورة جديدة.

وصف الحقول:

- **البحث عن العملاء** - اختر العميل (يظهر فقط في العرض على مستوى النظام، مُعبأ مسبقًا في عرض العميل)
- **العنوان** - عنوان/اسم الفاتورة (اختياري، الافتراضي هو "فاتورة لـ [الفترة]")
- **تاريخ البدء** - بداية فترة الفوترة (الافتراضي هو 14 يومًا مضت)
- **تاريخ الانتهاء** - نهاية فترة الفوترة (الافتراضي هو اليوم)
- **تاريخ الاستحقاق** - الموعد النهائي للدفع (الافتراضي هو اليوم)
- **معاينة المعاملة** - تظهر جميع المعاملات غير المفوترة في ��طاق التاريخ مع القدرة على تضمين/استبعاد معاملات معينة

اختيار المعاملات:

- ✓ **(علامة زائد خضراء)** - انقر لاستبعاد معاملة من الفاتورة
- × **(حمراء X علامة)** - انقر لتضمين معاملة تم استبعادها سابقًا
- **تحديد الكل** - تضمين جميع المعاملات المعروضة
- **مسح الكل** - استبعاد جميع المعاملات

- تظهر المعاملات المستبعدة بلون رمادي مع نص مخطوط
- يتم تحديث المجموعات في الوقت الحقيقي أثناء تحديد/إلغاء تحديد المعاملات

ما يحدث:

1. يقوم النظام بالبحث عن جميع المعاملات غير المفوترة للعميل ضمن نطاق الت��ريخ
2. يعرض معاينة المعاملة مع القدرة على تضمين/استبعاد معاملات فردية
3. يظهر حساب في الوقت الحقيقي للمجموع الفرعي، الضريبة، والإجمالي بناءً على المعاملات المحددة
4. تتم إضافة المعاملات المحددة (المتضمنة) فقط إلى الفاتورة
5. يتم إنشاء PDF الفاتورة وتخزينه
6. يتم وضع علامة على المعاملات المحددة على أنها مفوترة (تم تعبئته `invoice_id`)
7. تبقى المعاملات المستبعدة غير مفوترة ومتاحة للفواتير المستقبلية
8. تظهر الفاتورة في القائمة بحالة "غير مدفوعة"

حالات الاستخدام المثال:

الفوترة الشهرية: تعيين تاريخ البدء إلى الأول من الشهر، وتاريخ الانتهاء إلى آخر يوم من الشهر، تظهر المعاينة جميع المعاملات غير المفوترة من تلك الفترة. اختر الكل أو استبعد يدويًا معاملات معينة.

فاتورة خاصة بالخدمة: استخدم نفس نطاق التاريخ، ثم استبعد يدويًا المعاملات غير المرغوب فيها (مثل، استبعاد المعاملات غير المحمولة لإنشاء فاتورة خاصة بالهاتف المحمول فقط).

فاتورة لمرة واحدة: تعيين كل من تاريخ البدء والانتهاء إلى نفس اليوم، تظهر المعاينة فقط المعاملات من ذلك التاريخ. استبعد أي رسوم غير ذات صلة بهذه الفاتورة المحددة.

عرض تفاصيل الفاتورة

انقر على أي صف فاتورة في الجدول لعرض تفاصيل الفاتورة الكاملة بما في ذلك جميع المعاملات، المجموعات، والإجراءات المتاحة.

نافذة تفاصيل الفاتورة:

- **معلومات الفاتورة** - تظهر معرف الفاتورة، العنوان، التواريخ، حالة الدفع، وحالة الإلغاء
- **قائمة المعاملات** - تعرض جميع المعاملات المدرجة في الفاتورة مع:
  - تاريخ المعاملة
  - العنوان والوصف
  - تكلفة التجزئة
  - مبلغ الضريبة والنسبة المئوية (مُنسقة كـ `$10.00 (%10)`)
  - تظهر المعاملات المعفاة من الضريبة "-" في عمود الضريبة
- **ملخص المجموعات** - حساب في الوقت الحقيقي يظهر:
  - عدد المعاملات
  - المجموع الفرعي (مجموع جميع تكاليف التجزئة)
  - الضريبة (مجموع جميع مبالغ الضرائب)
  - إجمالي الفاتورة (المجموع الفرعي + الضريبة)
- **أزرار الإجراءات** - نفس الإجراءات المتاحة كما في الجدول:
  - **تنزيل PDF** - تنزيل PDF الفاتورة (متاح دائمًا)
  - **إرسال بريد إلكتروني** - إرسال الفاتورة إلى العميل (الفواتير غير الملغاة)
  - **دفع الفاتورة** - معالجة الدفع (الفواتير غير المدفوعة، غير الملغاة فقط)
  - **استرداد** Stripe - استرداد دفعة Stripe (فقط للفواتير المدفوعة عبر Stripe)
  - **حذف** - إلغاء الفاتورة (الفواتير غير المدفوعة، غير الملغاة فقط)

تنزيل PDFs الفواتير

في نافذة تفاصيل الفاتورة **"تنزيل PDF"** انقر على **أيقونة التنزيل (⬇)** في الجدول أو زر لتنزيل فاتورة كملف PDF.

عملية ا��تنزيل:

1. انقر على أيقونة التنزيل بجوار الفاتورة
2. يظهر مؤشر التحميل أثناء التوليد (في المرة الأولى فقط)
3. يطلب المتصفح حفظ الملف: `Invoice_01234.pdf`
4. أو يتم حفظه في مجلد التنزيلات PDF يفتح

سلوك التخزين PDF:

- من القالب، ويتم تخزينه في قاعدة البيانات (قد PDF **التنزيل الأول** - يتم إنشاء يستغرق 2-3 ثوانٍ)
- **التنزيلات اللاحقة** - تنزيل فوري من التخزين
- **إبطال التخزين** - يتم مسح التخزين إذا تم تعديل الفاتورة أو تحديث القالب

استكشاف مشكلات التنزيل:

- معطلة API **المؤشر لا يتوقف أبدًا** - تحقق من وحدة التحكم في المتصفح، قد تكون
- **PDF فارغ أو تالف** - تحقق من قالب الفاتورة بحثًا عن أخطاء بناء الجملة
- **فشل التنزيل** - تحقق من إعدادات حظر النوافذ المنبثقة، جرب متصفحًا مختلفًا

دفع الفواتير

انقر على **أيقونة الدفع (💳)** لدفع فاتورة عبر الإنترنت.

عملية الدفع:

1. انقر على أيقونة الدفع على فاتورة غير مدفوعة
2. تفتح نافذة الدفع مع عرض تفاصيل الفاتورة
3. اختر طريقة الدفع:
    - **معاملة Stripe** - خصم بطاقة الائتمان المحفوظة (متاحة لجميع المستخدمين)
    - **نقدًا** - دفع نقدي يدوي (للموظفين فقط)
    - **استرداد** - تطبيق الاسترداد كدفع (للموظفين فقط)
    - **معاملة POS** - جهاز نقطة البيع (للموظفين فقط)
    - **تحويل بنكي** - تحويل بنكي يدوي (للموظفين فقط)
4. إذا تم اختيار Stripe:
    - اختر بطاقة من طرق الدفع المحفوظة
    - يتم تحديد البطاقة الافتراضية مسبقًا
    - انقر لاختيار بطاقة مختلفة
5. إذا تم اختيار طريقة أخرى:
    - أدخل رقم المرجع (اختياري)
6. انقر على **"دفع الفاتورة"** لمعالجة
7. يقوم النظام بمعالجة الدفع:
    - **Stripe** - يخصم البطاقة عبر API Stripe
    - **طرق أخرى** - ينشئ معاملة سلبية لمبلغ الدفع
8. تتغير حالة الفاتورة إلى "مدفوعة"
9. يتم عرض إشعار النجاح

العناية الذاتية مقابل دفع الموظفين:

:doc:`بوابة العناية الذاتية <self_care_portal>` (العملاء):

- Stripe متاحة فقط طريقة الدفع عبر
- يجب أن يكون لديك طريقة دفع محفوظة
- يتم عرض تحذير إذا لم توجد طرق دفع
- يتم توفير رابط لإضافة طريقة دفع

بوابة الموظفين (المسؤولون):

- جميع طرق الدفع متاحة
- يمكنهم وضع علامة على الفاتورة كمدفوعة يدويًا (نقدًا، POS، تحويل بنكي)
- يمكنهم إدخال أرقام المرجع للتتبع

تحذير طريقة الدفع:

إذا لم يكن لدى العميل أي طرق دفع محفوظة، يتم عرض تحذير يطلب منهم إضافة طريقة دفع قبل أن يتمكنوا من دفع الفواتير.

إرسال الفواتير عبر البريد الإلكتروني

انقر على **أيقونة البريد الإلكتروني (✉)** لإرسال الفاتورة إلى العميل.

ما يحدث:

1. انقر على أيقونة البريد الإلكتروني بجوار الفاتورة

2. يسترجع النظام PDF (أو ينشئه إذا لم يكن مخزنًا) الفاتورة من التخزين
3. يرسل البريد الإلكتروني عبر Mailjet باستخدام قالب `api_crmCommunicationCustomerInvoice`
4. يتضمن البريد الإلكتروني:
    - PDF الفاتورة كمرفق
    - اسم العميل
    - رقم الفاتورة وتاريخ الاستحقاق
    - المبلغ الإجمالي المستحق
    - رابط لدفع الفاتورة عبر الإنترنت
    - رابط لعرض/تنزيل الفاتورة
5. إشعار النجاح: "تم إرسال بريد الفاتورة بنجاح"

المستلمون عبر البريد الإلكتروني:

يتم إرسال البريد الإلكتروني إلى جميع جهات اتصال العميل من نوع "الفوترة" أو جهة الاتصال الرئيسية إذا لم توجد جهة اتصال فوترة.

متغيرات قالب البريد الإلكتروني:

- `{{ var:customer_name }}` - الاسم الكامل للعميل
- `{{ var:invoice_number }}` - معرف الفاتورة
- `{{ var:invoice_date }}` - تاريخ إصدار الفاتورة
- `{{ var:due_date }}` - تاريخ استحقاق الدفع
- `{{ var:total_amount }}` - المبلغ الإجمالي المستحق
- `{{ var:invoice_url }}` - رابط لعرض/تنزيل PDF
- `{{ var:pay_url }}` - رابط لدفع الفاتورة عبر الإنترنت

استكشاف مشكلات البريد الإلكتروني:

- **لم يتم إرسال البريد الإلكتروني** - تحقق من بيانات اعتماد API Mailjet في `crm_config.yaml`
- **العميل لا يتلقى** - تحقق من عناوين البريد الإلكتروني لجهات اتصال العميل
- **لا يتم إرفاقه PDF** - تحقق من نجاح إنشاء PDF (جرب تنزيله أولاً)

إلغاء الفواتير

لإلغاء فاتورة **أيقونة الحذف (🗑)** انقر على.

المتطلبات:

- يجب أن تكون الفاتورة **غير مدفوعة**
- لا يمكن إلغاء الفواتير المدفوعة (يجب استردادها بدلاً من ذلك)

كيفية الإلغاء:

1. ابحث عن فاتورة غير مدفوعة في القائمة
2. انقر على أيقونة الحذف (🗑)
3. أكد في النافذة:

ما يحدث:

- تم وضع علامة على الفاتورة كـ `void = true` ي��م
- يتم فصل جميع المعاملات عن الفاتورة (`invoice_id` تم تعيينه إلى null)
- تصبح المعاملات "غير مفوترة" مرة أخرى
- يمكن تضمين المعاملات في فاتورة جديدة
- تظهر الفاتورة في القائمة مع بادئة "ملغاة:" في العنوان
- يتم تعطيل إجراءات الفاتورة (لا تنزيل، دفع، أو بريد إلكتروني)
- يمكن عرضها عن طريق تصفية الفواتير "الملغاة"

ملاحظات مهمة:

- الإلغاء ليس هو نفسه الاسترداد
- **الإلغاء** = "يجب ألا توجد هذه الفاتورة أبدًا" (خطأ في الفوترة، تكرار)
- **الاسترداد** = "عكس فاتورة مدفوعة صحيحة" (إرجاع المال للعميل)

استرداد الفواتير

انقر على **أيقونة الاسترداد (□)** لاسترداد فاتورة مدفوعة.

المتطلبات:

- يجب أن تكون الفاتورة **مدفوعة**
- يجب أن تكون الفاتورة مدفوعة عبر **Stripe**
- يجب أن تحتوي الفاتورة على `payment_reference` صالح (معرف نية الدفع من Stripe)
- متاحة ف��ط لمستخدمي الموظفين (ليس للعناية الذاتية)

كيفية الاسترداد:

1. ابحث عن فاتورة Stripe المدفوعة
2. انقر على أيقونة الاسترداد (□)
3. تفتح نافذة تأكيد الاسترداد:

4. انقر على **"تأكيد الاسترداد"**

5. يقوم النظام بمعالجة استرداد Stripe:
    - يستدعي API Stripe لاسترداد الدفع
    - ينشئ معاملة استرداد في Stripe
    - يحدث الفاتورة مع `refund_reference`
6. تتغير حالة الفاتورة إلى "مستردة"
7. يتم عرض إشعار النجاح

ما يحدث بعد الاسترداد:

- تبقى الفاتورة في النظام (غير ملغاة)
- تظهر الحالة "مستردة"
- تبقى المعاملات مرتبطة بالفاتورة
- يتلقى العميل الاسترداد إلى طريقة الدفع الأصلية (3-7 أيام عمل)
- يظهر لوحة معلومات Stripe مع معاملة الاسترداد

قيود الاسترداد:

- لا يمكن استرداد الفواتير المدفوعة نقدًا أو عبر POS (يتطلب الب��كي أو التحويل عكس يدوي)
- لا يمكن الاسترداد الجزئي (مبلغ الفاتورة بالكامل فقط)
- لا يمكن الاسترداد مرتين

البحث وتصفية الفواتير

البحث

استخدم شريط البحث للعثور على الفواتير. يتم البحث عبر:

- معرف الفاتورة
- عنوان الفاتورة
- اسم العميل (عرض على مستوى النظام فقط)

الفلاتر

تطبيق الفلاتر لتضييق قائمة الفواتير:

الفلاتر المتاحة:

- **حالة الإلغاء** - الكل، ملغاة، غير ملغاة
- **حالة الدفع** - الكل، مدفوعة، غير مدفوعة بعد

إجراءات الفلترة:

- **تطبيق الفلاتر** - تطبيق الفلاتر المحددة على القائمة
- **إعادة تعيين الفلاتر** - مسح جميع الفلاتر وعرض جميع الفواتير

الفرز

انقر على أي رأس عمود للفرز:

- **ID** - الفرز حسب معرف الفاتورة (الأحدث/الأقدم)
- **العنوان** - الفرز أبجديًا
- **تاريخ الاستحقاق** - الفرز حسب تاريخ الاستحقاق
- **تاريخ الإنشاء** - الفرز حسب تا��يخ الإنشاء
- **المبلغ** - الفرز حسب تكلفة التجزئة (الأعلى/الأدنى)
- **الحالة** - الفرز حسب حالة الدفع (المدفوعة أولاً أو غير المدفوعة أولاً)

انقر مرة أخرى لعكس اتجاه الفرز (تصاعدي ↔ تنازلي).

الصفحات

التنقل عبر قوائم الفواتير الكبيرة مع عناصر التحكم في الصفحات التي تظهر الصفحة الحالية، إجمالي الصفحات، ومحدد العناصر لكل صفحة (10، 25، 50، أو 100 عنصر).

سير العمل الشائع للفواتير

سير العمل 1: الفوترة الشهرية مع معاينة المعاملة

1. يصل نهاية الشهر (مثل، 31 يناير)
2. انتقل إلى **الفوترة** → **الفواتير**
3. انقر على **"+ إنشاء فاتورة أولية"**
4. اختر العميل (أو قم بذلك لكل عميل إذا كنت تفوتر العديد من العملاء)

5. تعيين التواريخ:
    - تاريخ البدء: 01-01-2025
    - تاريخ الانتهاء: 31-01-2025
    - تاريخ الاستحقاق: 15-02-2025 (15 يومًا من الآن)
    - العنوان: "خدمات يناير 2025" (اختياري)
6. ي��هر قسم **معاينة المعاملة** مع جميع المعاملات غير المفوترة من يناير
7. مراجعة المعاينة:
    - يتم تضمين جميع المعاملات افتراضيًا
    - تحقق من المجموعات: المجموع الفرعي، الضريبة، وإجمالي الفاتورة
    - تحقق من صحة جميع الرسوم
8. انقر على **"إنشاء فاتورة"** (يظهر عدد المعاملات، مثل، "إنشاء فاتورة (15)")
9. تم إنشاء الفاتورة مع جميع المعاملات المحددة
10. انقر على صف الفاتورة لعرض التفاصيل والتحقق
11. انقر على زر **"إرسال بريد إلكتروني"** في نافذة التفاصيل أو أيقونة البريد الإلكتروني في الجدول
12. ورابط الدفع PDF يتلقى العميل بريد الفاتورة مع

سير العمل 2: الفوترة الانتقائية للمعاملات

1. لدى العميل خدمات متعددة (الهاتف المحمول + الإنترنت) ورسوم متنوعة
2. يريد فواتير منفصلة لكل خدمة
3. **إنشاء الفاتورة الأولى (خدمات الهاتف المحمول):**
    - انقر على **"+ إنشاء فاتورة أولية"**
    - العنوان: "خدمات الهاتف المحمول - يناير 2025"
    - تاريخ البدء/الانتهاء: 1-31 يناير
    - تاريخ الاستحقاق: 15 فبراير
    - في معاينة المعاملة، **استبعد** جميع المعاملات غير المحمولة:
        - انقر على زر **X** بجوار معاملات الإنترنت
        - انقر على زر **X** بجوار الرسوم المتنوعة
        - تبقى فقط معاملات خدمات الهاتف المحمول محددة
    - تحقق من أن المجموعات تعكس فقط خدمات الهاتف المحمول
    - انقر على **"إنشاء فاتورة"** (يظهر عدد معاملات الهاتف المحمول)
4. **إنشاء الفاتورة الثانية (خدمات الإنترنت):**
    - انقر على **"+ إنشاء فاتورة أولية"** مرة أخرى

- "العنوان: "خدمات الإنترنت - يناير 2025
- تاريخ البدء/الانتهاء: 1-31 يناير (نفس الفترة)
- في معاينة المعاملة:
  - لا تظهر معاملات الهاتف المحمول التي تم فوترتها بالفعل (لا تظهر)
  - استبعد الرسوم المتنوعة باستخدام زر **X**
  - تبقى فقط معاملات خدمات الإنترنت
- انقر على **"إنشاء فاتورة"**

5. **إنشاء الفاتورة الثالثة (الرسوم الإضافية):**
   - انقر على **"إنشاء فاتورة أولية +"** مرة أخرى
   - "العنوان: "الرسوم الإضافية - يناير 2025
   - تظهر فقط الرسوم المتنوعة غير المفوترة في المعاينة
   - انقر على **"تحديد الكل"** لتضمين الجميع
   - انقر على **"إنشاء فاتورة"**
6. أرسل جميع الفواتير الثلاثة إلى العميل عبر البريد الإلكتروني

سير العمل 3: استبعاد المعاملات المتنازع عليها أو المعلقة

1. يصل نهاية فترة الفوترة
2. انتقل إلى علامة تبويب **الفوترة** للعميل
3. انقر على **"إنشاء فاتورة أولية +"**
4. تعيين تواريخ فترة الفوترة
5. تظهر معاينة المعاملة 20 معاملة
6. يتنازع العميل على رسوم واحدة وأخرى معلقة قيد التحقيق
7. في معاينة المعاملة:
   - حدد المعاملة المتنازع عليها (مثل، "رسوم تجاوز البيانات")
   - انقر على زر **X** لاستبعادها
   - حدد المعاملة المعلقة (مثل، "رسوم التثبيت")
   - انقر على زر **X** لاستبعادها
   - يتم تحديث عدد المعاملات: "18 معاملة محددة"
   - يتم إعادة حساب المجموعات تلقائيًا
8. مراجعة المجموعات المحدثة (تستبعد المبالغ المتنازع عليها)
9. انقر على **"إنشاء فاتورة (18)"**

10. تم إنشاء الفاتورة مع المعاملات المعتمدة فقط
11. تبقى المعاملات المتنازع عليها/المعلقة غير مفوترة للدورة التالية

سير العمل 4: مراجعة سريعة للفواتير وتعديلها

1. يقوم الموظف بإنشاء فاتورة شهرية
2. تظهر معاينة المعاملة إجماليًا مرتفعًا غير متوقع
3. مراجعة كل معاملة في المعاينة:
    - لاحظ رسومًا مكررة لنفس الخدمة
    - لاستبعاد المكررة **X** انقر على
    - لاحظ معاملة اختبار لا ينبغي فوترتها
    - لاستبعاد معاملة الاختبار **X** انقر على
4. يتم تحديث المجموعات في الوقت الحقيقي
5. تحقق من أن المجموع الجديد يتطابق مع المبلغ المتوقع
6. انقر على **"إنشاء فاتورة"** مع المعاملات المصححة
7. عد إلى الوراء وألغِ/احذف المعاملات المستبعدة إذا لزم الأمر
8. أرسل الفاتورة إلى العميل بثقة

سير العمل 5: فاتورة تثبيت لمرة واحدة

1. أكمل الفني الميداني التثبيت
2. يضيف الموظف معاملة التثبيت يدويًا
3. انتقل إلى علامة تبويب **الفوترة** للعميل
4. انقر على **"+ إنشاء فاتورة أولية"**
5. تعيين التواريخ:
    - تاريخ البدء: اليوم
    - تاريخ الانتهاء: اليوم
    - تاريخ الاستحقاق: اليوم + 7 أيام
    - العنوان: "خدمات التثبيت"
6. تظهر معاينة المعاملة فقط معاملات اليوم
7. تحقق من ظهور رسوم التثبيت
8. استبعد أي رسوم متكررة باستخدام زر **X** (إذا كانت موجودة)
9. انقر على **"إنشاء فاتورة"**
10. أرسلها إلى العميل على الفور

11. يدفع العميل عبر الإنترنت عبر Stripe

سير العمل 6: مراجعة الفاتورة قبل الاتصال بالعميل

1. يتصل العميل بسؤال عن الفوترة
2. يقوم الموظف بالتنقل إلى قائمة فواتير العميل
3. **انقر على صف الفاتورة** لفتح نافذة تفاصيل الفاتورة
4. مراجعة معلومات الفاتورة:
    - معرف الفاتورة، التواريخ، الحالة
    - جميع المعاملات المدرجة مع الأوصاف
    - تفاصيل الضريبة لكل معاملة
    - المجموع الفرعي، الضريبة، والمبالغ الإجمالية
5. أجب عن أسئلة العميل بالتفاصيل الدقيقة
6. إذا طلب العميل PDF، انقر على زر **"تنزيل PDF"** في النافذة
7. إذا طلب العميل إعادة إرسال البريد الإلكتروني، انقر على زر **"إرسال بريد إلكتروني"**
8. أغلق النافذة عند الانتهاء

سير العمل 7: تصحيح خطأ في الفوترة

1. يبلغ العميل عن رسوم غير صحيحة
2. ينقر الموظف على صف الفاتورة لعرض التفاصيل
3. مراجعة قائمة المعاملات في نافذة تفاصيل الفاتورة
4. تحديد المعاملة غير الصحيحة
5. الفاتورة غير مدفوعة، لذا يمكن إلغاؤها
6. انقر على زر **"حذف"** في تذييل النافذة
7. أكد الإلغاء
8. تصبح المعاملات غير مفوترة مرة أخرى
9. يقوم الموظف بتعديل أو إزالة المعاملة غير الصحيحة من قائمة المعاملات
10. إنشاء فاتورة جديدة مع المعاملات المصححة:
    - استخدم معاينة المعاملة لاستبعاد المعاملة المصححة إذا لزم الأمر
    - تضمين الرسوم الصحيحة فقط
11. أرسل الفاتورة المصححة إلى العميل

سير العمل 8: معالجة مدفوعات متعددة

1. يحضر العميل نقدًا لدفع فواتير متعددة
2. انتقل إلى علامة تبويب **الفوترة** للعميل
3. عرض الفواتير غير المدفوعة
4. انقر على صف الفاتورة الأولى لعرض التفاصيل
5. تحقق من المبلغ والمعاملات
6. انقر على زر **"دفع الفاتورة"** في تذييل النافذة
7. اختر طريقة الدفع **"نقدًا"**
8. أدخل المرجع: "تم الدفع نقدًا 15-01-2025"
9. انقر على **"دفع الفاتورة"**
10. تغلق النافذة، وتُعلّم الفاتورة كـ "مدفوعة"
11. كرر للفاتورات المتبقية
12. جميع الفواتير الآن مُعلمة كـ "مدفوعة"

سير العمل 9: التعامل مع طلب الاسترداد

1. يطلب العميل استردادًا لزيادة الدفع
2. يتحقق الموظف من أن الفاتورة تم دفعها عبر Stripe
3. انتقل إلى الفاتورة في القائمة
4. انقر على صف الفاتورة لعرض التفاصيل
5. تحقق من معلومات الدفع والمبلغ
6. انقر على زر **"استرداد"** في تذييل النافذة (يظهر فقط لفواتير Stripe)
7. أكد الاسترداد
8. يقوم النظام بمعالجة استرداد Stripe
9. تتغير حالة الفاتورة إلى "مستردة"
10. يتلقى العميل الاسترداد في 3-7 أيام عمل
11. يتابع الموظف مع العميل لتأكيد الاستلام

استكشاف الأخطاء وإصلاحها

لا يمكن إنشاء الفاتورة - لم يتم العثور على معاملات

- **السبب:** لا توجد معاملات غير مفوترة في نطاق التاريخ المحدد

- **الإصلاح:** تحقق من قائمة المعاملات، تحقق من وجود المعاملات وأنها ليست مفوترة بالفعل. قم بتعديل نطاق التاريخ أو إزالة الفلتر.

فشل إنشاء PDF الفاتورة

- **السبب:** خطأ في بناء جملة القالب، تعطل WeasyPrint، أو بيانات العميل مفقودة
- **الإصلاح:** تحقق من قالب الفاتورة HTML بحثًا عن أخطاء، تحقق من تعبئة حقول عنوان العميل، راجع سجلات API.

فشل الدفع مع خطأ Stripe

- **السبب:** تم رفض البطاقة، عدم كفاية الأموال، بطاقة منتهية، أو مشكلة في API Stripe
- **الإصلاح:** جرب طريقة دفع مختلفة، تحقق من صحة البطاقة، تحقق من لوحة معلومات Stripe لسبب الرفض.

لا يمكن إلغاء الفاتورة

- **السبب:** تم دفع الفاتورة بالفعل
- **الإصلاح:** لا يمكن إلغاء الفواتير المدفوعة. إذا كان الاسترداد مطلوبًا، استخدم وظيفة الاسترداد لفواتير Stripe أو أنشئ معاملة ائتمان يدويًا.

البريد الإلكتروني للفواتير لا يُرسل

- **السبب:** بيانات اعتماد API Mailjet غير صالحة، العميل ليس لديه جهة اتصال فوترة، أو قالب البريد الإلكتروني مفقود
- **الإصلاح:** تحقق من تكوين Mailjet في `crm_config.yaml`، تحقق من جهات اتصال العميل، تحقق من وجود قالب البريد الإلكتروني للفواتير.

زر الاسترداد لا يظهر

- **السبب:** تم دفع الفاتورة نقدًا/عبر POS/التحويل البنكي (ليس عبر Stripe)، أو لم يتم دفع الفاتورة
- **الإصلاح:** يظهر زر الاسترداد فقط لمدفوعات Stripe. بالنسبة لطرق الدفع الأخرى، أنشئ معاملة ائتمان يدوية.

يظهر PDF التنزيل تصميم القالب القديم

- **السبب:** تم تخزين PDF قبل تحديث القالب

- **الإصلاح:** قم بمسح تخزين PDF للفواتير: `DELETE FROM Invoice_PDF_Cache WHERE invoice_id = X;`

لا يمكن للعميل دفع الفاتورة (لا توجد طرق دفع)

- **السبب:** لا توجد طرق دفع محفوظة في بوابة العناية الذاتية
- **الإصلاح:** يجب على العميل إضافة بطاقة ائتمان في صفحة **طرق الدفع** قبل دفع الفواتير.

تم إنشاء فواتير متعددة لنفس الفترة

- **السبب:** قام الموظف بإنشاء فاتورة مرتين، أو تتداخل نطاقات التاريخ
- **الإصلاح:** قم بإلغاء الفاتورة المكررة. قم بتعديل نطاقات التاريخ لمنع التداخل. استخدم معاينة المعاملة لضمان مجموعات معاملات فريدة.

لا تظهر المعاملات في معاينة المعاملة

- **السبب:** جميع المعاملات في نطاق التاريخ تم فوترتها بالفعل أو لا توجد معاملات
- **الإصلاح:** تحقق من صحة نطاق التاريخ. تحقق من قائمة المعاملات للتأكد من وجود المعاملات. قم بتصفية الفواتير لرؤية أي فاتورة تحتوي على المعاملات.

لا يمكن استبعاد المعاملة من إنشاء الفاتورة

- **السبب:** تم فوترتها بالفعل أو مشكلة في المتصفح
- **الإصلاح:** تحقق من أن المعاملة تظهر في المعاينة مع علامة الاختيار. قم بتحديث الصفحة وحاول مرة أخرى. امسح ذاكرة التخزين المؤقت للمتصفح إذا استمرت المشكلة.

إجمالي الفاتورة لا يتطابق مع المبلغ المتوقع

- **السبب:** تم تضمين معاملات غير متوقعة، أو لم يتم حساب الضريبة، أو تم احتساب المعاملات المستبعدة
- **الإصلاح:** راجع المعاينة بعناية. تحقق من تكلفة التجزئة والضريبة لكل معاملة. تحقق من أن المعاملات المستبعدة تظهر بلون رمادي. تحقق من شارة عدد المعاملات على زر إنشاء الفاتورة.

زر إنشاء الفاتورة معطل

- **السبب:** لم يتم تحديد أي معاملات أو نطاق تاريخ غير صالح

- **الإصلاح:** تأكد من تضمين معاملة واحدة على الأقل (غير مستبعدة). تحقق من أن تاريخ البدء قبل تاريخ الانتهاء. تحقق من تعيين تاريخ الاستحقاق.

نافذة تفاصيل الفاتورة لا تفتح

- **السبب:** خطأ في JavaScript أو لم يتم تحميل الصفحة بالكامل
- **الإصلاح:** قم بتحديث الصفحة. تحقق من وحدة التحكم في المتصفح بحثًا عن الأخطاء. جرب متصفحًا مختلفًا. تحقق من اتصال الإنترنت.

الضريبة لا تظهر في تفاصيل الفاتورة

- **السبب:** المعاملة لديها ضريبة 0% أو `tax_amount` فارغ
- **الإصلاح:** تحقق من أن المعاملة تحتوي على `tax_percentage` محدد. تحقق من أنه تم حساب `tax_amount` عند إنشاء المعاملة. قم بتحديث المعاملة إذا لزم الأمر.

أزرار الإجراءات مفقودة في نافذة تفاصيل الفاتورة

- **السبب:** تم إلغاء الفاتورة أو يفتقر المستخدم إلى الأذونات
- **الإصلاح:** تظهر الفواتير الملغاة فقط زر تنزيل PDF. تحقق من حالة الفاتورة. تحقق من دور المستخدم والأذونات.

الوثائق ذات الصلة

- `integrations_mailjet` - تسليم الفواتير عبر البريد الإلكتروني والقوالب
- `administration_configuration` - تكوين قالب الفاتورة
- `payments_transaction` - إنشاء معاملات تظهر على الفواتير
- `payments_process` - معالجة مدفوعات الفواتير
- `basics_payment` - إدارة طرق الدفع
- `payment_system_guide` - مرجع API الدفع وتكوين البائع

معالجة المدفوعات

يتم معالجة معظم المدفوعات تلقائيًا بواسطة النظام. في بعض الأحيان، يجب عليك معالجة الدفع يدويًا.

لدفع فاتورة، حدد الفاتورة غير المدفوعة، وانقر على زر "دفع".

سيفتح هذا نموذج الدفع، حيث يمكنك إدخال طريقة الدفع، والنقر على "إرسال" لمعالجة الدفع.

سيتلقى العميل إيصالًا تلقائيًا للدفع، وسيتم وضع علامة على الفاتورة كمدفوعة.

بالنسبة للتحويلات البنكية، أدخل مرجع الدفع. أدخل أيضًا تاريخ إجراء الدفع، إذا كان مختلفًا عن التاريخ الحالي.

معاملات العملاء

أي شيء يكلف المال في النظام يتم تسجيله كمعاملة تحت العميل.

كل معاملة لها مبلغ مالي لتكلفة الجملة وتكلفة التجزئة، ووصف لما تتعلق به المعاملة.

يمكن أن يتم إنشاء المعاملات تلقائيًا بواسطة النظام، على سبيل المثال، عندما يتم توفير خدمة، يتم إنشاء معاملة لتكلفة الإعداد، وعندما يتم فوتره خدمة، يتم إنشاء معاملة لتكلفة التجزئة.

يمكن أيضًا إنشاء المعاملات يدويًا، على سبيل المثال، إذا تم منح العميل ائتمان، يتم إنشاء معاملة لمبلغ الائتمان، أو إذا تم فرض رسوم تركيب، يتم إنشاء معاملة لرسوم التركيب.

تتم تجميع المعاملات معًا لتشكيل الفواتير، والتي تُرسل إلى العميل للدفع.

الوصول إلى المعاملات

يمكن عرض المعاملات على مستوى النظام أو لكل عميل:

عرض لكل عميل:

1. انتقل إلى **العملاء** → **[اختر العميل]**
2. انقر على علامة التبويب **الفوترة**
3. عرض قائمة المعاملات في البطاقة الأولى

عرض على مستوى النظام:

1. انتقل إلى **الفوترة → المعاملات** (من القائمة الرئيسية)
2. عرض جميع المعاملات عبر جميع العملاء

أدوات إحصائيات المعاملات

في أعلى صفحة المعاملات، تعرض أربع بطاقات إحصائية ملخصات مالية:

وصف الأدوات:

- **إجمالي المعاملات** - مجموع جميع تكاليف التجزئة للمعاملات (على مر الزمن)
- **إجمالي المعاملات غير المفوترة** - مجموع المعاملات التي لم يتم تضمينها بعد في فاتورة
- **إجمالي المعاملات لهذا الشهر** - مجموع المعاملات التي تم إنشاؤها في هذا الشهر الميلادي
- **إجمالي المعاملات الشهر الماضي** - مجموع المعاملات التي تم إنشاؤها في الشهر الميلادي الماضي

تنسيق القيم:

- القيم التي تزيد عن 1,000: عرضها كـ "k" لاحقة (على سبيل المثال، 1.5$k)
- القيم التي تزيد عن 1,000,000: عرضها كـ "M" لاحقة (على سبيل المثال، 2.3$M)

- على سبيل المثال،) لاحقة "B" القيم التي تزيد عن 1,000,000,000: عرضها كـ 1.1$B)

قائمة المعاملات

تعرض جدول المعاملات جميع المعاملات مع الأعمدة التالية:

وصف الأعمدة:

- **ID** - معرف المعاملة الفريد
- **التاريخ** - تاريخ إنشاء المعاملة
- **العنوان** - اسم المعاملة القصير
- **الوصف** - وصف تفصيلي لما تتعلق به المعاملة
- **المبلغ** - تكلفة التجزئة (إيجابية للرسوم، سلبية للاعتمادات)
- **الفاتورة** - معرف الفاتورة إذا تم فوتره المعاملة (رابط قابل للنقر)
- **الحالة** - علامة تحقق إذا كانت مفوترة، شرطة إذا لم يتم فوترتها بعد

الإجراءات لكل صف:

كل صف يحتوي على قائمة إجراءات (⋮) مع الخيارات:

- **عرض التفاصيل** - يفتح نافذة تفاصيل المعاملة
- **تنزيل فاتورة PDF** - تنزيل PDF (فقط إذا كانت مفوترة)
- **إلغاء المعاملة** - وضع علامة على المعاملة كملغاة (فقط إذا لم تكن مفوترة)

أنواع المعاملات

تقع المعاملات ضمن فئتين رئيسيتين:

معاملات الخصم (الرسوم)

مبالغ إيجابية تزيد من رصيد العميل المستحق:

- **رسوم إعداد الخدمة** - رسوم لمرة واحدة عند توفير الخدمة
- **رسوم الخدمة الشهرية** - رسوم متكررة للخدمات

- **رسوم التركيب** - رسوم لزيارات الفنيين الميدانيين
- **رسوم المعدات** - رسوم للمودمات، أجهزة التوجيه، بطاقات SIM
- **رسوم التأخير في الدفع** - غرامات على الفواتير المتأخرة
- **رسوم يدوية** - رسوم مخصصة يضيفها الموظفون

معاملات الائتمان (المدفوعات/الاستردادات)

مبالغ سلبية تقلل من رصيد العميل المستحق:

- **المدفوعات النقدية** - دفع العميل نقدًا
- **مدفوعات البطاقة** - دفع العميل بواسطة بطاقة ائتمان/خصم
- **مدفوعات التحويل البنكي** - دفع العميل عبر التحويل البنكي
- **اعتمادات الحساب** - اعتمادات حسن النية، تعويض
- **الاستردادات** - الأموال التي تم إرجاعها للعميل
- **الخصومات** - خصومات ترويجية أو ولاء

إضافة معاملة يدويًا

انقر على **"+ إضافة معاملة"** لفتح نافذة إضافة المعاملة.

معاملة الخصم (رسوم):

معاملة الائتمان (مدفوعات/استردادات):

وصف الحقول:

- **نوع المعاملة** - اختر الخصم (رسوم) أو الائتمان (مدفوعات/استردادات)
- **نوع الائتمان** - إذا تم اختيار الائتمان، اختر طريقة الدفع (نقدي، بطاقة، تحويل بنكي)
- **العنوان** - اسم قصير للمعاملة (مطلوب)
- **الوصف** - شرح تفصيلي (اختياري)
- **تكلفة التجزئة** - المبلغ الذي يدفعه العميل (مطلوب، رقم إيجابي)
- **تكلفة الجملة** - تكلفتك (اختياري، لتتبع الهامش)
- **نسبة الضريبة** - معدل الضريبة المطبق على هذه المعاملة (اختياري، الافتراضي هو ضريبة المنتج أو 0%)
- **الخدمة** - ربط المعاملة بخدمة معينة (اختياري)
- **الموقع** - ربط المعاملة بموقع معين (اختياري)
- **تاريخ المعاملة** - تاريخ المعاملة (الافتراضي هو اليوم)

التحقق:

- العنوان وتكلفة التجزئة مطلوبان
- يجب أن تكون تكلفة التجزئة رقمًا إيجابيًا
- إذا تم اختيار نوع الائتمان، يجب اختيار نوع الائتمان

ما يحدث:

1. يتم إنشاء المعاملة في قاعدة البيانات
2. تظهر في قائمة معاملات العميل

3. تتضمن في عدد "المعاملات غير المفوترة"
4. متاحة للتضمين ف�� إنشاء الفاتورة التالية
5. يتم إنشاء إدخال في سجل النشاط

البحث وتصفية المعاملات

البحث

استخدم شريط البحث للعثور على المعاملات. يتم البحث عبر:

- معرف المعاملة
- العنوان
- الوصف
- معرف الفاتورة

الفلاتر

تطبيق الفلاتر لتضييق قائمة المعاملات:

الفلاتر المتاحة:

- **حالة الإلغاء** - الكل، ملغاة، غير ملغاة
- **حالة الفاتورة** - الكل، مفوترة، غير مفوترة

إجراءات الفلترة:

- **تطبيق الفلاتر** - تطبيق الفلاتر المحددة على القائمة
- **إعادة تعيين الفلاتر** - مسح جميع الفلاتر وعرض جميع المعاملات

الفرز

انقر على أي رأس عمود للفرز:

- **ID** - فرز حسب معرف المعاملة (الأحدث/الأقدم)
- **التاريخ** - فرز حسب تاريخ المعاملة
- **العنوان** - فرز أبجديًا

- **المبلغ** - فرز حسب تكلفة التجزئة (الأعلى/الأدنى)
- **الفاتورة** - فرز حسب معرف الفاتورة

(تصاعدي ↔ تنازلي) اتجاه الفرز لعك�انقر مرة أخرى.

إلغاء المعاملات

(وضع علامة عليها كملغاة) يمكن إلغاء المعاملات التي تمت إضافتها عن طريق الخطأ.

المتطلبات:

- يجب ألا تكون المعاملة مفوترة
- (يجب استردادها بدلاً من ذلك) بمجرد أن يتم فوترتها، لا يمكن إلغاء المعاملات

كيفية الإلغاء:

1. ابحث عن المعاملة في القائمة
2. (⋮) انقر على قائمة الإجراءات
3. اختر **"إلغاء المعاملة"**
4. أكد في النافذة

ما يحدث:

- يتم وضع علامة على المعاملة كـ `void = true`
- لم تعد تظهر في قائمة المعاملات الافتراضية

- مستبعدة من إنشاء الفاتورة
- يمكن عرضها عن طريق تصفية "المعاملات الملغاة"
- يتم خصمها من إجمالي "المعاملات غير المفوترة"

المعاملات المرتبطة بالمحفظة (`wallet_ledger_id` تم تعيينه):

- يتم عكس رصيد المحفظة قبل الإلغاء (خصم لإلغاء ائتمان، ائتمان لإلغاء خصم)
- **إضافة أموال بطاقة الشحن** (`payment_type=wallet_topup_card`): تم خصم المحفظة وتم استرداد البطاقة
- **إضافة نقدي / دفع فاتورة المحفظة / خطوط محفظة أخرى:** تم تعديل المحفظة فقط؛ **لا يوجد استرداد للبطاقة**
- **محفظة غير كافية:** يفشل الإلغاء إذا كانت الأموال التي تم شحنها قد تم إنفاقها بالفعل (تظل المعاملة نشطة)

رسوم الخدمة / التوفير `wallet_ledger_id` من الالتقاط الهجين عادةً ما لا تحتوي على على خط الفوترة (البطاقة → المحفظة → الخصم يحدث داخليًا). إلغاء تلك الخطوط يضع فقط `void=true` اليوم؛ تم التخطيط لائتمان كامل للمحفظة لاستردادات الخدمة في فرع الوثائق المالية.

API: `DELETE /crm/transaction/transaction_id/{transaction_id}`. انظر دليل نظام الدفع: إلغاء المعاملة.

ملاحظة: الإلغاء ليس هو نفسه الاسترداد. الإلغاء يعني "يجب ألا تكون هذه المعاملة موجودة أبدًا." الاسترداد يعني "عكس معاملة صالحة."

الضريبة على المعاملات

يمكن أن تشمل المعاملات ضريبة، والتي يتم حسابها تلقائيًا بناءً على تكوين ضريبة المنتج أو يتم تحديدها يدويًا لكل معاملة.

سلوك الضريبة:

- **معاملات الخصم (الرسوم)**: - يتم تطبيق الضريبة على الرسوم بناءً على
  - **نسبة ضريبة المنتج** - إذا كانت المعاملة مرتبطة بمنتج، يتم تطبيق نسبة الضريبة الخاصة بالمنتج تلقائيًا
  - **تجاوز يدوي** - يمكن للموظفين تجاوز نسبة الضريبة عند إنشاء معاملة
  - **مبلغ الضريبة**: - يتم حسابه كالتالي: `retail_cost × (tax_percentage / 100)`
  - **تنسيق العرض** - يظهر كالتالي: `10.00$ (%10)` في قوائم المعاملات
- **معاملات الائتمان (المدفوعات/الاستردادات)** - لا يتم تطبيق ضريبة على الاعتمادات
  - يتم إخفاء حقل نسبة الضريبة للمعاملات الائتمانية
  - يتم تعيين الضريبة تلقائيًا إلى 0% لجميع المدفوعات والاستردادات
  - تقلل الاعتمادات من الرصيد المستحق للعميل دون آثار ضريبية

مثال على حساب الضريبة:

- المنتج: خطة موبايل مع ضريبة 10%، 50.00$ تكلفة التجزئة
- حساب الضريبة التلقائي: 50.00$ × 0.10 = 5.00$
- العرض: 5.00$ (10%)

صفر ضريبة (NIL/معفى):

- يمكن أن تكون المنتجات معفاة من الضريبة عن طريق تعيين نسبة الضريبة إلى 0

- الضريبة الافتراضية هي 0% إذا لم يتم تحديدها
- تظهر المعاملات المعفاة من الضريبة كـ "-" في عمود الضريبة

عرض تفاصيل المعاملة

انقر على معاملة لعرض التفاصيل الكاملة:

المعاملات المفوترة مقابل غير المفوترة

المعاملات غير المفوترة:

- لم يتم تضمينها بعد في أي فاتورة
- متاحة لإنشاء الفاتورة التالية
- يمكن إلغاؤها
- تحسب ضمن إجمالي "المعاملات غير المفوترة"
- الحالة تظهر شرطة (-)

المعاملات المفوترة:

- تم تضمينها في فاتورة
- لا يمكن إلغاؤها (يجب استردادها إذا لزم الأمر)
- معرف الفاتورة قابل للنقر (يرتبط بتفاصيل الفاتورة)

- الحالة تظهر علامة تحقق (✓)
- لا يمكن تعديلها

إنشاء الفاتورة:

عند إنشاء فاتورة لعميل:

1. يجد النظام جميع المعاملات غير المفوترة لذلك العميل
2. يمكن تصفية حسب نطاق التاريخ
3. يتم تضمين المعاملات في الفاتورة الجديدة
4. يتم ملء حقل `invoice_id` للمعاملة
5. يتم الآن وضع علامة على المعاملة كـ "مفوترة"

انظر `payments_invoices` لتفاصيل إنشاء الفاتورة.

سير العمل الشائعة

سير العمل 1: ائتمان يدوي لعدم توفر الخدمة

1. يتصل العميل: "كانت الخدمة معطلة لمدة يومين"
2. يقرر الموظف منح ائتمان بقيمة 10£
3. انتقل إلى علامة التبويب **الفوترة** للعميل
4. انقر على "**+ إضافة معاملة**"
5. اختر نوع المعام��ة **ائتمان**
6. اختر نوع ائتمان **دفع نقدي**
7. أدخل العنوان: "ائتمان عدم توفر الخدمة"
8. أدخل الوصف: "تعويض عن انقطاع الخدمة لمدة يومين 8-9 يناير"
9. أدخل تكلفة التجزئة: 10.00
10. اختر الخدمة المتأثرة من القائمة المنسدلة
11. انقر على "**إضافة معاملة**"
12. تظهر المعاملة بمبلغ -10.00£
13. سيتم تضمينها في الفاتورة التالية كائتمان

سير العمل 2: رسوم تركيب يدوية

1. يقوم الفني الميداني بتركيب الخدمة
2. يحتاج الموظف إلى فرض رسوم تركيب بقيمة 75£
3. انتقل إلى علامة التبويب **الفوترة** للعميل
4. انقر على "**+ إضافة معاملة**"
5. اختر نوع المعاملة **خصم**
6. أدخل العنوان: "رسوم التركيب"
7. أدخل الوصف: "زيارة فني ميداني لتركيب الألياف"
8. أدخل تكلفة التجزئة: 75.00
9. أدخل تكلفة الجملة: 45.00 (اختياري، لتتبع الهامش)
10. اختر الخدمة التي تم تركيبها
11. اختر الموقع الذي تم التركيب فيه
12. انقر عل**** "**إضافة معاملة**"
13. تظهر المعاملة في قائمة غير المفوترة
14. سيتم تضمينها في الفاتورة التالية

سير العمل 3: إلغاء معاملة مكررة

1. يلاحظ الموظف معاملة مكررة
2. تحقق من أن المعاملة لم يتم فوترتها بعد
3. انقر على قائمة الإجراءات (⋮) على المعاملة المكررة
4. اختر "**إلغاء المعاملة**"
5. أكد في النافذة
6. تتم إزالة المعاملة من القائمة
7. ينخفض إجمالي غير المفوتر وفقًا لذلك

سير العمل 4: العثور على المعاملات لفاتورة

1. تحتاج إلى إنشاء فاتورة شهرية
2. انقر على **فلتر الفاتورة:** "**غير مفوتر**"
3. انقر على **تطبيق الفلاتر**
4. عرض جميع المعاملات غير المفوترة
5. لاحظ المبلغ الإجمالي من الأدوات

6. انتقل لإنشاء الفاتورة
7. اختر نطاق التاريخ (على سبيل المثال، 1-31 يناير)
8. يتم تضمين المعاملات في النطاق في الفاتورة

استكشاف الأخطاء وإصلاحها

لا يمكن إلغاء ا��معاملة

- **السبب:** تم فوترت المعاملة بالفعل
- **الإصلاح:** المعاملة جزء من تاريخ الفواتير. إذا كان الاسترداد مطلوبًا، قم بإنشاء معاملة ائتمان بدلاً من ذلك.
- **السبب:** فشل عكس شحن المحفظة (رصيد المحفظة غير كافٍ)
- **الإصلاح:** أنفق العميل الأموال التي تم شحنها. استخدم خصم/تعديل يدوي للمحفظة أو تدفقات عكسية للوثائق المالية؛ لا يمكن إلغاء خط الشحن حتى تغطي المحفظة المبلغ.
- **السبب:** فشل استرداد البطاقة (فقط لشحن إضافة الأموال)
- **الإصلاح:** يتم إلغاء عكس المحفظة تلقائيًا. حل مشكلة بائع الدفع وحاول الإلغاء مرة أخرى.

ظهور معاملات مكررة

- **السبب:** تم فرض رسوم على الخدمة عدة مرات أو خطأ في التوفير
- **الإصلاح:** إلغاء المعاملة (المعاملات) المكررة إذا لم تكن مفوترة. إذا كانت مفوترة، قم بإصدار ائتمان.

المعاملة لا تظهر في القائمة

- **السبب:** تم تطبيق الفلاتر أو تم إلغاء المعاملة
- **الإصلاح:** انقر على "إعادة تعيين الفلاتر" لعرض جميع المعاملات. لرؤية المعاملات الملغاة، قم بتصفية "ملغاة: ملغاة".

الإجمالي غير المفوتر لا يتطابق مع المتوقع

- **السبب:** تم فوترت بعض المعاملات بالفعل، أو تم استبعاد المعاملات الملغاة

- **الإصلاح:** تطبيق فلتر "الفاتورة: غير مفوترة" لرؤية غير المفوتر فقط. تحقق من المعاملات الملغاة بشكل منفصل.

لا يمكن إضافة معاملة (حقل العميل معطل)

- **السبب:** عرض صفحة معاملات محددة للعميل
- **الإصلاح:** تم اختيار العميل مسبقًا. إذا كنت بحاجة إلى إضافة معاملة لعميل مختلف، انتقل إلى صفحة المعاملات على مستوى النظام.

الوثائق ذات الصلة

- `payments_invoices` - إنشاء وإدارة الفواتير
- `payments_process` - معالجة المدفوعات مقابل الفواتير
- `basics_payment` - نظرة عامة على طرق الدفع
- `csa_activity_log` - عرض تاريخ المعاملات في سجل النشاط

الشحن والمدفوعات من Playbooks

تشرح هذه الدليل كيفية تنفيذ معالجة الشحن والمدفوعات داخل Playbooks الخاصة بـ Ansible لعمليات توفير OmniCRM.

> **مرجع API الكامل**: للحصول على تفاصيل كاملة حول واجهات برمجة التطبيقات الخاصة بالمدفوعات، وتوجيه المحفظة، والمبالغ المستردة، والهندسة المعمارية المستقلة عن البائع، راجع **دليل نظام المدفوعات API**

نظرة عامة

يمكن لـ Playbooks الخاصة بـ OmniCRM Ansible التعامل مع معالجة المدفوعات بعدة طرق مختلفة، ولكن جميعها تقوم فقط باستدعاء واجهات برمجة التطبيقات على CRM للشحن.

1. **تدفق دفع الالتزام ذو المرحلتين** - للخدمات المدفوعة التي تتطلب تفويضًا فوريًا للدفع والاحتفاظ (أي الخدمات المدفوعة مسبقًا)
2. **إنشاء معاملة مباشرة** - لإضافة رسوم / ائتمانات دون معالجة الدفع الفورية (مثل Postpad) أي خدمات) رسوم الإعداد، الائتمانات اليدوية) والتي يمكن معالجتها لاحقًا

تضمن هذه الأساليب معاملات ذرية حيث يتم شحن العملاء فقط إذا نجح التوفير، مع التراجع التلقائي إذا فشل أي خطوة.

كأداة برمجة Playbooks: مرونة التسعير

مهم: القيم بالدولار المحددة في المنتجات والخدمات (مثل `retail_cost`، `wholesale_cost`، `retail_setup_cost`) هي ببساطة **قيم افتراضية** مخزنة في قاعدة البيانات. لا تحدد ما يجب بتحكم كامل على التسعير النهائي Playbook عليك شحنه. يتمتع.

يعتبر Playbook الخاص بـ Ansible بمثابة **محرك برمجة** يمكنك تخصيصه لتلبية أي متطلبات منطق العمل. يمكنك:

- **استخدام الأسعار المخزنة كما هي** - ببساطة قم بالإشارة إلى `api_response_product.json.retail_cost` في تفويضك

- **تجاوز الأسعار تمامًا** - شحن مبلغ مختلف بغض النظر عما هو موجود في تعريف المنتج
- **ت��بيق خصومات ديناميكية** - حساب النسب المئوية بناءً على مستوى العميل، أو العروض الترويجية، أو الولاء
- **تنفيذ تسعير متدرج** - شحن معدلات مختلفة بناءً على الكمية، أو مستويات الاستخدام، أو شروط العقد
- **تسعير مجمع** - دمج منتجات متعددة مع خصومات مجمعة مخصصة
- **تسعير قائم على الوقت** - تعديل الأسعار بناءً على الوقت من اليوم، أو الموسم، أو فترات العروض الترويجية
- **تسعير خاص بالعميل** - البحث عن الأسعار المتفاوض عليها من عقود العملاء

مثال: تجاوز سعر المنتج

```
# سعر المنتج retail_cost هو 99.00 دولار، لكننا نريد شحن 79.00 دولار كعرض ترويجي
- name: Set promotional price (ignoring product's retail_cost)
  set_fact:
    charge_amount: 79.00

- name: Authorize promotional payment
  uri:
    url: "http://localhost:5000/crm/payments/authorize/hold"
    method: POST
    body_format: json
    body:
      customer_id: "{{ customer_id | int }}"
      amount: "{{ charge_amount | float }}"  # باستخدام تجاوزنا، وليس retail_cost
      # ...
```

مثال: خصم مستوى العميل

```
# تطبيق خصم بناءً على مستوى العميل
- name: Get customer tier
  uri:
    url: "http://localhost:5000/crm/customer/{{ customer_id }}"
    # ...
  register: customer_info

- name: Calculate tier discount
  set_fact:
    base_price: "{{ api_response_product.json.retail_cost | float
}}"
    discount_percent: >-
      {% if customer_info.json.tier == 'gold' %}20
      {% elif customer_info.json.tier == 'silver' %}10
      {% else %}0{% endif %}

- name: Apply discount to final price
  set_fact:
    final_price: "{{ (base_price | float) * (1 - (discount_percent
| float / 100)) | round(2) }}"
```

مثال: تسعير الشركاء بالجملة

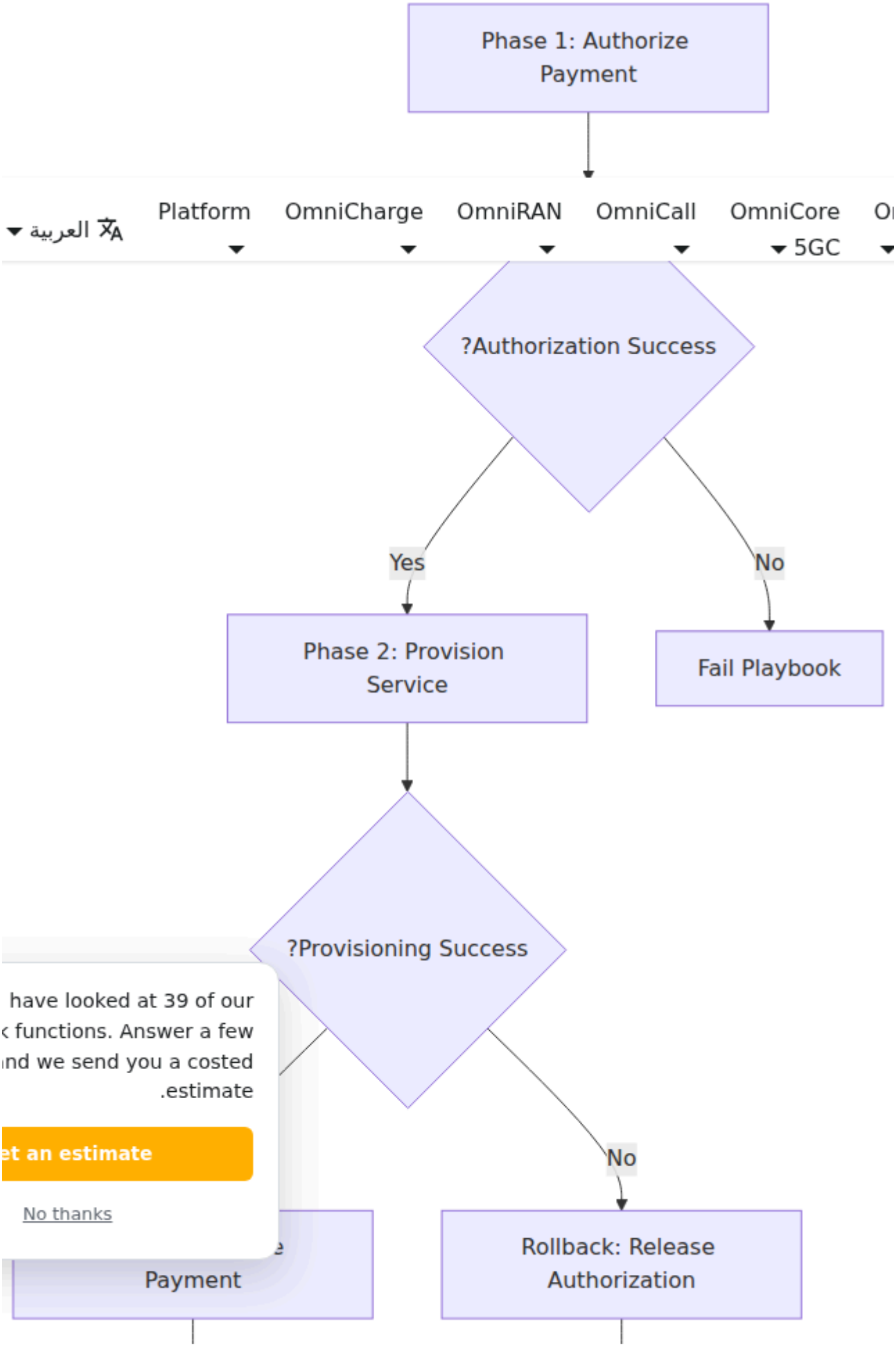
```
# يدفع الشركاء بالجملة wholesale_cost بدلاً من retail_cost
- name: Determine price based on customer type
  set_fact:
    charge_amount: >-
      {% if customer_info.json.is_wholesale_partner %}
      {{ api_response_product.json.wholesale_cost | float }}
      {% else %}
      {{ api_response_product.json.retail_cost | float }}
      {% endif %}
```

التحكم البرمجي Playbook النقطة الأساسية هي أنك **لست مقيدًا بأي نموذج تسعير**. يمنحك هي مجرد CRM الكامل لتنفيذ أي قواعد عمل تتطلبها مؤسستك. أسعار المنتجات / الخدمات في قيم افتراضية ملائمة يمكن لـ Playbooks استخدامها أو تجاهلها حسب الحاجة.

تدفق دفع الالتزام ذو المرحلتين

يتم استخدام نمط الالتزام ذو المرحلتين عندما تحتاج إلى شحن طريقة دفع العميل أثناء التوفير. يضمن ذلك أن يتم شحن العميل فقط إذا اكتمل التوفير بنجاح.

نظرة عامة على التدفق



have looked at 39 of our
k functions. Answer a few
nd we send you a costed
.estimate
et an estimate
No thanks



نمط التنفيذ

يتبع النمط هذه المراحل:

المرحلة 1: التفويض - احتجاز الأموال على طريقة الدفع
المرحلة 2: التوفير - تنفيذ تحديثات الرصيد / الإجراء OCS
المرحلة 3: الالتقاط - إنهاء الدفع عند نجاح التوفير
التراجع - تحرير التفويض إذا فشل التوفير

مثال كامل

إليك مثال كامل من `play_topup_charge_then_action.yaml`:

```yaml
- name: Play Topup - Charge card then action the Topup
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Include vars of crm_config
      ansible.builtin.include_vars:
        file: "../../crm_config.yaml"
        name: crm_config

    # الحصول على معلومات المنتج والخدمة
    - name: Get Product information from CRM API
      uri:
        url: "http://localhost:5000/crm/product/product_id//{{
product_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
        validate_certs: no
      register: api_response_product

    - name: Get Service information from CRM API
      uri:
        url: "http://localhost:5000/crm/service/service_id/{{
service_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
        validate_certs: no
      register: api_response_service

    - name: Set service and package facts
      set_fact:
        service_uuid: "{{ api_response_service.json.service_uuid
}}"
        customer_id: "{{ api_response_service.json.customer_id }}"
        package_name: "{{ api_response_product.json.product_name
}}"
        monthly_cost: "{{ api_response_product.json.retail_cost
}}"
```

```
        wholesale_cost: "{{
api_response_product.json.wholesale_cost }}"

    # الحصول على طريقة الدفع الافتراضية للعميل
    - name: Get the Customer Payment Methods from Payment
Controller
      uri:
        url: "http://localhost:5000/crm/payments/methods?
customer_id={{ customer_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
        validate_certs: no
      register: api_response_payment_methods

    - name: Get the default payment_method_id from the response
      set_fact:
        payment_method_id: "{{ api_response_payment_methods.json |
json_query(query) }}"
      vars:
        query: "data[?is_default==`true`].payment_method_id | [0]"

    # ============================================================
    # تدفق دفع الالتزام ذو المرحلتين
    # ============================================================

    - name: "Phase 1: Authorize payment (hold funds)"
      uri:
        url: "http://localhost:5000/crm/payments/authorize/hold"
        method: POST
        headers:
          Content-Type: "application/json"
          Authorization: "Bearer {{ access_token }}"
        body_format: json
        body:
          {
            "customer_id": "{{ customer_id | int }}",
            "amount": "{{ monthly_cost | float }}",
            "currency": "{{ crm_config.currency | default('AUD')
}}",
            "payment_method_id": "{{ payment_method_id | int }}",
            "metadata": {
              "description": "{{ package_name }} on {{
```

```
api_response_service.json.service_name }}",
              "service_id": "{{
api_response_service.json.service_id | int }}",
              "site_id": "{{ api_response_service.json.site_id |
int }}",
              "product_id": "{{ product_id }}",
              "user_id": "{{ (initiating_user | int) if
(initiating_user is defined and initiating_user is not none) else
omit }}",
              "title": "{{ package_name }}",
              "wholesale_cost": "{{ wholesale_cost | float }}",
              "invoice": true,
              "contract_days": "{{ days | int }}",
              "send_email": true
            }
          }
        return_content: yes
      register: api_response_authorization

    - name: Assert authorization was successful
      assert:
        that:
          - api_response_authorization.status == 200
          - api_response_authorization.json.success == true

    - name: Store authorization_id for capture/release
      set_fact:
        authorization_id: "{{
api_response_authorization.json.data.authorization_id }}"

    # توفير 2: ال��رحلة 2 OCS (مغلفة في كتلة للتراجع المعاملاتي)
    - block:
        - name: "Phase 2: Execute CGRateS action"
          uri:
            url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
            method: POST
            body_format: json
            headers:
              Content-Type: "application/json"
              Authorization: "Bearer {{ access_token }}"
            body:
              {
                "id": "{{ 999999999 | random }}",
                "method": "APIerSv1.ExecuteAction",
```

```
                "params": [{
                  "Tenant": "{{ crm_config.ocs.ocsTenant }}",
                  "Account": "{{ service_uuid }}",
                  "ActionsId": "{{ cgr_action_name }}"
                }]
              }
            status_code: 200
          register: action_execute_response

        - name: Assert that the response from the action is OK
          assert:
            that:
              - action_execute_response.status == 200
              - action_execute_response.json.result == "OK"

        # المرحلة 3: التقاط الدفع - إنهاء المعاملة بعد نجاح التوفير
        - name: "Phase 3: Capture authorized payment"
          uri:
            url: "http://localhost:5000/crm/payments/capture/{{
authorization_id }}"
            method: POST
            headers:
              Content-Type: "application/json"
              Authorization: "Bearer {{ access_token }}"
            body_format: json
            body:
              {
                "metadata": {
                  "provisioning_status": "success",
                  "cgr_action": "{{ cgr_action_name }}"
                }
              }
            return_content: yes
          register: api_response_capture

        - name: Assert capture was successful
          assert:
            that:
              - api_response_capture.status == 200
              - api_response_capture.json.success == true

      rescue:
        # التراجع عن المعاملة: إبطال التفويض لتحرير الأموال المحتجزة
        - name: "Rollback: Release payment authorization"
```

```
          uri:
            url: "http://localhost:5000/crm/payments/release/{{
authorization_id }}"
            method: POST
            headers:
              Content-Type: "application/json"
              Authorization: "Bearer {{ access_token }}"
            body_format: json
            body:
              {
                "metadata": {
                  "release_reason": "provisioning_failed",
                  "cgr_action": "{{ cgr_action_name }}"
                }
              }
            return_content: yes
          register: api_response_release

        - name: Terminate playbook execution after rollback
          fail:
            msg: "تم إبطال تفويض الدفع {{ authorization_id }}. فشل توفير OCS. لم يتم شحن العميل."
```

نقاط نهاية API للمدفوعات

> **توجيه المحفظة أولاً**: يقوم نظام الدفع تلقائيًا بتحسين رسوم البطاقات باستخدام رصيد المحفظة أولاً. إذا كان لدى العميل 1 دولار في محفظته واشترى إضافة بقيمة 10 دولارات، يتم شحن 9 دولارات فقط إلى بطاقته. راجع قسم توجيه المحفظة أولاً للحصول على التفاصيل.

1. تفويض / احتجاز الدفع

نقطة النهاية: `POST /crm/payments/authorize/hold`

تضع هذه النقطة احتجازًا على طريقة دفع العميل للمبلغ المحدد. يتم حجز الأموال ولكن لم يتم التقاطها بعد.

سلوك المحفظة أولاً: يقوم التفويض تلقائيًا بحساب النقص بعد ��لتحقق من رصيد المحفظة. إذا كان رصيد المحفظة 150 دولارًا وكان التفويض بقيمة 500 دولار، يتم تفويض 350 دولارًا فقط

على البطاقة. يتم خصم المحفظة عند وقت الالتقاط.

جسم الطلب:

```
{
  "customer_id": 123,
  "amount": 49.99,
  "currency": "AUD",
  "payment_method_id": 456,
  "metadata": {
    "description": "اسم الحزمة ووصفها",
    "service_id": 789,
    "site_id": 12,
    "product_id": "34",
    "user_id": 5,
    "title": "اسم الحزمة",
    "wholesale_cost": 25.00,
    "invoice": true,
    "contract_days": 30,
    "send_email": true
  }
}
```

وصف الحقول:

- **customer_id** (مطلوب) - معرف العميل الذي يتم شحنه
- **amount** (مطلوب) - المبلغ الذي سيتم تفويضه بتنسيق عشري
- **currency** (اختياري) - رمز العملة (يتم تعيينه افتراضيًا إلى الافتراضي للنظام)
- **payment_method_id** (مطلوب) - طريقة الدفع التي سيتم شحنها
- **metadata** (اختياري) - بيانات إضافية لإنشاء المعاملة:
  - **description** - وصف المعاملة
  - **service_id** - الخدمة التي يتم شحنها
  - **site_id** - الموقع المرتبط بالخدمة
  - **product_id** - المنتج الذي يتم توفيره
  - **user_id** - (اختياري) المستخدم الذي يقوم ببدء الشحن
  - **title** - عنوان المعاملة
  - **wholesale_cost** - (لتتبع الهامش) تكلفتك
  - **invoice** - إذا كانت `true`، يتم إنشاء المعاملة تلقائيًا عند الالتقاط

- **contract_days** - مدة العقد
- **send_email** - إذا كانت `true`، يتم إرسال إشعار عبر البريد الإلكتروني

الاستجابة:

```
{
  "success": true,
  "message": "(تم إنشاء الاحتجاز) تم تفويض الدفع",
  "data": {
    "authorization_id": 301,
    "vendor_authorization_id": "auth_xxxxx",
    "amount": 500.00,
    "currency": "USD",
    "status": "authorized",
    "wallet_balance": 150.00,
    "wallet_to_use": 150.00,
    "card_amount": 350.00,
    "message": "(تعبئة المحفظة) تم تفويض البطاقة بمبلغ 350 دولارًا
سيتم خصم 500 دولار من المحفظة عند الالتقاط."
  }
}
```

مهم:

- احفظ `authorization_id` لاستخدامه في عمليات الالتقاط أو الإلغاء
- لاحظ أن `card_amount` يظهر أن النقص فقط تم تفويضه
- يتم التحقق من رصيد المحفظة ولكن لا يتم خصمه حتى وقت الالتقاط

2. التقاط الدفع

نقطة النهاية: `POST /crm/payments/capture/{authorization_id}`

تقوم هذه النقطة بإنهاء تفويض الدفع وشحن العميل. إذا تم تعيين `invoice: true` في بيانات التعريف الخاصة بالتفويض، يتم إنشاء معاملة تلقائيًا.

جسم الطلب:

```
{
  "metadata": {
    "provisioning_status": "success",
    "cgr_action": "Action_Topup_Standard",
    "additional_info": "أي بيانات ذات صلة أخرى"
  }
}
```

الاستجابة:

```
{
  "success": true,
  "data": {
    "payment_id": "pay_xyz789",
    "transaction_id": 1234
  }
}
```

ماذا يحدث:

1. **تم التقاط البطاقة** للمبلغ الناقص (إذا تم تفويض البطاقة)
2. **تم ائتمان المحفظة** بمبلغ البطاقة الملتقطة
3. **تم خصم المحفظة** للمبلغ الكامل للخدمة
4. إذا كانت `invoice: true` في بيانات التعريف الخاصة بالتفويض:
   - **تم إنشا�� معاملة الخصم** (مبلغ إيجابي = شحن)
   - **تم إنشاء فاتورة** مرتبطة بمعاملة الخصم
   - **تم إنشاء معاملة ائتمان** (مبلغ سالب = المدفوعات المستلمة)
   - تم وضع علامة على الفاتورة كمدفوعة (تكون الخصومات والائتمانات صافية إلى صفر)
   - تظهر المعاملة في فواتير العميل
5. يتم إنشاء سجل الدفع في قاعدة البيانات
6. إذا كانت `send_email: true`، يتلقى العميل بريدًا إلكترونيًا بالفاتورة

مثال: تفويض بقيمة 500 دولار مع رصيد محفظة بقيمة 150 دولار:

- تم التقاط البطاقة: 350 دولار
- تم ائتمان المحفظة: +350 دولار (الآن رصيد المحفظة 500 دولار)

- تم خصم المحفظة: -500 دولار (رسوم الخدمة)
- المحفظة النهائية: 0 دولار

راجع أمثلة توجيه المحفظة أولاً للحصول على تدفق مفصل.

3. إلغاء تفويض الدفع

نقطة النهاية: `POST /crm/payments/release/{authorization_id}`

تقوم هذه النقطة بإلغ��ء تفويض الدفع وإطلاق الأموال المحتجزة. استخدم هذا في سيناريوهات الإنقاذ / التراجع عندما يفشل التوفير.

جسم الطلب:

```
{
  "metadata": {
    "release_reason": "provisioning_failed",
    "error_details": "فشل إنشاء حساب OCS"
  }
}
```

الاستجابة:

```
{
  "success": true,
  "message": "تم إطلاق التفويض"
}
```

ماذا يحدث:

1. **تم إطلاق تفويض البطاقة** (إذا كانت البطاقة قد تم تفويضها)
2. تم إرجاع الأموال المحتجزة إلى رصيد العميل المتاح
3. **لم يتم خصم المحفظة** (لأن الخصم يحدث فقط عند الالتقاط)
4. لم يتم شحن العميل
5. لم يتم إنشاء أي معاملة

ملاحظة: مع توجيه المحفظة أولاً، لا حاجة لاسترداد المحفظة حيث لا يتم خصم المحفظة حتى وقت الالتقاط.

إنشاء معاملة مباشرة

بالنسبة للرسوم التي لا تتطلب معالجة الدفع (رسوم الإعداد، الائتمانات اليدوية، التعديلات)، يمكنك إنشاء معاملات مباشرة عبر API.

إضافة معاملة عبر API

نقطة النهاية: `PUT /crm/transaction/`

مثال من `play_simple_service.yaml`:

```
- name: Add Setup Cost Transaction via API
  uri:
    url: "http://localhost:5000/crm/transaction/"
    method: PUT
    headers:
      Content-Type: "application/json"
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "customer_id": {{ customer_id | int }},
        "service_id": {{ service_creation_response.json.service_id
| int }},
        "title": "{{ package_name }} - Setup Costs",
        "description": "Setup Costs for {{ package_comment }}",
        "invoice_id": null,
        "wholesale_cost": {{
api_response_product.json.wholesale_setup_cost | float }},
        "retail_cost": "{{
api_response_product.json.retail_setup_cost | float }}"
      }
    return_content: yes
  register: api_response_transaction

- name: Assert that the response from the transaction is OK
  assert:
    that:
      - api_response_transaction.status == 200
```

حقول جسم الطلب:

- **customer_id** (مطلوب) - معرف العميل
- **service_id** (اختياري) - معرف الخدمة لربط المعاملة بها
- **title** (مطلوب) - اسم قصير للمعاملة
- **description** (اخ��ياري) - وصف تفصيلي
- **invoice_id** (اختياري) - معرف الفاتورة إذا تم إصدار فاتورة بالفعل (عادةً null)
- **wholesale_cost** (اختياري) - تكلفتك
- **retail_cost** (مطلوب) - التكلفة التي يواجهها العميل
- **site_id** (اختياري) - معرف الموقع لربط المعاملة به
- **tax_percentage** (اختياري) - نسبة ضريبة

حالات الاستخدام:

- رسوم الإعداد أثناء توفير الخدمة
- رسوم التركيب
- الائتمانات أو التعديلات اليدوية
- رسوم المعدات
- الرسوم الإدارية

ملاحظة: إنشاء المعاملات مباشرة لا يعالج الدفع - بل ينشئ سجل فواتير فقط. ستظهر المعاملة كغير مفوترة ويمكن تضمينها في الفواتير المستقبلية.

حساب الرسوم النسبية

يسمح الشحن النسبي لك بشحن العملاء بشكل متناسب بناءً على فترات الفوترة الجزئية. هذا شائع عندما:

- يقوم العميل بالتسجيل في منتصف الشهر
- يتم ترقية / تخفيض الخدمة في منتصف الدورة
- يتم حساب الفوترة بناءً ��لى الأيام المستخدمة

صيغة حساب النسبة

```
pro_rata_charge = (monthly_cost × days_remaining) / days_in_month
```

مثال على التنفيذ

إليك كيفية حساب رسوم نسبية في Playbook:

```yaml
# حساب رسوم نسبية لشهر جزئي
# إذا قام العميل بالتسجيل في اليوم الخامس عشر والفوترة في اليوم الأول،
# يتم الشحن بشكل متناسب للأيام المتبقية

- name: Get current day of month
  command: "date +%d"
  register: current_day

- name: Get total days in current month
  command: "date -d 'last day of this month' +%d"
  register: days_in_month

- name: Get last day of month
  command: "date -d 'last day of this month' +%Y-%m-%d"
  register: last_day_of_month

- name: Calculate days remaining in month
  set_fact:
    days_remaining: "{{ (days_in_month.stdout | int) -
(current_day.stdout | int) + 1 }}"

- name: Calculate pro-rata cost
  set_fact:
    pro_rata_cost: "{{ ((monthly_cost | float) * (days_remaining |
float) / (days_in_month.stdout | float)) | round(2) }}"

- name: Display calculation details
  debug:
    msg:
      - "Monthly cost: ${{ monthly_cost }}"
      - "Days in month: {{ days_in_month.stdout }}"
      - "Days remaining: {{ days_remaining }}"
      - "Pro-rata charge: ${{ pro_rata_cost }}"

# استخدم pro_rata_cost في التفويض أو إنشاء المعاملة
- name: "Authorize pro-rata payment"
  uri:
    url: "http://localhost:5000/crm/payments/authorize/hold"
    method: POST
    headers:
      Content-Type: "application/json"
      Authorization: "Bearer {{ access_token }}"
    body_format: json
```

```
    body:
      {
        "customer_id": "{{ customer_id | int }}",
        "amount": "{{ pro_rata_cost | float }}",
        "currency": "{{ crm_config.currency | default('AUD') }}",
        "payment_method_id": "{{ payment_method_id | int }}",
        "metadata": {
          "title": "{{ package_name }} (Pro-rata {{ days_remaining
}} days)",
          "description": "Pro-rated charge for {{ days_remaining
}}/{{ days_in_month.stdout }} days of {{ package_name }}",
          "service_id": "{{ service_id | int }}",
          "invoice": true
        }
      }
```

النسبة من تاريخ بدء مخصص

إذا كنت بحاجة إلى حساب النسبة من تاريخ بدء محدد إلى تاريخ الفوترة:

```yaml
- name: Set custom start date and billing date
  set_fact:
    service_start_date: "2024-01-15"
    next_billing_date: "2024-02-01"

- name: Calculate days between dates
  shell: |
    echo $(( ( $(date -d "{{ next_billing_date }}" +%s) - $(date -d "{{ service_start_date }}" +%s) ) / 86400 ))
  register: days_until_billing

- name: Get days in billing period (usually 30)
  set_fact:
    billing_period_days: 30

- name: Calculate pro-rata cost
  set_fact:
    pro_rata_cost: "{{ ((monthly_cost | float) * (days_until_billing.stdout | float) / (billing_period_days | float)) | round(2) }}"

- name: Display calculation
  debug:
    msg:
      - "Start date: {{ service_start_date }}"
      - "Next billing: {{ next_billing_date }}"
      - "Days until billing: {{ days_until_billing.stdout }}"
      - "Pro-rata charge: ${{ pro_rata_cost }}"
```

سيناريوهات النسبة

السيناريو 1: التسجيل في منتصف الشهر

- يقوم العميل بالتسجيل في 15 يناير
- التكلفة الشهرية: 60.00 دولار
- الأيام في يناير: 31
- الأيام المتبقية: 17 (من 15 إلى 31 شامل)
- الرسوم النسبية: `32.90$ = 31 ÷ 17 × 60.00$`

السيناريو 2: ترقية الخدمة

- يقوم العميل بالترقية في اليوم العاشر من دورة الفوترة
- الخطة القديمة: 30 دولارًا شهريًا
- الخطة الج��يدة: 50 دولارًا شهريًا
- الأيام في الدورة: 30
- الأيام المتبقية: 21
- الفرق: 20 دولارًا شهريًا
- الرسوم النسبية: `14.00$ = 30 ÷ 21 × 20.00$`

أفضل الممارسات

1. استخدم دائمًا نمط الكتلة / الإنقاذ

قم بتغليف التقاط الدفع والتوفير في كتلة / إنقاذ لضمان التراجع:

```
- block:
    # مهام التوفير
    - name: Provision service
      uri: ...

    # التقاط الدفع فقط بعد النجاح
    - name: Capture payment
      uri: ...

  rescue:
    # إطلاق التفويض إذا فشل أي شيء
    - name: Release payment authorization
      uri: ...

    - name: Fail playbook
      fail:
        msg: "فشل التوفير، لم يتم شحن العميل"
```

2. تحقق من جميع استجابات API

تأكد دائمًا من أن العمليات الحرجة قد نجحت:

```
- name: Authorize payment
  uri: ...
  register: api_response_authorization

- name: Assert authorization was successful
  assert:
    that:
      - api_response_authorization.status == 200
      - api_response_authorization.json.success == true
    fail_msg: "فشل تفويض الدفع: {{ api_response_authorization.json
}}"
```

3. احفظ معرف التفويض

احفظ دائمًا authorization_id الإلغاء / لاستخدامه في الالتقاط:

```
- name: Store authorization_id for capture/release
  set_fact:
    authorization_id: "{{
api_response_authorization.json.data.authorization_id }}"
```

4. استخدم بيانات التعريف بفعالية

قم بتضمين بيانات تعريف شاملة في طلبات التفويض:

```
metadata:
  description: "وصف واضح لما يتم شحن العميل من أجله"
  service_id: "{{ service_id | int }}"
  product_id: "{{ product_id }}"
  user_id: "{{ initiating_user | int }}"
  title: "عنوان قصير للمعاملة"
  wholesale_cost: "{{ wholesale_cost | float }}"
  invoice: true  # إنشاء المعاملة تلقائيًا عند الالتقاط
  send_email: true  # إرسال إشعار للعميل
```

5. قم بتقريب قيم العملة

قم دائمًا بتقريب حسابات العملة إلى مكانين عشريين:

```
- name: Calculate cost with rounding
  set_fact:
    final_cost: "{{ (base_cost | float * multiplier | float) |
round(2) }}"
```

6. التعامل مع طرق الدفع المفقودة

تحقق مما إذا كان لدى العميل طريقة دفع افتراضية قبل محاولة التفويض:

```
- name: Get default payment method
  set_fact:
    payment_method_id: "{{ api_response_payment_methods.json |
json_query(query) }}"
  vars:
    query: "data[?is_default==`true`].payment_method_id | [0]"

- name: Verify payment method exists
  assert:
    that:
      - payment_method_id is defined
      - payment_method_id != ""
      - payment_method_id != None
    fail_msg: "لم يتم العثور على طريقة دفع افتراضية للعميل {{
customer_id }}"
```

الأنماط الشائعة

النمط 1: توفير خدمة مدفوعة

بالنسبة للخدمات التي تتطلب دفعًا فوريًا:

1. الحصول على طريقة الدفع الخاصة بالعميل
2. تفويض الدفع

3. توفير الخدمة في OCS / CGRateS
4. إنشاء سجل الخدمة في CRM
5. التقاط الدفع
6. في حالة الفشل: إطلاق التفويض والتراجع

راجع `play_topup_charge_then_action.yaml` للحصول على مثال كامل.

النمط 2: خدمة مجانية مع رسوم إعداد

بالنسبة للخدمات التي تكون مجانية ولكن لديها تكلفة إعداد لمرة واحدة:

1. توفير الخدمة
2. إنشاء سجل الخدمة
3. إضافة معاملة رسوم الإعداد مباشرة (دون معالجة الدفع)
4. تظهر رسوم الإعداد في الفاتورة التالية

راجع `play_simple_service.yaml` للحصول على مثال كامل في الأسطر 202-232.

النمط 3: إضافة / ملحق مجاني

بالنسبة للإضافات المجانية التي لا تتطلب دفعًا:

1. الحصول على معلومات الخدمة
2. تنفيذ إجراء CGRateS
3. تحديث تواريخ الخدمة
4. لا حاجة للدفع أو إنشاء المعاملة

النمط 4: رسوم متكررة عبر ActionPlan

بالنسبة للرسوم التلقائية المتكررة:

1. إنشاء إجراء مع `*http_post` إلى نقطة النهاية الخاصة بالتوفير
2. إنشاء ActionPlan مع توقيت `*monthly`
3. تعيين ActionPlan إلى الحساب
4. ستقوم CGRateS تلقائيًا باستدعاء نقطة النهاية شهريًا
5. تتعامل Playbook الخاصة بنقطة النهاية مع معالجة الدفع

استكشاف الأخطاء وإصلاحها

فشل التفويض

العرض: ترجع نقطة نهاية التفويض خطأ

الأسباب الشائعة:

- ط��يقة الدفع غير موجودة أو غير صالحة
- أموال غير كافية
- انتهاء صلاحية طريقة الدفع
- عدم تطابق معرف العميل

الحل: تحقق من حالة طريقة الدفع ورصيد العميل.

فشل الالتقاط بعد التوفير الناجح

العرض: تم توفير الخدمة ولكن فشل التقاط الدفع

المشكلة: هذه حالة فشل حرجة - الخدمة نشطة ولكن لم يتم شحن العميل

الحل:

- قد يكون التفويض قد انتهى (عادةً 7 أيام)
- تحقق من أن التفويض لا يزال صالحًا قبل محاولة الالتقاط
- نفذ مراقبة لفشل الالتقاط
- قد تكون هناك حاجة لتدخل يدوي

لم يتم إنشاء معاملة بعد الالتقاط

العرض: تم التقاط الدفع ولكن لا توجد معاملة في الفوترة

السبب: لم يتم تعيين `invoice: true` في بيانات التعريف الخاصة بالتفويض

الحل: إما:

- تعيين `invoice: true` في بيانات التعريف الخاصة بالتفويض، أو

- إنشاء معاملة يدويًا بعد الالتقاط الناجح

حساب النسبة غير صحيح

العرض: الرسوم النسبية لا تتطابق مع القيم المتوقعة

الأسباب الشائعة:

- أخطاء في العد (شاملة / غير شاملة تواريخ البدء / الانتهاء)
- الشهر الخاطئ المستخدم لحساب الأيام
- أخطاء تقريب

الحل:

- استخدم نطاقات تواريخ شاملة (شاملة كل من اليوم الأول والأخير)
- قم دائمًا بالتقريب إلى مكانين عشريين
- اختبر الحسابات بقيم معروفة
- وثق التواريخ التي يتم تضمينها في الحسابات

المبالغ المستردة ومعالجة الأخطاء

خيارات المبالغ المستردة

يدعم نظام الدفع نوعين من المبالغ المستردة:

1. **استرداد إلى مصدر الدفع** / PayPal - إرجاع الأموال إلى البطاقة الأصلية

```
- name: Refund payment to customer's card
  uri:
    url: "http://localhost:5000/crm/payments/refund"
    method: POST
    headers:
      Content-Type: "application/json"
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "transaction_id": "{{ vendor_transaction_id }}",
        "vendor": "stripe",
        "amount": "{{ refund_amount | float }}",
        "reason": "customer_request"
      }
```

2. ائتمان إلى المحفظة - رصيد فوري للمشتريات المستقبلية (يتم التعامل معه تلقائيًا في سيناريوهات الخطأ)

بالنسبة لفشل التوفير، يقوم النظام تلقائيًا بإضافة رصيد إلى المحفظة بدلاً من استرداد البطاقة لـ:

- تجنب رسوم الاسترداد
- توفير توفر فوري لإعادة المحاولة
- تحسين تجربة العميل

راجع خيارات المبالغ المستردة للحصول على تفاصيل كاملة.

دعم البائع

نظام الدفع هو **مستقل عن البائع** ويدعم حاليًا:

- Stripe (البطاقات، ACH)
- PayPal (البطاقات، حسابات PayPal)

يمكن إضافة بائعين جدد (Square، Adyen، Braintree، إلخ) دون تغيير Playbooks.

انظر أيضًا:

- الهندسة المعمارية المودولية - كيف تعمل تجريد البائع

- إعداد تكوين البائع - Stripe / PayPal ومفاتيح API

الوثائق ذات الصلة

أدلة محددة لـ Playbook

- `concepts_ansible.md` - أنماط Playbook العامة والبنية
- `concepts_provisioning.md` - نظرة عامة على نظام التوفير

وثائق نظام المدفوعات

- **دليل نظام المدفوعات API** - مرجع API كامل، توجيه المحفظة، المبالغ المستردة
  - واجهات برمجة التطبيقات لطريقة الدفع
  - واجهات برمجة التطبيقات لتدفق الدفع
  - توجيه المحفظة أولاً
  - خيارات المبالغ المستردة
  - الهندسة المعمارية المودولية
- `payments_transaction.md` - إدارة المعاملات والرسوم اليدوية
- `payments_process.md` - معالجة المدفوعات والفواتير
- `basics_payment.md` - طرق الدفع وفوترة العملاء

التحكم في الوصول القائم على القواعد

الأدوار والأذونات والمستخدمون في OmniCRM

تستخدم OmniCRM **التحكم في الوصول القائم على الأدوار (RBAC)**: يتم تعيين الأشخاص (المستخدمون) لدور واحد أو أكثر، وكل دور هو مجموعة من الأذونات. الأذونات هي أصغر وحدة وصول (مثل، *view_customer*، *create_inventory*). الوصول الفعلي للمستخدم هو **اتحاد** الأذونات من جميع الأدوار المعينة.

الغرض

يمكن أن يوفر RBAC:

1. **حماية البيانات**: يرى المستخدمون فقط ما يُسمح لهم برؤيته والقيام به.
2. **تناسب تشغيلي**: تعكس الأدوار وظائف العمل (مدير، دعم، مالية، إدارة العملاء).
3. **إدارة بسيطة**: منح الوصول عن طريق تعيين الأدوار. تجنب الإدارة الدقيقة لكل مستخدم.
4. **عزل المستأجرين**: تحدد أذونات "عرض الخاص ..." الرؤ��ة لبيانات العملاء/المستأجرين الخاصة بالمستخدم.

كيف تتناسب المستخدمون والأدوار والأذونات معًا

- **المستخدمون**: أشخاص حقيقيون يقومون بتسجيل الدخول إلى OmniCRM.
- **الأذونات**: قدرات ذرية (مثل، *view_customer*، *delete_product*).
- **الأدوار**: مجموعات مسماة من الأذونات (مثل، *Admin*، *Support*، *Finance*).
- **التعيين**: يحصل المستخدمون على دور واحد أو أكثر. تتجمع الأذونات.

تحدد التفويض (الأدوار/ .(مدرج في القائمة البيضاء IP أو، مفتاح API، JWT) يثبت المصادقة *من أنت* الأذونات) *ما يمكنك القيام به*.

إدارة المستخدمين

يسمح نظام إدارة المستخدمين في OmniCRM للمسؤولين بإنشاء وإدارة مستخدمي الموظفين (المسؤولين، وكلاء خدمة العملاء)، وعرض وتعديل أدوار المستخدمين، وإعادة تعيين كلمات المرور، وإدارة المصادقة الثنائية، والتحكم في وصول المست��دمين.

أنواع المستخدمين

مستخدمو العملاء - يتم إنشاؤهم عبر التسجيل الذاتي أو بواسطة المسؤولين. يتم تعيينهم تلقائيًا لدور "العميل". يمكن لهؤلاء المستخدمين الوصول إلى بوابة الرعاية الذاتية لإدارة خدماتهم، وعرض الاستخدام، ودفع الفواتير، إلخ.

مستخدمو الموظفين - يتم إنشاؤهم بواسطة المسؤولين الذين لديهم الأذونات المناسبة. يمكن تعيينهم لأدوار مثل المدير، الدعم، المالية، إلخ. يمكن لهؤلاء المستخدمين الوصول إلى واجهة CRM لإدارة العملاء، وتوفير الخدمات، والتعامل مع الفواتير، إلخ.

المستخدمون الإداريون - مستخدمون لديهم إذن `admin`. لديهم وصول كامل إلى النظام بما في ذلك إدارة المستخدمين، وإدارة الأدوار، وجميع نقاط النهاية المحمية.

يتم إنشاء المستخدم الإداري الأول بواسطة فريق Omnitouch عند نشر النظام.

إضافة مستخدمين جدد (المسؤولون وCSAs)

يمكن للمسؤولين إنشاء مستخدمين جدد من الموظفين من خلال واجهة الويب أو API.

عبر واجهة الويب

1. **انتقل إلى المستخدمين والأدوار** - الوصول إلى واجهة إدارة المستخدمين من قائمة الإدارة
2. **انقر على "إضافة مستخدم"** - يفتح نموذج إنشاء المستخدم

3. **املأ تفاصيل المستخدم:**
    - **اسم المستخدم** - اسم مستخدم فريد لتسجيل الدخول (مطلوب)
    - **البريد الإلكتروني** - عنوان البريد الإلكتروني للمستخدم (مطلوب، يجب أن يكون فريدًا)
    - **كلمة المرور** - كلمة مرور مؤقتة (مطلوبة، يجب على المستخدم تغييرها عند تسجيل الدخول لأول مرة)
    - **الاسم الأول** - الاسم الأول للمستخدم (مطلوب)
    - **الاسم الأوسط** - الاسم الأوسط للمستخدم (اختياري)
    - **اسم العائلة** - اسم العائلة للمستخدم (مطلوب)
    - **رقم الهاتف** - رقم الهاتف للتواصل (اختياري)
    - **الدور** - اختر دورًا أو أكثر للتعيين (مطلوب)
    - **جهة اتصال العميل** - ربط المستخدم بسجل جهة اتصال عميل (للمستخدمين العملاء)
4. **انقر على "إنشاء مستخدم"** - يتم إنشاء المستخدم ويمكنه تسجيل الدخول على الفور باستخدام بيانات الاعتماد المقدمة
5. **يتلقى المستخدم إشعارًا** - إرسال بريد إلكتروني ترحيبي مع تعليمات تسجيل الدخول (اختياري)

أفضل الممارسات:

- استخدم كلمة مرور مؤقتة مثل `TempP@ssw0rd!` واطلب من المستخدم تغييرها عند تسجيل الدخول لأول مرة
- تعيين الأدوار المناسبة بناءً على وظيفة العمل (انظر تصميمات الأدوار النموذجية أدناه)

- تمكين 2FA والدعم لجميع الموظفين الإداريين
- ربط مستخدمي العملاء بسجل جهة اتصال العميل الخاص بهم لتحديد البيانات بشكل صحيح

عبر API

إنشاء مستخدم برمجيًا:

نقطة النهاية: `POST /auth/users`

الإذن المطلوب: `admin`

جسم الطلب:

```
{
  "username": "john.smith",
  "email": "john.smith@company.com",
  "password": "TempP@ssw0rd!",
  "first_name": "John",
  "middle_name": "D",
  "last_name": "Smith",
  "phone_number": "+61412345678",
  "role": "Support"
}
```

الاستجابة:

```
{
  "id": 123,
  "username": "john.smith",
  "email": "john.smith@company.com",
  "first_name": "John",
  "last_name": "Smith",
  "roles": ["Support"],
  "created": "2025-01-04T10:30:00Z"
}
```

تعيين أدوار متعددة:

يمكن أن يكون لدى المستخدمين أدوار متعددة. الأذونات تضاف (اتحاد جميع أذونات الأدوار المعينة).

لتعيين أدوار متعددة، قم بتضمينها في الطلب:

```
{
  "username": "jane.doe",
  "email": "jane.doe@company.com",
  "password": "TempP@ssw0rd!",
  "first_name": "Jane",
  "last_name": "Doe",
  "role": "Support,Finance"
}
```

أو استخدم نقطة نهاية تعيين الدور بعد إنشاء المستخدم:

```
POST /auth/roles/{role_id}/users/{user_id}
```

عرض والبحث عن المستخدمين

قائمة جميع المستخدمين (المسؤول):

`GET /auth/users`

يُرجع قائمة مقسمة من جميع المستخدمين مع أدوارهم ومعلومات أساسية.

البحث عن المستخدمين:

`GET /auth/users/search?search={query}&filters={"role":["Support"]}&page=1&per_page=50`

تصفية حسب:

- اسم الدور
- نطاق البريد الإلكتروني
- حالة نشط/محذوف
- حالة تمكين 2FA
- تاريخ آخر تسجيل دخول

الحصول على مستخدم محدد:

`GET /auth/users/{user_id}`

يُرجع تفاصيل المستخدم الكاملة بما في ذلك:

- المعلومات الشخصية
- الأدوار المعينة والأذونات الفعالة
- حالة 2FA
- آخر تسجيل دخول ومعلومات الجلسة
- جهة اتصال العميل المرتبطة (إذا كانت قابلة للتطبيق)

إنشاء وإدارة الأدوار

الأدوار هي مجموعات من الأذونات يمكن تعيينها للمستخدمين. بدلاً من تعيين الأذونات بشكل فردي لكل مستخدم، يمكنك إنشاء أدوار تجمع الأذونات ذات الصلة وتعيين تلك الأدوار للمستخدمين.

إنشاء دور جديد

عبر واجهة الويب:

1. انتقل إلى **المستخدمون والأدوار** ← علامة التبويب **الأدوار**
2. انقر على **"إنشاء دور"**
3. أدخل تفاصيل الدور:
   - **الاسم** - اسم قصير ووصف (مثل، "Tier2_Support")
   - **الوصف** - شرح الغرض من الدور ومسؤولياته
4. انقر على **"إنشاء"**
5. يتم إنشاء الدور بدون أذونات؛ أضف الأذونات في الخطوة التالية

عبر API:

نقطة النهاية: `POST /auth/roles`

الإذن المطلوب: `admin`

الطلب:

```
{
  "name": "Tier2_Support",
  "description": "فريق الدعم من المستوى 2 مع وصول مرتفع للتوفير"
}
```

الاستجابة:

```
{
  "id": 45,
  "name": "Tier2_Support",
  "description": "فريق الدعم من المستوى 2 مع وصول مرتفع للتوفير",
  "permissions": [],
  "users": []
}
```

إضافة أذونات إلى دور

بمجرد إنشاء الدور، قم بتعيين الأذونات لتحديد ما يمكن أن يفعله المستخدمون الذين لديهم هذا الدور.

عبر واجهة الويب:

1. انتقل إلى **مستخدمون والأدوار**ا�ا → علامة التبويب **الأدوار**
2. انقر على اسم الدور لعرض التفاصيل
3. في قسم **الأذونات**، انقر على **"إضافة إذن"**
4. اختر إذنًا أو أكثر من القائمة
5. انقر على **"إضافة"** - يتم تعيين الأذونات على الفور

عبر API:

نقطة النهاية: `POST /auth/roles/{role_id}/permissions`

الطلب:

```
{
  "permission_id": 123
}
```

أو أضف أذونات متعددة:

```
{
  "permission_ids": [123, 124, 125]
}
```

مثال: إنشاء دور "أخصائي التوفير"

يمكن لهذا الدور عرض العملاء، وإدارة الخدمات، والتوفير:

1. إنشاء الدور:

```
POST /auth/roles
{
  "name": "Provisioning_Specialist",
  "description": "يمكنه توفير الخدمات وإدارة خدمات العملاء"
}
```

2. إضافة الأذونات:

```
POST /auth/roles/45/permissions
{
  "permission_ids": [
    1,   # view_customer
    20,  # view_customer_service
    21,  # create_customer_service
    22,  # update_customer_service
    30,  # view_provision
    31,  # create_provision
    40,  # view_inventory
    50,  # view_product
  ]
}
```

إزالة الأذونات من دور

عبر واجهة الويب:

1. انتقل إلى تفاصيل الدور
2. أو زر **"إزالة"** بجوار الإذن **"X"** في قائمة **الأذونات**، انقر على
3. أكد الإزالة

عبر API:

نقطة النهاية: `DELETE /auth/roles/{role_id}/permissions/{permission_id}`

مثال:

```
DELETE /auth/roles/45/permissions/31
```

هذا يزيل إذن `create_provision` من الدور.

تحرير تفاصيل الدور

تحديث اسم الدور أو الوصف:

عبر واجهة الويب:

1. انتقل إلى **المستخدمون والأدوار** ← علامة التبويب **الأدوار**
2. انقر على الدور لتحريره
3. تعديل اسم الدور أو الوصف
4. انقر على **"حفظ"**

عبر API:

نقطة النهاية: `PUT /auth/roles/{role_id}`

```
{
  "name": "Senior_Support",
  "description": "فريق الدعم الكبير مع وصول كامل للعملاء"
}
```

حذف دور

تحذير: يؤدي حذف دور إلى إزالته من جميع المستخدمين المعينين. تأكد من أن لدى المستخدمين أدوار بديلة أو سيفقدون الوصول.

عبر API:

`DELETE /auth/roles/{role_id}`

أفضل ممارسة: بدلاً من الحذف، اعتبر أرشفة أو إعادة تسمية الأدوار التي لم تعد مطلوبة.

تعيين الأدوار للمستخدمين

أثناء إنشاء المستخدم:

قم بتضمين الدور في طلب إنشاء المستخدم (انظر "إضافة مستخدمين جدد" أعلاه).

للمستخدمين الحاليين:

عبر واجهة الويب:

1. انتقل إلى **المستخدمون والأدوار** ← علامة التبويب **المستخدمون**
2. انقر على المستخدم لتحريره
3. في قسم **الأدوار**، حدد/ألغِ تحديد الأدوار
4. انقر على **"حفظ"**

عبر API:

تحديث أدوار المستخدم:

نقطة النهاية: `PUT /auth/users/{user_id}`

```
{
  "role": "Support,Finance"
}
```

أو تعيين دور واحد لمستخدم عبر نقطة نهاية الدور:

نقطة النهاية: `POST /auth/roles/{role_id}/users/{user_id}`

عرض تعيينات الأدوار

جميع المستخدمين في دور:

`GET /auth/roles/{role_id}/users`

يُرجع قائمة بجميع المستخدمين المعينين لذلك الدور.

جميع الأدوار لمستخدم:

`GET /auth/users/{user_id}`

تتضمن الاستجابة مصفوفة `roles` مع جميع الأدوار المعينة.

إدارة كلمات مرور المستخدمين

توفر OmniCRM طرقًا متعددة لإدارة كلمات المرور اعتمادًا على السياق.

إعادة تعيين كلمة المرور بواسطة الخدمة الذاتية للمستخدم

يمكن للمستخدمين الذين نسوا كلمة مرورهم إعادة تعيينها بأنفسهم عبر صفحة تسجيل الدخول:

1. **انقر على "نسيت كلمة المرور"** في صفحة تسجيل الدخول
2. **أدخل عنوان البريد الإلكتروني** - يرسل النظام بريدًا إلكترونيًا لإعادة تعيين كلمة المرور
3. **تحقق من البريد الإلكتروني** - يحتوي البريد الإلكتروني على رابط إعادة تعيين آمن (صالح لمدة ساعة واحدة) مع رمز
4. **انقر على الرابط** - يفتح نموذج إعادة تعيين كلمة المرور
5. **أدخل كلمة المرور الجديدة** - يجب أن تلبي متطلبات تعقيد كلمة المرور:
   - الحد الأدنى 8 أحرف
   - على الأقل حرف كبير واحد
   - على الأقل حرف صغير واحد

- على الأقل رقم واحد
- على الأقل رمز خاص واحد

6. **إرسال** - يتم تحديث كلمة المرور على الفور؛ يمكن للمستخدم تسجيل الدخول باستخدام كلمة المرور الجديدة

تدفق API:

1. **طلب إعادة تعيين:**

   نقطة النهاية: `POST /auth/forgot_password`

   ```
   {
     "email": "user@example.com"
   }
   ```

   يقوم النظام بإنشاء رمز إعادة تعيين ويرسل البريد الإلكتروني.

2. **إعادة تعيين باستخدام الرمز:**

   نقطة النهاية: `POST /auth/reset_password`

   ```
   {
     "token": "abc123...",
     "new_password": "NewSecureP@ssw0rd!"
   }
   ```

��عادة تعيين كلمة المرور بواسطة المسؤول

يمكن للمسؤولين إعادة تعيين كلمة مرور المستخدم دون الحاجة إلى التحقق من البريد الإلكتروني. هذا يحدد كلمة مرور مؤقتة يجب على المستخدم تغييرها عند تسجيل الدخول التالي.

عبر واجهة الويب:

1. انتقل إلى **المستخدمون والأدوار** ← **المستخدمون**
2. ابحث عن المستخدم وانقر على زر **"إعادة تعيين كلمة المرور"**
3. أدخل كلمة مرور مؤقتة
4. انقر على **"إعادة تعيين"**

5. أبلغ المستخدم بكلمة المرور المؤقتة الخاصة به (عبر قناة آمنة)
6. يجب على المستخدم تغيير كلمة المرور عند تسجيل الدخول التالي

عبر API:

نقطة النهاية: `POST /auth/users/{user_id}/admin_reset_password`

الإذن المطلوب: `admin`

الطلب:

```
{
  "new_password": "TempP@ssw0rd!",
  "force_change": true
}
```

المعلمات:

- `new_password` - كلمة المرور المؤقتة التي سيتم تعيينها
- `force_change` (اختياري) - إذا كانت صحي��ة، يجب على المستخدم تغيير كلمة المرور عند تسجيل الدخول التالي

تغيير المستخدم لكلمة المرور الخاصة به

يمكن للمستخدمين المعتمدين تغيير كلمة المرور الخاصة بهم من ملفهم الشخصي:

نقطة النهاية: `POST /auth/change_password`

الطلب:

```
{
  "current_password": "OldP@ssw0rd!",
  "new_password": "NewSecureP@ssw0rd!"
}
```

يتحقق النظام من كلمة المرور الحالية قبل السماح بالتغيير.

أمان كلمة المرور

- يتم تشفير كلمات المرور باستخدام bcrypt (أمان werkzeug)
- لا يتم تخزينها بنص عادي
- تنتهي صلاحية رموز إعادة التعيين بعد ساعة واحدة
- يمكن أن تؤدي محاولات تسجيل الدخول الفاشلة إلى قفل الحساب (قابل للتكوين)
- يمنع تتبع تاريخ كلمة المرور إعادة الاستخدام (قابل للتكوين)
- يتم فرض متطلبات التعقيد

إدارة المصادقة الثنائية (2FA)

تدعم OmniCRM المصادقة الثنائية المعتمدة على TOTP. يمكن للمسؤولين تمكين 2FA وتعطيل وإعادة تعيين 2FA للمستخدمين.

تمكين 2FA لمستخدم

عبر واجهة الويب:

1. انتقل إلى **المستخدمون والأدوار** ← **المستخدمون**

2. انقر على المستخدم لعرض التفاصيل
3. في قسم **الأمان**، انقر على **"تمكين 2FA"**
4. يقوم النظام بإنشاء:
    - سر TOTP (يتم عرض رمز QR)
    - 10 رموز احتياطية (للاستخدام لمرة واحدة)
5. يقوم المستخدم بمسح رمز QR باستخدام تطبيق المصادقة (Google Authenticator، Authy، إلخ.)
6. يدخل المستخدم رمز التحقق من التطبيق لتأكيد الإعداد
7. يحتفظ المستخدم بالرموز الاحتياطية في مكان آمن
8. تم تمكين 2FA الآن؛ مطلوب لجميع تسجيلات الدخول المستقبلية

عبر API:

1. **إنشاء سر TOTP:**

نقطة النهاية: `POST /2fa/enable/user/{user_id}`

الاستجابة:

```
{
  "totp_secret": "JBSWY3DPEHPK3PXP",
  "qr_code_url": "otpauth://totp/OmniCRM:user@example.com?
secret=JBSWY3DPEHPK3PXP&issuer=OmniCRM",
  "backup_codes": [
    "12345678",
    "23456789",
    "34567890",
    ...
  ]
}
```

2. **تحقق من الإعداد:**

نقطة النهاية: `POST /2fa/verify-setup/user/{user_id}`

```
{
  "code": "123456"
}
```

تعيد `{"verified": true}` عند النجاح.

تدفق تسجيل الدخول 2FA

بمجرد تمكين 2FA، يتغير عملية تسجيل الدخول:

1. يدخل المستخدم اسم المستخدم وكلمة المرور
2. يتحقق النظام من بيانات الاعتماد
3. إذا كانت صحيحة، يطلب رمز 2FA
4. يدخل المستخدم الرمز من تطبيق المصادقة أو الرمز الاحتياطي
5. يتحقق النظام من الرمز
6. عند النجاح، يتم تسجيل دخول المستخدم

الرموز الاحتياطية:

- تم إنشاء 10 رموز أثناء إعداد 2FA
- للاستخدام لمرة واحدة فقط (تستهلك بعد الاستخدام)
- تستخدم إذا كان تطبيق المص��دقة غير متاح
- يمكن أن يتم إعادة إنشائها بواسطة المستخدم أو المسؤول

التحقق من رمز 2FA

نقطة النهاية: `POST /2fa/verify/user/{user_id}`

```
{
  "code": "123456"
}
```

يقبل كلا من:

- **رمز TOTP** (6 أرقام من تطبيق المصادقة)
- **رمز احتياطي** (8 أرقام من قائمة الرموز الاحتياطية)

إعادة إنشاء الرموز الاحتياطية

إذا استنفد المستخدم الرموز الاحتياطية أو فقدها، قم بإنشاء رموز جديدة:

عبر واجهة الويب:

1. انتقل إلى تفاصيل المستخدم
2. انقر على **"إعادة إنشاء الرموز الاحتياطية"**
3. عرض أو إرسال الرموز الجديدة إلى المستخدم
4. تصبح الرموز القديمة غير صالحة

عبر API:

نقطة النهاية: `POST /2fa/backup-codes/regenerate/user/{user_id}`

الاستجابة:

```
{
  "backup_codes": [
    "98765432",
    "87654321",
    "76543210",
    ...
  ]
}
```

إعادة تعيين 2FA بواسطة المسؤول

إذا فقد المستخدم الوصول إلى تطبيق المصادقة الخاص به وجميع الرموز الاحتياطية، يمكن للمسؤول تعطيل وإعادة تمكين 2FA.

عبر واجهة الويب:

1. انتقل إلى **المستخدمون والأدوار** → **المستخدمون**
2. انقر على المستخدم
3. انقر على زر **"إعادة تعيين 2FA"**
4. أكد إعادة التعيين
5. يتم تعطيل 2FA فقط؛ يمكن للمستخدم تسجيل الدخول باستخدام كلمة المرور

6. إرشاد المستخدم لإعداد 2FA مرة أخرى مع السر الجديد

عبر API:

نقطة النهاية: `POST /2fa/admin/disable/user/{user_id}`

الإذن المطلوب: `admin`

هذا يعطل 2FA تمامًا للمستخدم:

- يتم مسح سر TOTP
- يتم مسح الرموز الاحتياطية
- يتم تعيين علامة `is_2fa_enabled` إلى false

يمكن للمستخدم بعد ذلك إعادة تمكين 2FA ورموز احتياطية جديد للحصول على سر.

إعادة تعيين 2FA بواسطة الخدمة الذاتية للمستخدم (جهاز جديد)

إذا حصل المستخدم على جهاز جديد ولكن لا يزال لديه وصول إلى الرم��ز الاحتياطية:

نقطة النهاية: `POST /2fa/reset-for-new-device/user/{user_id}`

```
{
  "backup_code": "12345678"
}
```

يتحقق النظام من الرمز الاحتياطي، ثم ينشئ سر TOTP جديد ورموز احتياطية. يمكن للمستخدم إعداد تطبيق المصادقة على الجهاز الجديد.

أفضل الممارسات لـ 2FA

- **تطلب 2FA لجميع الموظفين الإداريين والدعم**
- **تخزين الرموز الاحتياطية بشكل آمن** (مدير كلمات المرور أو ملاحظة آمنة)

- **إعادة إنشاء الرموز الاحتياطية** بعد استخدام عدة
- **استخدام تطبيقات المصادقة ذات السمعة الطيبة** (Google Authenticator، Authy، Microsoft Authenticator)
- **توثيق إجراءات إعادة تعيين 2FA** لموظفي الدعم
- **تدقيق استخدام 2FA** - مراقبة المستخدمين الذين لديهم 2FA مفعل

تحديث معلومات المستخدم

يمكن للمسؤولين تحديث تفاصيل المستخدم في أي وقت.

عبر واجهة الويب:

1. انتقل إلى **المستخدمون والأدوار** → **المستخدمون**
2. انقر على المستخدم لتحريره
3. تعديل أي حقول قابلة للتحرير:
   - الاسم الأول، الاسم الأوسط، اسم العائلة
   - عنوان البريد الإلكتروني (يتطلب التحقق)
   - رقم الهاتف
   - الأدوار
   - ربط جهة اتصال العميل
4. انقر على **"حفظ"**

عبر API:

نقطة النهاية: `PUT /auth/users/{user_id}`

```
{
  "first_name": "Jane",
  "last_name": "Doe-Smith",
  "email": "jane.doesmith@newcompany.com",
  "phone_number": "+61498765432",
  "role": "Support,Finance"
}
```

تغييرات البريد الإلكتروني:

عند تغيير البريد الإلكتروني، يتم وضع البريد الإلكتروني الجديد في حالة الانتظار حتى يتم التحقق منه:

- يتم تخزين البريد الإلكتروني الجديد في حقل `pending_email`
- يتم إرسال بريد إلكتروني للتحقق إلى العنوان الجديد
- ينقر المستخدم على الرابط للتحقق
- يتم تحديث حقل `email` إلى القيمة الجديدة
- يتم تعيين علامة `email_verified` إلى true

حذف المستخدمين

الحذف الناعم ��لمستخدمين - يتم وضع علامة عليهم كمحذوفين ولكن لا OmniCRM تستخدم تتم إزالتهم من قاعدة البيانات. هذا يحافظ على سجلات التدقيق والبيانات التاريخية.

حذف مستخدم

عبر واجهة الويب:

1. انتقل إلى **المستخدمون والأدوار** → **المستخدمون**
2. ابحث عن المستخدم الذي تريد حذفه
3. انقر على زر **"حذف"**
4. أكد الحذف
5. يتم تسجيل خروج المستخدم على الفور ولا يمكنه تسجيل الدخول مرة أخرى

عبر API:

نقطة النهاية: `DELETE /auth/users/{user_id}`

الإذن المطلوب: `admin`

ماذا يحدث:

- يتم تعيين علامة `deleted` إلى `True`
- يتم تسجيل الطابع الزمني `deleted_at`
- لا يمكن للمستخدم تسجيل الدخول
- يتم إبطال جميع الجلسات النشطة

- لا يزال يظهر المستخدم في سجلات التدقيق والسجلات التاريخية
- يتم الحفاظ على البيانات المرتبطة (جهات اتصال العملاء، الأنشطة)

عرض المستخدمين المحذوفين

تصفية المستخدمين المحذوفين:

`GET /auth/users/search?filters={"deleted":[true]}`

استعادة مستخدم محذوف

إذا تم حذف مستخدم عن طريق الخطأ، يمكن للمسؤولين استعادته:

نقطة النهاية: `PUT /auth/users/{user_id}`

```
{
  "deleted": false
}
```

هذا clears the `deleted` flag ويسمح للمستخدم بتسجيل الدخول مرة أخرى.

ملاحظة: تظل كلمة مرور المستخدم دون تغيير، لذا يمكنهم استخدام كلمة المرور السابقة.

حذف مستخدم بشكل دائم

تحذير: هذا لا يمكن عكسه ويزيل جميع بيانات المستخدم من قاعدة البيانات.

لا يتم الكشف عنها عبر واجهة المستخدم. متاحة فقط عبر الوصول المباشر إلى قاعدة البيانات لأغراض الامتثال (مثل، طلبات حذف بيانات GDPR).

أفضل الممارسات لحذف المستخدمين

- **حذف ناعم بشكل افتراضي** - يحافظ على سجلات التدقيق
- **توثيق أسباب الحذف** - إضافة ملاحظة في سجل النشاط قبل الحذف
- **نقل الملكية** - إعادة تعيين تذاكر المستخدم المفتوحة، و��لمهام قبل الحذف
- **مراجعة الوصول** - التأكد من عدم اعتماد العمليات الحرجة على المستخدم
- **أرشفة البيانات** - تصدير تاريخ عمل المستخدم إذا لزم الأمر

- **إبلاغ الفرق ذات الصلة** - إبلاغ المديرين/الزملاء عن الحذف

كتالوج الأذونات

تتبع الأذونات عمومًا أنماط CRUD:

- *view_**: قراءة/تصفح
- *create_**: إنشاء/إضافة
- *update_**: تعديل/تغيير
- *delete_**: حذف/إزالة

تتضمن بعض الكيانات أيضًا متغيرات **"عرض الخاص ..."** التي تحد من الرؤية لبيانات العميل/المستأجر الخاصة بالمستخدم.

العالمية / الإدارية

- `admin`: وصول إداري كامل (إدارة المستخدمين، الأدوار، والأذونات؛ الوصول إلى جميع نقاط النهاية المحمية).
- `can_impersonate`: التصرف مؤقتًا كـ مستخدم آخر (مراقب؛ للدعم/استكشاف الأخطاء وإصلاحها).
- `grafana_access`: الوصول إلى لوحات تحليلات Grafana لإنشاء وعرض التقارير والتصورات المخصصة. انظر قسم تحليلات Grafana أدناه للحصول على

التفاصيل.

العملاء والسجلات ذات الصلة

- **العميل**
  - *view_customer، create_customer، update_customer، delete_customer*
  - **نطاق خاص:** *view own customer*
- **موقع العميل**
  - *view_customer_site، create_customer_site، update_customer_site، delete_customer_site*
  - **نطاق خاص:** *view own customer site*
- **جهة اتصال العميل**
  - *view_customer_contact، create_customer_contact، update_customer_contact، delete_customer_contact*
  - **نطاق خاص:** *view own customer contact*
- **سمة العميل** (انظر سمات العملاء)
  - *view_customer_attribute، create_customer_attribute، update_customer_attribute، delete_customer_attribute*
  - **نطاق خاص:** *view own customer attribute*
- **علامة العميل** (انظر علامات العملاء)
  - *view_customer_tag، create_customer_tag، update_customer_tag، delete_customer_tag*
  - **نطاق خاص:** *view own customer tag*
- **خدمة العميل**
  - *view_customer_service، create_customer_service، update_customer_service، delete_customer_service*
  - **نطاق خاص:** *view own customer service*
- **نشاط العميل**
  - *view_customer_activity، create_customer_activity، update_customer_activity، delete_customer_activity*
  - **نطاق خاص:** *view own customer activity*

الفواتير

- **بطاقة Stripe**
  - `view_customer_stripe_card`، `create_customer_stripe_card`، `update_customer_stripe_card`، `delete_customer_stripe_card`
  - **نطاق خاص:** *view own customer stripe card*
- **المعاملات**
  - `view_customer_transaction`، `create_customer_transaction`، `update_customer_transaction`، `delete_customer_transaction`
  - **نطاق خاص:** *view own customer transaction*
- **الفواتير**
  - `view_customer_invoice`، `create_customer_invoice`، `update_customer_invoice`، `delete_customer_invoice`
  - **نطاق خاص:** *view own customer invoice*

الاتصالات

- *view_communication، create_communication، update_communication، delete_communication*
- **نطاق خاص:** *view own communication*

المخزون والقوالب

- **المخزون**
  - `view_inventory`، `create_inventory`، `update_inventory`، `delete_inventory`
  - **نطاق خاص:** *view own inventory*
- **قالب المخزون**
  - `view_inventory_template`، `create_inventory_template`، `update_inventory_template`، `delete_inventory_template`

    - **نطاق خاص:** *view own inventory template*

المنتجات

- `view_product`، `create_product`، `update_product`، `delete_product`

بث الخلايا (CBC)

- `view_cbc_message`، `create_cbc_message`، `update_cbc_message`، `delete_cbc_message`

التوفير

- **التوفير**
    - `view_provision`، `create_provision`، `update_provision`، `delete_provision`
    - **نطاق خاص:** *view own provision*
- **حدث التوفير**
    - `view_provision_event`، `create_provision_event`، `update_provision_event`، `delete_provision_event`

"الوصول "عرض الخاص

تحدد أذونات "عرض الخاص ..." قراءات (وأحيانًا تعديلات، حيثما تم تنفيذها) للبيانات المرتبطة بعميل/مستأجر **خاص بالمستخدم**. على سبيل المثال، يمكن لدور *مدير العميل* إدارة جهات الاتصال، والمواقع، والفواتير، والخدمات الخاصة بـ *مستأجرهم*، ولكن لا يمكنهم رؤية مستأجرين آخرين.

تصميمات الأدوار النموذجية

الدور	الأذونات النموذجية	الملاحظات
مسؤول النظام	`admin`، اختياريًا `can_impersonate`؛ بالإضافة إلى CRUD واسع حسب الحاجة	تحكم كامل في المستخدمين/الأدوار/الأذونات
الدعم	`view_customer`، `view_customer_service`، `view_communication`، `view_provision`؛ تحديثات اختيارية	إضافة `can_impersonate` إذا كان مسموحًا
المالية	`view_customer_invoice`، `view_customer_transaction`، `view_product`؛ `create_customer_invoice` اختيارية	ثقيلة القراءة؛ كتابة محدودة
إدارة العملاء (المستأجر)	"عرض الخاص ..." عبر العملاء، المواقع، جهات الاتصال، الخدمات، المخزون، الفواتير، المعاملات، الاتصالات، التوفير	إدارة محددة للمستأجر
المدقق ذو الوصول فقط	`view_*` فقط واسع	لا إنشاء/تحديث/حذف

أمثلة على الأدوار والأذونات المضمنة (ملخص)

API إدارة الأدوار والأذونات عبر

تتطلب جميع نقاط النهاية إذن `admin`.

قائمة الأذونات

نقطة النهاية: `GET /auth/permissions`

إنشاء إذن (نادر)

نقطة النهاية: `POST /auth/permissions`

جسم الطلب:

```
{
    "name": "view_example",
    "description": "الوصول للقراءة فقط إلى كائنات المثال"
}
```

قائمة الأدوار

نقطة النهاية: `GET /auth/roles`

إنشاء دور

نقطة النهاية: `POST /auth/roles`

جسم الطلب:

```
{
"name": "Support",
"description": "فريق الدعم من المستوى 1"
}
```

إضافة إذن إل�ى دور

نقطة النهاية: `POST /auth/roles/{role_id}/permissions`

جسم الطلب:

```
{
  "permission_id": 123
}
```

إزالة إذن من دور

نقطة النهاية: `DELETE /auth/roles/{role_id}/permissions/{permission_id}`

تعيين الأدوار للمستخدمين

إنشاء مستخدم مع دور

نقطة النهاية: `POST /auth/users`

جسم الطلب:

```
{
  "username": "sara",
  "email": "sara@example.com",
  "password": "TempP@ssw0rd!",
  "first_name": "Sara",
  "last_name": "Ng",
  "phone_number": "+61...",
  "role": "Support"
}
```

تحديث دور مستخدم

نقطة النهاية: `PUT /auth/users/{user_id}`

جسم الطلب:

```
{
  "role": "Finance"
}
```

قائمة المستخدمين (للمسؤولين فقط)

نقطة النهاية: `GET /auth/users`

انتحال الشخصية (مراقب)

- **مطلوب:** `can_impersonate` أو `admin`

بدء انتحال الشخصية

نقطة النهاية: `POST /auth/impersonate`

جسم الطلب:

```
{ "user_id": 42 }
```

إيقاف انتحال الشخصية

نقطة النهاية: `POST /auth/stop_impersonation`

فضل الممارسات��

- **الأقل تفضيلًا أولاً.** ابدأ بأدوار الحد الأدنى؛ أضف الأذونات حسب الحاجة.
- **تفضل "عرض الخاص ...".** استخدم الأذونات المحددة للمستأجر للأدوار الموجهة للعملاء.
- **احتفظ بالأدوار مستقرة.** تحديث أذونات الدور عند تغيير الميزات. لا تعدل كل مستخدم.
- **تدقيق بانتظام.** مراجعة من لديه `admin` أو `can_impersonate`.

Grafana ولوحات المعلومات تحليلات

تتكامل OmniCRM مع Grafana لتوفير قدرات تحليلية وتصور قوية. يمكن للمستخدمين الذين لديهم إذن `grafana_access` الوصول إلى Grafana، وتقارير لإنشاء لوحات معلومات مخصصة، وتصويرات.

Grafana الوصول إلى

إذا تم منح حساب المستخدم الخاص بك إذن `grafana_access`، يمكنك الوصول إلى Grafana في متصفحك. سيتم مصادقتك تلقائيًا باستخدام بيانات اعتماد عن طريق الانتقال إلى `/grafana` الخاصة بك - لا حاجة لتسجيل دخول منفصل إلى OmniCRM Grafana.

من واجهة Grafana، يمكنك:

- إنش��ء لوحات معلومات مخصصة مع الرسوم البيانية، والمخططات، والجداول
- بناء استعلامات معقدة لتحليل بيانات العملاء، واتجاهات الإيرادات، ومقاييس الخدمة
- إعداد تنبيهات بناءً على مؤشرات الأداء الرئيسية
- مشاركة لوحات المعلومات مع أعضاء الفريق الآخرين

تضمين لوحات المعلومات في OmniCRM

يمكنك تضمينها مباشرة في واجهة Grafana، بمجرد إنشاء لوحات المعلومات في OmniCRM. يتيح لك ذلك عرض أهم مقاييسك دون مغادرة CRM، مع الحفاظ على تنقل OmniCRM سليمًا.

لتضمين لوحة معلومات:

1. **إنشاء لوحة المعلومات الخاصة بك في Grafana** - انتقل إلى `/grafana` وصمم لوحة المعلومات الخاصة بك مع التصويرات التي تحتاجها.
2. **لاحظ معرف لوحة المعلومات** - عند عرض لوحة المعلومات الخاصة بك في Grafana، تحقق من عنوان URL. سيبدو مثل `/grafana/d/abc123/dashboard-name`. المعرف هو الجزء بعد `/d/` (في هذا المثال، `abc123`).
3. **تكوين لوحة المعلو��ات في بيئتك** - اتصل بمسؤول النظام الخاص بك لإضافة معرف لوحة المعلومات واسم العرض إلى تكوين النظام. يتم تكوين لوحات المعلومات عبر متغيرات البيئة.
4. **الوصول من الشريط الجانبي** - بمجرد تكوينها، ستظهر لوحة المعلومات الخاصة بك تحت قائمة "لوحات المعلومات". يمكنك النقر تلقائيًا في الشريط الجانبي لـ OmniCRM عليها لعرض لوحة المعلومات الكاملة مع الحفاظ على تنقل OmniCRM مرئيًا.

تكوين لوحة المعلومات

يتم تكوين لوحات المعلومات بواسطة مسؤول النظام الخاص بك باستخدام متغيرات البيئة. تتطلب كل لوحة معلومات:

- **معرف لوحة المعلومات** - المعرف الفريد من عنوان URL الخاص بـ Grafana
- **تسمية لوحة المعلومات** - الاسم الودود المعروض في الشريط الجانبي لـ OmniCRM

يمكن تكوين لوحات معلومات متعددة وستظهر كعناصر قائمة بالترتيب الذي تم تعريفه. انظر وثائق تكوين الإدارة ��لحصول على تفاصيل حول تكوين متغيرات البيئة.

من يمكنه الوصول إلى Grafana؟

الوصول إلى Grafana وعرض `grafana_access` يمكن فقط للمستخدمين الذين لديهم إذن لوحات المعلومات المضمنة. عادةً ما يتم منح هذا الإذن لـ:

- المسؤولين الذين يحتاجون إلى رؤية كاملة لمقاييس النظام
- محللي الأعمال الذين يقومون بإنشاء التقارير والتصورات
- فرق المالية التي تتعقب الإيرادات ومقاييس الفواتير
- فرق العمليات التي تراقب أداء الخدمة

الأسئلة الشائعة

هل يمكن أن يكون لدى المستخدم أدوار متعددة؟ نعم. الأذونات تضاف.

هل أحتاج إلى أذونات مخصصة؟ عادة لا. يغطي كتالوج الأذونات المدمج معظم الاحتياجات.

كيف تعرف قواعد "عرض الخاص ..." ما هو خاص بي؟ تقوم بتقييم الرابط بين المستخدم/الاتصال الخاص بك وعميلك (المستأجر).

هل يمكنني إنشاء لوحات معلومات Grafana الخاصة بي؟ نعم، إذا كان لدي�� إذن `grafana_access`. انتقل إلى `/grafana` للوصول إلى واجهة Grafana الكاملة.

بوابة الرعاية الذاتية

بوابة الرعاية الذاتية هي واجهة موجهة للعملاء تتيح للمستخدمين النهائيين إدارة حساباتهم الخاصة، عرض الاستخدام، دفع الفواتير، وتعديل الخدمات دون الحاجة إلى مساعدة من موظفي خدمة العملاء.

طرق الوصول:

- تسجيل الدخول المباشر عبر بيانات اعتماد العميل
- تقمص الموظف لأغراض استكشاف الأخطاء (من العملاء ← جهات الاتصال ← "تسجيل الدخول كـ مستخدم")

انظر أيضًا: رعاية العملاء لتفاصيل التقمص، تدفقات المصادقة لعملية تسجيل الدخول.

الغرض

توفر بوابة الرعاية الذاتية للعملاء:

1. **إدارة الحساب** - عرض وتحديث المعلومات الشخصية، جهات الاتصال، والعناوين
2. **نظرة عامة على الخدمات** - رؤية جميع الخدمات النشطة، الاستخدام، وتواريخ انتهاء الصلاحية
3. **تتبع الاستخدام** - مراقبة البيانات، الصوت، الرسائل القصيرة، والأرصدة النقدية
4. **الوصول إلى الفواتير** - عرض ودفع الفواتير، إدارة طرق الدفع
5. **تعديلات الخدمة** - إضافة شحنات، شراء إضافات، تعديل الخدمات
6. **توافر على مدار الساعة** - الوصول إلى معلومات الحساب في أي وقت دون الاتصال بالدعم

نظرة عامة على البوابة

عندما يقوم العملاء بتسجيل الدخول إلى بوابة الرعاية الذاتية، يرون لوحة تحكم مخصصة تحتوي على:

أقسام التنقل:

- **لوحة التحكم** - نظرة سريعة على الخدمات والنشاطات الأخيرة

- **الحساب** - التفاصيل الشخصية، جهات الاتصال، المواقع
- **الخدمات** - قائمة بجميع الخدمات مع الحالة والتفاصيل
- **الاستخدام** - معلومات استهلاك الرصيد وتواريخ انتهاء الصلاحية
- **الفوترة** - الفواتير، المعاملات، طرق الدفع
- **شحن** - شراء بيانات، صوت، رسائل قصيرة، أو رصيد نقدي

تفاصيل الحساب

يعرض قسم الحساب معلومات العميل ويسمح بتحديثات خدمة ذاتية محدودة.

الحقول القابلة للتحرير:

يمكن للعملاء تحديث:

- عنوان البريد الإلكتروني (يتطلب التحقق)
- رقم الهاتف
- كلمة المرور
- تفضيلات الإشعارات

معلومات عرض فقط:

- معرف العميل
- تاريخ إنشاء الحساب
- نوع العميل (فرد/عمل)
- المواقع (العناوين)
- جهات الاتصال المرتبطة

تحديث معلومات الحساب:

1. انتقل إلى **الحساب** ← **التفاصيل**
2. انقر على **"تعديل"** بجوار الحقل الذي تريد تحديثه
3. أدخل المعلومات الجديدة
4. انقر على **"حفظ التغييرات"**
5. لتغييرات البريد الإلكتروني، تحقق عبر الرابط المرسل إلى العنوان الجديد

ميزات الأمان:

- تتطلب تغييرات كلمة المرور ��لمة المرور الحالية
- تتطلب تغييرات البريد الإلكتروني التحقق
- يتم تسجيل النشاط لأغراض التدقيق
- إعدادات المصادقة الثنائية (إذا كانت مفعلة)

نظرة عامة على الخدمات

يعرض قسم الخدمات جميع الخدمات النشطة وغير النشطة للعميل.

عرض بطاقة الخدمة:

تظهر كل خدمة:

- **اسم الخدمة** - معرف قابل للقراءة البشرية (مثل "محمول - +44 7700 900123")
- **المنتج** - اسم الخطة أو المنتج
- **الحالة** - نشط، معلق، منتهي، ملغى
- **تاريخ الإنشاء** - تاريخ تفعيل الخدمة
- **تاريخ انتهاء الصلاحية** - متى تنتهي الخدمة (إذا كان ذلك مناسبًا)
- **التكلفة الشهرية** - الرسوم المتكررة
- **التجديد التلقائي** - مؤشر مفعل/معطل

إجراءات الخدمة:

- **عرض الاستخدام** - رؤية استهلاك الرصيد (البيانات، الصوت، الرسائل القصيرة، النقدية)
- **شحن** - إضافة رصيد أو بيانات
- **إضافات** - شراء ميزات إضافية
- **تعديل** - تغيير معلمات الخدمة (إذا كان مسموحًا)
- **عرض التفاصيل** - رؤية تكوين الخدمة الكامل

مؤشرات حالة الخدمة:

- ✅ الخدمة تعمل - **نشط**
- ⏳ يتم تجديدها أو تنتهي في أقل من 7 أيام - **ستنتهي قريبًا**
- ⚠️ الخدمة معطلة مؤقتًا (مشكلة في الدفع، تعليق يدوي) - **معلق**
- ● الخدمة لم تعد نشطة - **منتهي**

تتبع الاستخدام

يمكن للعملاء مراقبة استخدامهم عبر جميع أنواع الأرصدة في الوقت الحقيقي.

استخدام البيانات

عرض استهلاك البيانات مع تفاصيل حسب الفئة وتاريخ انتهاء الصلاحية.

عرض استخدام البيانات:

- **إجمالي الرصيد** - جميع فئات البيانات مجتمعة
- **المستخدم في هذه الفترة** - الاستهلاك منذ آخر تجديد
- **شريط التقدم** - تمثيل بصري للاستهلاك

- **معلوم��ت انتهاء الصلاحية** - متى تنتهي كل فئة
- **تفصيل الفئات** - فئات بيانات متعددة مع ترتيب الأولوية

أولوية الفئات:

عندما توجد فئات بيانات متعددة (مثل الخطة الأساسية + الشحنات)، يتم استهلاكها بترتيب الوزن:

- **الوزن 10** - يتم استهلاكه أولاً (عادةً بيانات ترويجية/مكافأة)
- **الوزن 20** - يتم استهلاكه ثانيًا (عادةً بيانات الشحن)
- **الوزن 30** - يتم استهلاكه أخيرًا (عادةً بيانات الخطة الأساسية)

استخدام الصوت

تتبع دقائق المكالمات المتبقية عبر جميع فئات الصوت.

عرض استخدام الصوت:

- **الدقائق المتبقية** - إجمالي رصيد الصوت
- **الدقائق المستخدمة** - الاستهلاك في هذه الفترة
- **سجل المكالمات** - المكالمات الأخيرة (إذا كانت مفعلة)
- **تواريخ انتهاء الصلاحية** - متى تنتهي فئات الصوت

- **الدقائق الدولية** - تتبع منفصل (إذا كان ذلك مناسبًا)

تفصيل الاستخدام:

- المكالمات داخل الشبكة - المكالمات ضمن نفس الشبكة
- المكالمات خارج الشبكة - المكالمات إلى شبكات أخرى
- المكالمات الدولية - المكالمات خارج البلاد
- الأرقام المميزة - أرقام بسعر خاص

استخدام الرسائل القصيرة

مراقبة مخصصات الرسائل القصيرة والاستهلاك.

عرض استخدام الرسائل القصيرة:

- **الرسائل المتبقية** - رصيد الرسائل القصيرة
- **المستخدمة هذا الشهر** - الرسائل المرسلة
- **الرسائل متعددة الوسائط مشمولة** - ما إذا كانت الرسائل متعددة الوسائط تحتسب ضمن الرصيد
- **الرسائل القصيرة الدولية** - تتبع منفصل (إذا كان ذلك مناسبًا)

الرصيد النقدي

عرض رصيد الائتمان المدفوع مسبقًا لخدمات الدفع حسب الاستخدام.

عرض الرصيد النقدي:

- **الرصيد الحالي** - الائتمان المتاح
- **آخر شحن** - أحدث مبلغ وتاريخ إعادة الشحن
- **تاريخ انتهاء الصلاحية** - متى ينتهي الرصيد (إذا كان ذلك مناسبًا)
- **إعادة الشحن التلقائي** - حالة مفعل/معطل

إدارة الفوترة

يمكن للعملاء عرض الفواتير، المعاملات، وإدارة طرق الدفع.

الفواتير

عرض ودفع الفواتير المستحقة مباشرة من البوابة.

عرض قائمة الفواتير:

- **رقم الفاتورة** - معرف فريد
- **التاريخ** - تاريخ إنشاء الفاتورة
- **تاريخ الاستحقاق** - موعد الدفع
- **المبلغ** - إجمالي مبلغ الفاتورة
- **الحالة** - مدفوعة، غير مدفوعة، متأخرة
- **الإجراءات** - تنزيل PDF، دفع عبر الإنترنت

دفع فاتورة:

1. انتقل إلى **الفواتير** → **الفوترة**
2. ابحث عن الفاتورة غير المدفوعة في القائمة
3. انقر على زر **"ادفع ا��آن"**
4. اختر طريقة الدفع (بطاقة محفوظة أو بطاقة جديدة)
5. أكد الدفع
6. استلم بريدًا إلكترونيًا للتأكيد

تنزيل الفواتير:

1. انقر على أيقونة **"تنزيل"** بجوار الفاتورة
2. يتم تنزيل الفاتورة الكاملة PDF مع تفاصيل الفاتورة
3. محفوظ للسجلات الضريبية والتوثيق

المعاملات

عرض تاريخ المعاملات الكامل بما في ذلك الرسوم، الائتمانات، والمدفوعات.

عرض المعاملات:

- **التاريخ** - تاريخ إنشاء المعاملة
- **الوصف** - ما هي الرسوم/الائتمان
- **المبلغ** - الرسوم (إيجابية) أو الائتمان (سلبية)
- **الفاتورة** - أي فاتورة تتضمن هذه المعاملة
- **الحالة** - مفوترة أو غير مفوترة

طرق الدفع

إدارة بطاقات الائتمان المحفوظة للفوترة التلقائية والمدفوعات عبر الإنترنت.

إدارة طرق الدفع:

- **إضافة بطاقة** - إضافة بطاقة ائتمان/خصم جديدة بشكل آمن عبر Stripe
- **تعيين كافتراضي** - اختيار طريق الدفع الأساسية
- **إزالة بطاقة** - حذف بطاقات منتهية أو غير مستخدمة
- **تفاصيل البطاقة** - عرض آخر 4 أرقام، تاريخ انتهاء الصلاحية، علامة البطاقة

انظر طرق الدفع لوثائق إدارة الدفع التفصيلية.

شحن / إعادة شحن

شراء بيانات إضافية، صوت، رسائل قصيرة، أو رصيد نقدي على الفور.

عملية الشحن:

1. انتقل إلى **الخدمات** ← اختر الخدمة ← **شحن"**
2. اختر منتج الشحن من الكتالوج
3. اختر المبلغ (خيارات مسبقة أو مخصصة)
4. راجع معلومات التكلفة والانتهاء
5. اختر طريقة الدفع
6. أكد الشراء
7. يتم تحديث الرصيد على الفور

الشحنات المتاحة:

- **شحن البيانات** - خيارات 1GB، 5GB، 10GB، 20GB، 50GB
- **شحن الصوت** - دقائق مكالمات إضافية
- **حزم الرسائل القصيرة** - حزم الرسائل
- **رصيد نقدي** - رصيد مدفوع مسبقًا (5£، 10£، 20£، 50£، 100£)

انظر الشحن وإعادة الشحن لعمل��ات الشحن التفصيلية.

إضافات الخدمة

شراء ميزات إضافية وتحسينات للخدمات الحالية.

الإضافات المتاحة:

- **التجوال الدولي** - تفعيل الخدمة في الخارج
- **عنوان IP ثابت** - IP ثابت للإنترنت المنزلي
- **محتوى متميز** - قنوات IPTV، خدمات البث
- **ترقيات الأجهزة** - تأجير مودم، صناديق استقبال
- **تعزيزات السرعة** - زيادات مؤقتة في عرض النطاق الترددي

شراء الإضافات:

1. انتقل إلى **الخدمات** ← اختر الخدمة ← **الإضافات"**
2. تصفح الإضافات المتاحة لهذا النوع من الخدمة
3. انقر على **إضافة إلى الخدمة"** على الإضافة المرغوبة
4. راجع التكلفة (مرة واحدة + متكررة)
5. أكد الشراء

6. يتم توفير الإضافة تلقائيًا

انظر تعديل الخدمات - الإضافات لتفاصيل إدارة الإضافات.

الإشعارات والتنبيهات

يتلقى العملاء إشعارات آلية للأحداث المهمة:

إشعارات البريد الإلكتروني:

- تم إنشاء فاتورة وجاهزة للدفع
- تأكيد استلام الدفع
- تحذيرات انتهاء الخدمة (7 أيام، 3 أيام، 1 يوم)
- تنبيهات رصيد منخفض (بيانات، صوت، نقدي)
- تفعيل/تعطيل الخدمة
- طلبات إعادة تعيين كلمة المرور
- تنبيهات الأمان (تسجيل الدخول من جهاز جديد)

تنبيهات داخل البوابة:

- فواتير غير مدفوعة
- خدمات تنتهي قريبًا
- تحذيرات بيانات منخفضة (10% المتبقية)
- انتهاء صلاحية طريقة الدفع
- الإجراءات المطلوبة (التحقق من البريد الإلكتروني، تحديث طريقة الدفع)

تفضيلات الإشعارات:

يمكن للعملاء تكوين:

- تكرار إشعارات البريد الإلكتروني
- تنبيهات الرسائل القصيرة (إذا كانت مفعلة)
- عتبات التنبيه (مثل، الإخطار عندما يكون هناك أقل من 20% من البيانات المتبقية)
- فئات الإشعارات (الفوترة، الاستخدام، الخدمة)

قيود الخدمة الذاتية

بعض العمليات تتطلب مساعدة الموظفين:

تتطلب خدمة العملاء:

- تغيير نوع العميل (فرد ↔ عمل)
- نقل الخدمات بين العملاء
- إلغاء الخدمات (قد تكون خدمة ذاتية إذا كانت مفعلة)
- الاعتراض على الفواتير
- طلب استرداد الأموال
- تغيير جهة الاتصال الأساسية
- مشاكل توفير معقدة

قيود الأمان:

- لا يمكن عرض أو تعديل حسابات المستخدمين الآخرين
- محدود إلى بيانات العميل الخاصة (عزل المستأجر)
- لا يمكن الوصول إلى الوظائف الإدارية
- لا يمكن إلغاء الفواتير أو المعاملات
- لا يمكن تعديل تكوين الخدمة (فقط الإضافات/الشحنات)

الوصول إلى تقمص الموظف

يمكن لموظفي الدعم الوصول إلى بوابة الرعاية الذاتية كعميل لأغراض استكشاف الأخطاء.

عملية التقمص:

1. اختر العميل ← انتقل إلى **العملاء**
2. انتقل إلى علامة التبويب **جهات الاتصال**
3. ابحث عن جهة الاتصال المرتبطة بحساب المستخدم
4. انقر على زر **"تسجيل الدخول كـ مستخ��م"**
5. يفتح تبويب جديد مع عرض بوابة الرعاية الذاتية للعميل
6. يتم تسجيل جميع الإجراءات ونسبها إلى المستخدم المتقمص
7. يرى الموظف بالضبط ما يراه العميل

حالات الاستخدام:

- **استكشاف الأخطاء** - إعادة إنتاج المشكلات التي أبلغ عنها العميل
- **التحقق** - التأكد من أن تكوينات الخدمة تظهر بشكل صحيح
- **التدريب** - عرض ميزات البوابة
- **الدعم** - إرشاد العميل عبر البوابة أثناء عرض شاشته

الأمان والتدقيق:

- يتطلب إذن `admin` أو `can_impersonate`
- يتم تسجيل جميع الإجراءات في سجل التدقيق
- يرى العميل التقمص في سجل النشاط
- انتهاء جلسة بعد عدم النشاط
- لا يمكن تغيير كلمة مرور العميل أثناء التقمص

انظر رعاية العملاء - تقمص المستخدم لوثائق التقمص الكاملة.

استجابة الهاتف المحمول

بوابة الرعاية الذاتية مت��اوبة تمامًا ومُحسّنة للأجهزة المحمولة.

ميزات الهاتف المحمول:

- تنقل مُحسّن للمس
- تخطيطات مبسطة للشاشات الصغيرة
- إيماءات السحب للتنقل في الكاروسيل
- نماذج ومدخلات صديقة للهاتف المحمول
- eSIM مسح رمز الاستجابة السريعة لتوفير
- الاتصال الهاتفي بنقرة واحدة
- تكامل GPS للاكمال التلقائي للعناوين

تطبيق ويب تقدمي (PWA):

- تثبيت كتطبيق على الشاشة الرئيسية
- عرض البيانات الأخيرة دون اتصال
- إشعارات دفع (إذا كانت مفعلة)
- تحميل سريع مع عمال الخدمة

إعادة تعيين كلمة المرور واستعادة الحساب

يمكن للعملاء إعادة تعيين كلمات المرور المنسية دون الاتصال بالدعم.

إعادة تعيين كلمة المرور الخدمة الذاتية:

1. انقر على **"نسيت كلمة المرور"** في صفحة تسجيل الدخول
2. أدخل عنوان البريد الإلكتروني
3. استلم بريدًا إلكترونيًا لإعادة تعيين كلمة المرور (صالح لمد�ة ساعة)
4. انقر على الرابط في البريد الإلكتروني
5. أدخل كلمة مرور جديدة (يجب أن تلبي متطلبات التعقيد)
6. قدم وسجل الدخول باستخدام كلمة المرور الجديدة

متطلبات كلمة المرور:

- الحد الأدنى 8 أحرف
- على الأقل حرف كبير واحد
- على الأقل حرف صغير واحد
- على الأقل رقم واحد

- على الأقل رمز خاص واحد (!@#$%^&*)

قفل الحساب:

بعد 5 محاولات تسجيل دخول فاشلة:

- يتم قفل الحساب لمدة 30 دقيقة
- يتم إرسال بريد إلكتروني لإعادة تعيين كلمة المرور تلقائيًا
- يتم إرسال إشعار أمني إلى البريد الإلكتروني المسجل

أفضل الممارسات للعملاء

توصيات الأمان:

1. تفعيل المصادقة الثنائية لتعزيز الأمان
2. استخدام كلمة مرور قوية وفريدة
3. الحفاظ على تحديث عنوان البريد الإلكتروني للإشعارات
4. إعداد طريقة الدفع الافتراضية للتجديد التلقائي
5. مراقبة الاستخدام بانتظام لتجنب رسو�� الاستخدام الزائد
6. حفظ رموز النسخ الاحتياطي في مكان آمن (إذا كانت المصادقة الثنائية مفعلة)
7. تسجيل الخروج بعد استخدام أجهزة الكمبيوتر المشتركة/العامة

إدارة الاستخدام:

1. تفعيل تنبيهات الرصيد المنخفض (10-20% المتبقية)
2. شحن قبل انتهاء الأرصدة لتجنب انقطاع الخدمة
3. مراجعة الفواتير الشهرية للرسوم غير المتوقعة
4. تحديث طرق الدفع قبل انتهاء صلاحية البطاقات
5. مراقبة استخدام البيانات طوال الشهر لتجنب التقييد

تصعيد الدعم:

إذا لم تحل الخدمة الذاتية المشكلة:

1. تحقق من قاعدة المعرفة / مقالات المساعدة (إذا كانت متاحة)
2. مراجعة سجل النشاط للتغييرات الأخيرة
3. الاتصال بخدمة العملاء عبر الهاتف أو البريد الإلكتروني أو الدردشة

4. تقديم معرف العميل وتفاصيل الخدمة للحصول على حل أسرع

الوصول إلى واجهة برمجة التطبيقات

يمكن للعملاء الذين لدي��م متطلبات تقنية استخدام واجهة برمجة التطبيقات مباشرة.

تو‌ليد مفتاح واجهة برمجة التطبيقات:

متاح للعملاء التجاريين أو عند الطلب:

1. انتقل إلى **الحساب** ← **الوصول إلى واجهة برمجة التطبيقات**
2. انقر على **"تو‌ليد مفتاح واجهة برمجة التطبيقات"**
3. تعيين الأذونات (للقراءة فقط أو القراءة والكتابة)
4. تعيين تاريخ انتهاء الصلاحية
5. حفظ مفتاح واجهة برمجة التطبيقات بشكل آمن (يظهر مرة واحدة فقط)

حالات استخدام واجهة برمجة التطبيقات:

- مراقبة الاستخدام الآلي
- التكامل مع أنظمة الفوترة الداخلية
- الشحن البرمجي
- توفير الخدمة عبر السكريبتات
- تصدير البيانات للتحليلات

انظر وثائق واجهة برمجة التطبيقات لتفاصيل النقاط النهائية والأمثلة.

الأسئلة الشائعة

س: لماذا لا أستطيع رؤية جميع خدماتي؟

ج: تأكد من أنك مسجل الدخول باستخدام الحسا�� الصحيح. إذا كان لديك حسابات عملاء متعددة، فإن لكل منها خدمات منفصلة. اتصل بالدعم لدمج الحسابات إذا لزم الأمر.

س: فشلت مدفوعتي لكن تم تحميلي. ماذا أفعل؟

ج: تحقق من كشف حسابك البنكي للرسوم المعلقة. إذا ظهرت الرسوم ولكن الفاتورة لا تزال تظهر غير مدفوعة، اتصل بالدعم مع رقم مرجع المعاملة.

س: كيف يمكنني إلغاء خدمة؟

ج: انتقل إلى تفاصيل الخدمة وانقر على "إلغاء الخدمة" (إذا كانت الخدمة الذاتية مفعلة). خلاف ذلك، اتصل بخدمة العملاء لمعالجة الإلغاء.

س: هل يمكنني نقل خدمة إلى شخص آخر؟

ج: لا، تتطلب عمليات نقل الخدمة مساعدة خدمة العملاء لأسباب تتعلق بالأمان والامتثال.

س: لماذا يتناقص رصيد بياناتي أسرع مما هو متوقع؟

ج: تحقق من تحديثات التطبيقات في الخلفية، جودة بث الفيديو، والنسخ الاحتياطي التلقائي السحابي. راجع تفصيل الاستخدام في البوابة للاستهلاك التفصيلي.

س: فقدت جهاز المصادقة الثنائية الخاص بي. كيف يمكنني استعادة الوصول؟

ج: استخدم رموز النسخ الاحتياطي لتسجيل الدخول، ثم قم بتعطيل وإعادة تفعيل المصادقة الثنائية. إذا لم يكن لديك رموز نسخ احتياطي، اتصل بالدعم لإعادة تعيين المصادقة الثنائية (يتطلب التحقق من الهوية).

س: هل يمكنني دفع فاتورة دون تسجيل الدخول؟

ج: قد يتوفر دفع الفاتورة كزائر عبر رابط الفاتورة المباشر. خلاف ذلك، يتطلب تسجيل الدخول لأسباب أمنية.

س: كيف يمكنني تنزيل جميع فواتيري دفعة واحدة؟

ج: حاليًا يتطلب تنزيل كل فاتورة بشكل فردي. للتنزيلات الجماعية، اتصل بالدعم أو استخدم واجهة برمجة التطبيقات إذا كانت متاحة.

الوثائق ذات الصلة

- رعاية العملاء - تقمص المستخدم - وصول موظفي الدعم لاستكشاف الأخطاء
- طرق الدفع - إدارة البطاقات والدفع
- الشحن وإعادة الشحن - شراء رصيد إضافي
- استخدام الخدمة - فهم تتبع الأرصدة
- نظرة عامة على الفوترة - مفاهيم الفوترة والفواتير
- تدفقات المصادقة - تسجيل الدخول والأمان

- المصادقة الثنائية - إعداد المصادقة الثنائية
- تعديلات الخدمة - إضافة ميزات وإضافات

نظام قاعدة المعرفة للأسئلة الشائعة

يوفر نظام الأسئلة الشائعة (FAQ) قاعدة معرفة ذاتية الخدمة لكل من العملاء والموظفين، مما يقلل من حجم تذاكر الدعم ويساعد في حل المشكلات بشكل أسرع.

انظر أيضًا: تذاكر الدعم لإدارة التذاكر، رعاية العملاء لميزات دعم العملاء.

نظرة عامة

يمكن نظام الأسئلة الشائعة من:

1. **الخدمة الذاتية**: يجد العملاء الإجابات دون إنشاء تذاكر
2. **قاعدة المعرفة**: مستودع مركزي للحلول
3. **موارد الموظفين**: مرجع سريع لموظفي الدعم
4. **التحسين المستمر**: تتبع الأسئلة الشائعة الأكثر فائدة
5. **متعددة الفئات**: تنظيم الأسئلة الشائعة حسب الموضوع والعلامات
6. **تحكم في الرؤية**: الأسئلة الشائعة العامة مقابل الداخلية
7. **التحليلات**: عدد المشاهدات وتقييمات الفائدة

الميزات الرئيسية

إدارة الأسئلة الشائعة

- **إنشاء الأسئلة الشائعة**: كتابة الأسئلة وإجابات مفصلة
- **محتوى غني**: تنسيق Markdown، كتل كود، صور
- **التصنيف**: تنظيم حسب الفئة والعلامات
- **الرؤية**: التحكم في الوصول للعملاء مقابل الموظفين فقط
- **البحث**: بحث نص كامل عبر العناوين والمحتوى
- **إصدار**: تتبع التغييرات والتحديثات بمرور الوقت

التحليلات

- **تتبع المشاهدات**: حساب عدد مرات مشاهدة السؤال الشائع
- **الفائدة**: يقوم العملاء بتقييم ما إذا كان السؤال الشائع مفيدًا
- **الأسئلة الشائعة الشعبية**: أكثر الأسئلة مشاهدة
- **الأكثر فائدة**: الحلول الأعلى تقييمًا
- **الأداء المنخفض**: الأسئلة الشائعة التي تم وضع علامة عليها على أنها غير مفيدة (تحتاج إلى تحسي��)

التنظيم

- **الفئات**: تجميع الأسئلة الشائعة ذات الصلة (الفواتير، التقنية، الحساب، إلخ)
- **العلامات**: علامات متعددة لكل سؤال شائع للتصنيف المتقاطع
- **البحث**: العثور السريع عن طريق الكلمة الرئيسية
- **التصفية**: التصفية حسب الفئة، العلامات، الرؤية
- **الفرز**: حسب التاريخ، المشاهدات، الفائدة، الأبجدية

هيكل الأسئلة الشائعة

الحقول المطلوبة

- **السؤال**: سؤال واضح وموجز (عنوان السؤال الشائع)
- **الإجابة**: رد مفصل مع تعليمات خطوة بخطوة
- **الفئة**: التصنيف الأساسي
- **الرؤية**: مرئي للعملاء أو للموظفين فقط

الحقول الاختيارية

- **العلامات**: كلمات رئيسية إضافية للتصنيف
- **الأسئلة الشائعة ذات الصلة**: روابط لأسئلة مشابهة (ميزة مستقبلية)
- **المرفقات**: لقطات شاشة، مستندات، مقاطع فيديو (ميزة مستقبلية)
- **المؤلف**: عضو الموظف الذي أنشأ السؤال الشائع
- **آخر تحديث**: متى تم تعديل السؤال الشائع آخر مرة

مثال على سؤال شائع

السؤال: كيف يمكنني التحقق من استخدام بياناتي؟

الفئة: إدارة الحساب

العلامات: البيانات، الاستخدام، الرصيد، الخدمة الذاتية

الإجابة:

```
يمكنك التحقق من استخدام بياناتك الحالي بعدة طرق:

## عبر بوابة الخدمة الذاتية

1. تسجيل الدخول إلى حسابك على portal.example.com
2. الانتقال إلى علامة **الخدمات**
3. النقر على خدمة الهاتف المحمول الخاصة بك
4. عرض قسم **الرصيد**
5. يظهر استخدام بياناتك كـ "البيانات المستخدمة / إجمالي البيانات"

## عبر التطبيق المحمول

1. فتح التطبيق المحمول
2. النقر على **خدماتي**
3. اختيار خط الهاتف المحمول الخاص بك
4. يظهر استخدام البيانات على شاشة النظرة العامة

## عبر الرسائل القصيرة

أرسل **BAL** إلى 123456 لتلقي رصيدك الحالي عبر الرسائل القصيرة.

## تحتاج إلى مس��عدة؟

إذا لم تتمكن من رؤية استخدامك، يرجى الاتصال بالدعم على
support@example.com
```

الرؤية: مرئي للعملاء

عرض وتحرير الأسئلة الشائعة

غني مع تنسيق مناسب HTML كـ Markdown عند عرض سؤال شائع، يتم عرض جميع تنسيقات للعناوين والقوائم وكتل الكود والروابط.

المعروض مع تنسيقات Markdown الشكل 1: عرض تفاصيل السؤال الشائع يظهر محتوى للعناوين والقوائم وكتل الكود

تقدم صفحة تفاصيل السؤال الشائع:

- منسق HTML تلقائيًا إلى Markdown **المحتوى المعروض**: يتم تحويل
- **الإحصائيات**: عدد المشاهدات وتقييمات الفائدة
- Markdown **وضع التحرير**: تحرير مباشر مع دعم
- **تبديل الرؤية**: إظهار/إخفاء من العملاء
- **إجراءات سريعة**: نسخ الرابط، حذف السؤال الشائع

عرض العملاء

يرى العملاء عرضًا نظيفًا ومنسقًا للأسئلة الشائعة دون قدرات التحرير:

الشكل 2: عرض السؤال الشائع الموجه للعملاء مع تقييم الفائدة وتنسيق نظيف

إنشاء الأسئلة الشائعة

عبر واجهة الويب

1. انتقل إلى **الدعم → الأسئلة الشائعة**
2. انقر على **إنشاء سؤال شائع**
3. املأ تفاصيل السؤال الشائع:
    - السؤال (العنوان)
    - الإجابة (المحتوى مع Markdown)
    - الفئة
    - العلامات (مفصولة بفواصل)
    - الرؤية (مربع اختيار مرئي للعملاء)
4. انقر على **إنشاء**

عبر واجهة برمجة التطبيقات

```
POST /crm/faq/
```

الإذن المطلوب: `create_faq`

الطلب:

```
{
  "question": "كيف يمكنني إعادة تعيين كلمة المرور الخاصة بي؟",
  "answer": "لإعادة تعيين كلمة المرور الخاصة بك:\n1. انتقل إلى صفحة تسجيل الدخول\n2. انقر على 'نسيت كلمة المرور'\n3. أدخل بريدك الإلكتروني\n4. تحقق من صندوق الوارد الخاص بك للحصول على رابط إعادة التعيين\n5. انقر على الرابط واضبط كلمة مرور جديدة",
  "category": "الحساب",
  "tags": "كلمة المرور، تسجيل الدخول، الحساب، إعادة التعيين",
  "is_visible_to_customer": true
}
```

الاستجابة:

```
{
  "faq_id": 42,
  "question": "كيف يمكنني إعادة تعيين كلمة المرور الخاصة بي؟",
  "answer": "لإعادة تعيين كلمة المرور الخاصة بك:\n1. انتقل إلى صفحة تسجيل الدخول...",
  "category": "الحساب",
  "tags": "كلمة المرور، تسجيل الدخول، الحساب، إعادة التعيين",
  "is_visible_to_customer": true,
  "view_count": 0,
  "helpful_count": 0,
  "not_helpful_count": 0,
  "created": "2026-01-07T10:00:00Z",
  "last_modified": "2026-01-07T10:00:00Z",
  "created_by_user": {
    "id": 1,
    "username": "admin",
    "first_name": "Admin",
    "last_name": "User"
  }
}
```


تحرير الأسئلة الشائعة

تحديث المحتوى

1. افتح السؤال الشائع من القائمة
2. انقر على **تحرير**
3. تعديل السؤال، الإجابة، أو البيانات الوصفية
4. انقر على **حفظ**

يتم ت��بع التغييرات مع:

- الطابع الزمني لآخر تعديل
- آخر تعديل بواسطة المستخدم
- تاريخ التغييرات (سجل التدقيق)

تبديل الرؤية

تحكم في ما إذا كان يمكن للعملاء رؤية سؤال شائع:

1. افتح السؤال الشائع
2. انقر على تبديل **الرؤية**
3. اختر:
    - **مرئي للعملاء**: يظهر في بوابة العملاء
    - **موظف فقط**: مرئي فقط لموظفي الدعم
4. حفظ التغييرات

حالات الاستخدام للموظفين فقط:

- الإجراءات والعمليات الداخلية
- أدلة استكشاف الأخطاء مع معلومات حساسة
- مسودات الأسئلة الشائعة في انتظار الموافقة
- الحلول المهجورة التي تُحتفظ بها للرجوع إليها

فئات الأسئلة الشائعة

الفئات القياسية

تنظيم الأسئلة الشائعة في مجموعات منطقية:

- **إدارة الحساب**: تسجيل الدخول، كلمات المرور، الملفات الشخصية
- **الفواتير**: الفواتير، المدفوعات، الرسوم
- **الدعم الفني**: مشكلات الخدمة، استكش��ف الأخطاء
- **الميزات**: كيفية استخدام ميزات معينة
- **الخدمات المحمولة**: بطاقات SIM، التجوال، البيانات
- **خدمات الإنترنت**: الاتصال، السرعات، المعدات
- **خدمات VoIP**: المكالمات، البريد الصوتي، الميزات
- **خدمات IPTV**: القنوات، البث، الأجهزة

إنشاء الفئات

يتم إنشاء الفئات تلقائيًا عند استخدامها في سؤال شائع. لتوحيد المعايير:

1. استخدم أسماء فئات متسقة
2. اتبع قواعد التسمية
3. مراجعة قائمة الفئات بشكل دوري
4. دمج الفئات المكررة/المشابهة

إدارة الفئات

عرض جميع الفئات:

```
GET /crm/faq/categories
```

يعيد قائمة بجميع الفئات مع عدد الأسئلة الشائعة.

عرض العملاء:

```
GET /crm/faq/customer/categories
```

يعيد فقط الفئات التي تحتوي على أسئلة شائعة مرئية.

العلامات

الغرض

توفر العلامات تنظيمًا متقاطعًا:

- يمكن أن يحتوي سؤال شائع واحد على علامات متعددة
- تكمل العلامات هيكل الفئة
- تمكين التصفية والبحث المرن
- تساعد العملاء في العثور على المحتوى ذي الصلة

أمثلة على العلامات

- **تقني:** استكشاف الأخطاء، خطأ، عطل، إصلاح، إعادة تعيين
- **حساب:** تسجيل الدخول، كلمة المرور، الملف الشخصي، الإعدادات، الأمان
- **فواتير:** فاتورة، دفع، رسوم، استرداد، نزاع
- **خدمة:** تفعيل، إلغاء، ترقية، تغيير
- **منصة:** محمول، ويب، تطبيق، بوابة، خدمة ذاتية

أفضل الممارسات

1. **تسمية متسقة**: استخدم الحروف الصغيرة، الشكل المفرد
2. **علامات محددة**: "إعادة تعيين كلمة المرور" وليس فقط "كلمة المرور"
3. **تجنب التكرار**: لا تكرر الفئة في العلامات
4. **تحديد العدد**: 3-5 علامات لكل سؤال شائع
5. **توحيد**: الحفاظ على مسرد العلامات

بوابة الأسئلة الشائعة للعملاء

الوصول إلى الأسئلة الشائعة

يمكن للعملاء الوصول إلى الأسئلة الشائعة من خلال:

1. **بوابة الخدمة الذاتية**: قسم مخصص للأسئلة الشائعة
2. **أثناء إنشاء التذكرة**: اقتراحات "تحقق من هذه الأسئلة الشائعة أولاً"
3. **البحث**: البحث العالمي يشمل محتوى الأسئلة الشائعة
4. **البريد الإلكتروني**: روابط في الردود الآلية

عرض الأسئلة الشائعة

يعرض عرض العملاء:

- **السؤال**: عنوان السؤال الشائع/السؤال
- **الإجابة**: المحتوى الكامل المنسق
- **الفئة**: لتصفح الأسئلة الشائعة ذات الصلة

- **العلامات**: تنقل إضافي
- **تقييم الفائدة**: ملاحظات بالإبهام لأعلى/لأسفل
- **الأسئلة الشائعة ذات الصلة**: أسئلة مشابهة (مستقبلية)

البحث والتصفية

يمكن للعملاء:

- **البحث**: بحث نص كامل عبر جميع الأسئلة الشائعة المرئية
- **التصفية حسب الفئة**: عرض الأسئلة الشائعة في فئة معينة
- **التصفية حسب العلامة**: العثور على الأسئلة الشائعة مع علامات معينة
- **الفرز**: حسب الشعبية، الحداثة، أو الأبجدية
- **التصفح**: استكشاف جميع الأسئلة الشائعة أو قوائم الفئات

تقييم الأسئلة الشائعة

يقدم العملاء ملاحظات:

1. قراءة السؤال الشائع
2. التمرير إلى الأسفل
3. النقر على **هل كان هذا مفيدًا؟**
4. اختيار **نعم** أو **لا**
5. تسجيل التقييم

تساعد التقييمات في تحديد:

- **الأسئلة الشائعة المفيدة**: تصويتات نعم عالية
- **تحتاج إلى تحسين**: تصويتات لا عالية
- **محتوى قديم**: انخفاض الفائدة
- **المواضيع الشعبية**: مشاهدات عالية + تقييمات عالية

أدوات الأسئلة الشائعة للموظفين

الأسئلة الشائعة الداخلية

أسئلة شائعة للموظفين فقط لـ:

- **الإجراءات**: سير العمل والعمليات الداخلية
- **استكشاف الأخطاء**: خطوات التشخيص المتقدمة
- **التصعيد**: متى وكيف يتم تصعيد المشكلات
- **الأدوات**: كيفية استخدام الأنظمة الداخلية
- **السياسات**: سياسات وإرشادات الشركة

المرجع السريع

يمكن للموظفين:

- **البحث في جميع الأسئلة الشائعة**: بما في ذلك المحتوى الخاص بالموظفين فقط
- **عرض التحليلات**: رؤية أي الأسئلة الشائعة الأكثر فائدة
- **نسخ الروابط**: مشاركة روابط الأسئلة الشائعة مع العملاء
- **اقتراح التعديلات**: اقتراح تحسينات على الأسئلة الشائعة
- **إنشاء من التذكرة**: تحويل حل التذكرة إلى سؤال شائع

تحليلات الأسئلة الشائعة

تتبع أداء الأسئلة الشائعة:

عدد المشاهدات:

- إجمالي المشاهدات لكل سؤال شائع
- المشاهدات بمرور الوقت
- الأسئلة الشائعة الأكثر شعبية
- الأسئلة الشائعة المتداولة (ارتفاع مفاجئ)

الفائدة:

- تصويتات مفيدة (إبهام لأعلى)

- تصويتات غير مفيدة (إبهام لأسفل)
- نسبة الفائدة
- الأسئلة الشائعة الأكثر/الأقل فائدة

نقاط النهاية لواجهة برمجة التطبيقات:

```
GET /crm/faq/popular?limit=10
GET /crm/faq/helpful?limit=10
```

لتحسين المستمر��

استخدم التحليلات لـ:

1. **تحديد الفجوات**: عمليات البحث الشائعة بدون نتائج
2. **تحديث المحتوى**: تحسين الأسئلة الشائعة ذات التقييم المنخفض
3. **إيقاف القديمة**: إزالة أو تحديث الأسئلة الشائعة القديمة
4. **ترويج الأفضل**: عرض الأسئلة الشائعة ذات التقييم العالي
5. **إنشاء جديدة**: إضافة أسئلة شائعة للتذاكر المتكررة

تنسيق Markdown

تدعم إجابات الأسئلة الشائعة Markdown للتنسيق الغني:

تنسيق النص

```
**نص عريض** للتأكيد
*نص مائل* للتأكيد الطفيف
~~شطب~~ للمعلومات المهجورة
`كود مضمن` للمصطلحات التقنية
```

العناوين

```
## عنوان القسم
### عنوان الفرع
#### عنوان فرعي
```

القوائم

```
قائمة مرتبة:
1. الخطوة الأولى
2. الخطوة الثانية
3. الخطوة الثالثة

قائمة غير مرتبة:
- النقطة الأولى
- النقطة الثانية
- النقطة الثالثة
```

الروابط

```
[نص الرابط](https://example.com)
[البريد الإلكتروني للدعم](mailto:support@example.com)
```

كتل الكود

````
```bash
command --with-flags
```

```json
{
  "example": "data"
}
```
````

الصور

```
![نص بديل](https://example.com/image.png)
```

الجداول

```
| العمود 1 | العمود 2 |
|----------|----------|
| البيانات 1 | البيانات 2 |
```

الاقتباسات

```
> ملاحظة أو نصيحة مهمة
> تستمر على عدة أسطر
```

مرجع واجهة برمجة التطبيقات

الحصول على جميع الأسئلة الشائعة (موظف)

```
GET /crm/faq/
```

الإذن المطلوب: `view_faq`

معلمات الاستعلام:

- `page`: رقم الصفحة (الافتراضي: 1)
- `per_page`: النتائج لكل صفحة (الافتراضي: 20)
- `sort`: حقل الفرز (created، view_count، helpful_count)
- `order`: ترتيب الفرز (asc، desc)
- `search`: استعلام البحث
- `category`: تصفية حسب الفئة
- `tags`: تصفية حسب العلامات (مفصولة بفواصل)

- `is_visible_to_customer`: تصفية حسب الرؤية (true/false)

الحص��ل على الأسئلة الشائعة للعملاء

```
GET /crm/faq/customer
```

المصادقة مطلوبة: نعم **الإذن:** أي مستخدم مصدق

يعيد فقط الأسئلة الشائعة المرئية للعملاء.

الحصول على سؤال شائع واحد

```
GET /crm/faq/faq_id/{faq_id}
```

عرض الموظف: لا يزيد من عدد المشاهدات

```
GET /crm/faq/customer/faq_id/{faq_id}
```

عرض العميل: يزيد من عدد المشاهدات

تحديث السؤال الشائع

```
PATCH /crm/faq/faq_id/{faq_id}
```

الإذن المطلوب: `update_faq`

حذف السؤال الشائع

```
DELETE /crm/faq/faq_id/{faq_id}
```

الإذن المطلوب: `delete_faq`

يؤدي إلى حذف ناعم (يتم وضع علامة كحذف، يحتفظ بالبيانات).

تبديل الرؤية

```
POST /crm/faq/faq_id/{faq_id}/visibility
```

الإذن المطلوب: `manage_faq_visibility`

الطلب:

```
{
  "is_visible_to_customer": true
}
```

وضع علامة على السؤال الشائع كمفيد

```
POST /crm/faq/faq_id/{faq_id}/helpful
```

المصادقة مطلوبة: نعم **الإذن:** أي مستخدم مصدق

الطلب:

```
{
  "is_helpful": true
}
```

الحصول على الأسئلة الشائعة الشعبية

```
GET /crm/faq/popular?limit=10
```

يعيد الأسئلة الشائعة مرتبة حسب عدد المشاهدات.

الحصول على الأسئلة الشائعة الأكثر فائدة

```
GET /crm/faq/helpful?limit=10
```

يعيد الأسئلة الشائعة مرتبة حسب عدد الفائدة.

أفضل الممارسات

كتابة المحتوى

1. **أسئلة واضحة**: استخدم اللغة الطبيعية التي سيستخدمها العملاء
2. **إجابات كاملة**: تضمين جميع الخطوات، دون افتراضات
3. **لقطات شاشة**: إضافة وسائل بصرية للإجراءات المعقدة
4. **روابط**: الإشارة إلى الأسئلة الشائعة والموارد ذات الصلة
5. **اختبار**: التحقق من دقة التعليمات

التنظيم

1. **فئات متسقة**: استخدم أسماء فئات موحدة
2. **علامات محددة**: استخدم علامات وصفية وقابلة للبحث
3. **تجنب التكرار**: دمج الأسئلة الشائعة المماثلة
4. **المرجعية المتبادلة**: ربط ا��أسئلة الشائعة ذات الصلة معًا
5. **المراجعة المنتظمة**: تحديث المحتوى القديم ربع سنويًا

الصيانة

1. **مراقبة التحليلات**: تتبع المشاهدات والفائدة
2. **التحديث بانتظام**: الحفاظ على تحديث المحتوى
3. **إيقاف الأسئلة الشائعة القديمة**: إزالة المعلومات القديمة
4. **دمج الملاحظات**: تحسين بناءً على التقييمات
5. **إنشاء من التذاكر**: تحويل القضايا الشائعة إلى أسئلة شائعة

استكشاف الأخطاء

المشكلة: السؤال الشائع لا يظهر للعملاء

- الحل: تحقق من تمكين إعداد "مرئي للعملاء"

- الحل: تحقق من أن السؤال الشائع ليس م marked as deleted
- الحل: مسح ذاكرة التخزين المؤقت لبوابة العملاء

المشكلة: البحث لا يجد السؤال الشائع

- الحل: تحقق من الإملاء وجرب كلمات رئيسية مختلفة
- الحل: تحقق من أن السؤال الشائع في الفئة الصحيحة
- الحل: إضافة المزيد من العلامات ذات الصلة إلى السؤال الشائع

المشكلة: التحليلات لا تتحدث

- الحل: انتظر لتحديث الذاكرة المؤقتة (قد يستغرق بضع دقائق)
- الحل: تحقق من تمكين تتبع المشاهدات
- الحل: تحقق من أن العميل مصدق عند العرض

المشكلة: لا يمكن تحرير السؤال الشائع

- الحل: تحقق من أن لديك إذن `update_faq`
- الحل: تحقق من أن السؤال الشائع ليس مقفلاً بواسطة مستخدم آخر
- الحل: تأكد من أن السؤال الشائع موجود وليس محذوفًا

نظام تذاكر الدعم

يوفر نظام تذاكر الدعم في OmniCRM منصة شاملة لإدارة طلبات دعم العملاء، وتتبع المشكلات، والحفاظ على تاريخ اتصال مفصل مع العملاء.

انظر أيضًا: الأسئلة الشائعة لقاعدة المعرفة الذاتية، رعاية العملاء للتجسيد واستكشاف الأخطاء، العملاء لإدارة العملاء.

الشكل 1: قائمة تذاكر الدعم تظهر جميع التذاكر مع التصفية، البحث، الحالة، الأولوية، والموظفين المعينين

نظرة عامة

يمكن نظام التذاكر من:

1. **الدعم المركزي**: تتبع جميع مشكلات العملاء في مكان واحد
2. **تكامل البريد الإلكتروني**: إنشاء تذاكر تلقائيًا من رسائل البريد الإلكتروني للعملاء
3. **التعاون**: يمكن لعدة موظفين العمل على التذاكر معًا
4. **قاعدة المعرفة**: ربط الأسئلة الشائعة بالمشكلات الشائعة لحل أسرع
5. **تتبع SLA**: مراقبة عمر التذكرة وأوقات الاستجابة

6. **بوابة العملاء**: يمكن للعملاء عرض والرد على تذاكرهم
7. **سجل التدقيق**: تاريخ كامل لجميع تفاعلات التذاكر

الميزات الرئيسية

إدارة التذاكر

- **إنشاء التذاكر**: إنشاء يدوي أو تلقائي من البريد الإلكتروني
- **تحديث التذاكر**: إضافة تعليقات، تغيير الحالة، إعادة التعيين
- **المرفقات**: تحميل الملفات (صور، مستندات، سجلات)
- **العلامات**: تصنيف التذاكر للتقارير والتصفية
- **البحث**: بحث نصي كامل عبر التذاكر، التحديثات، والمرفقات
- **التصفية**: تصفية حسب الحالة، الأولوية، المعين، العميل، الخدمة
- **عرض في الوقت الحقيقي**: تتبع من يقوم حاليًا بعرض كل تذكرة
- **آخر نشاط**: رؤية متى تم عرض أو تحديث التذاكر آخر مرة

التواصل

- **إشعارات البريد الإلكتروني**: إشعارات تلقائية عند تحديث التذاكر
- **إشعارات داخل التطبيق** OmniCRM: إشعارات في الوقت الحقيقي في واجهة
- **الإشارات** @username: ذكر أعضاء الفريق باستخدام@ لإخطارهم
- **المشتركين**: إضافة موظفين وجهات اتصال إلى إشعارات التذاكر
- **رؤية العميل**: التحكم في التحديثات التي يمكن للعملاء رؤيتها
- **ملاحظات داخلية**: إضافة ملاحظات خاصة غير مرئية للعملاء
- **تتبع سلاسل البريد الإلكتروني**: تتبع تلقائي لسلاسل البريد الإلكتروني
- **المشاهدون النشطون**: رؤية من يقوم حاليًا بعرض التذكرة في الوقت الحقيقي

تكامل الخدمة

- **ربط بالخدمات**: ربط التذاكر بخدمات العملاء المحددة
- **سياق الخدمة**: عرض تفاصيل الخدمة، الاستخدام، والتاريخ من التذكرة
- **إجراءات سريعة**: إعادة شحن، عرض الإضافات، تعديل الخدمة مباشرة من التذكرة

دورة حياة التذكرة

1. إنشاء التذكرة

عبر واجهة الويب:

1. انتقل إلى **الدعم** ← **التذاكر**
2. انقر على **إنشاء تذكرة**
3. املأ تفاصيل التذكرة:
    - العنوان (مطلوب)
    - الوصف (مطلوب)
    - العميل (اختياري، يمكن تعيينه لاحقًا)
    - الخدمة (اختياري، يرتبط بخدمة العملاء المحددة)
    - الأولوية (منخفضة، متوسطة، عالية، عاجلة)
    - الحالة (مفتوحة، قيد التقدم، معلقة، تم الحل، مغلقة)
    - العلامات (للتصنيف)
4. انقر على **إنشاء**

عبر البريد الإلكتروني:

يمكن للعملاء إنشاء تذاكر عن طريق إرسال بريد إلكتروني إلى عنوان الدعم الخاص بك. يقوم النظام تلقائيًا:

- بإنشاء تذكرة من موضوع البريد الإلكتروني ومحتواه
- ربط العميل إذا كان بريد المرسل يتطابق مع جهة اتصال
- إضافة مرفقات البريد الإلكتروني إلى التذكرة
- إرسال تأكيد للعميل

عبر API:

```
POST /crm/ticket/
```

الطلب:

```
{
  "title": "لا يمكن الوصول إلى البريد الصوتي",
  "description": "يبلغ العميل أن البريد الصوتي لا يعمل منذ الأمس",
  "customer_id": 290,
  "service_id": 167,
  "priority": "High",
  "status": "Open",
  "tags": "technical,voicemail"
}
```

2. تعيين التذكرة

يمكن تعيين التذاكر لموظفي الدعم للتعامل معها:

1. **التعيين اليدوي**: انقر على "تعيين" واختر عضو الفريق
2. **التعيين التلقائي**: تعيين تلقائي بناءً على القواعد (ميزة مستقبلية)
3. **إعادة التعيين**: نقل التذاكر بين أعضاء الفريق
4. **غير المعين**: يمكن أن تظل التذاكر غير المعينة في قائمة الانتظار

عندما يتم تعيين تذكرة:

- يتلقى الموظف المعين إشعارًا عبر البريد الإلكتروني
- يتم اشتراك الموظف تلقائيًا في تحديثات التذكرة
- تظهر التذكرة في عرض "تذاكري"

3. العمل على التذاكر

إضافة التحديثات:

1. افتح التذكرة
2. انقر على **إضافة تحديث** أو رد على التذكرة
3. أدخل رسالتك
4. استخدم **@username** لذكر وإخطار أعضاء الفريق المحددين (مثل، "هل @john يمكنك المساعدة في هذا؟")
5. اختر الرؤية:
   - **مرئي للعميل**: سيرى العميل هذا التحديث
   - **داخلي فقط**: ملاحظة خاصة للموظفين

6. انقر على **إرسال**

الإشارات@:

في تحديث التذكرة @username عندما تذكر عضو فريق باستخدام:

- يتلقون إشعارًا داخل التطبيق
- يتم اشتراكهم تلقائيًا في التذكرة
- يتضمن الإشعار رابطًا مباشرًا إلى التذكرة
- يعمل في كل من التحديثات المرئية للعملاء والتحديثات الداخلية

إرفاق الملفات:

1. انقر على زر **إرفاق الملفات**
2. اختر الملفات (حد أقصى 10 ميغابايت لكل منها)
3. يتم تحميل الملفات وربطها بالتذكرة
4. التنسيقات المدعومة: PDF، صور، مستندات، أرشيفات

تغيير الحالة:

- **مفتوحة**: تذكرة جديدة، تنتظر الاستجابة الأولية
- **قيد التقدم**: يعمل الموظف بنشاط على المشكلة
- **معلقة**: تنتظر رد العميل أ�� اعتماد خارجي
- **تم الحل**: تم إصلاح المشكلة، تنتظر التأكيد
- **مغلقة**: التذكرة مكتملة ومؤرشفة

تعيين الأولوية:

- **عاجلة** (SLA: 1 ساعة): مشكلة حرجة تؤثر على الخدمة
- **عالية** (SLA: 4 ساعات): مشكلة مهمة، الخدمة متدهورة
- **متوسطة** (SLA: 24 ساعة): مشكلة قياسية
- **منخفضة** (SLA: 48 ساعة): مشكلة بسيطة أو استفسار

4. حل التذكرة

عند حل تذكرة:

1. أضف التحديث النهائي الذي يشرح الحل

2. غير الحالة إلى **تم الحل**
3. يتلقى العميل إشعارًا
4. يمكن للعميل إعادة فتحها إذا استمرت المشكلة
5. بعد التأكيد، غير الحالة إلى **مغلقة**

عرض تفاصيل التذكرة

عند عرض تذكرة، ترى سلسلة المحادثة الكاملة مع جميع التحديثات، المرفقات، وشريط جانبي شامل يظهر معلومات العميل والخدمة.

الشكل 2: صفحة تفاصيل التذكرة تظهر سلسلة المحادثة، التحديثات، وبيانات التذكر

شريط جانبي تفاصيل التذكرة

تظهر كل تذكرة شريطًا جانبيًا شاملاً بمعلومات العميل، تفاصيل الخدمة، وإجراءات سريعة:

الشكل 3: شريط جانبي التذكرة مع معلومات العميل، تفاصيل الخدمة، الموظفين المعينين، وأزرار الإجراءات السريعة

يتضمن الشريط الجانبي:

معلومات العميل

- **اسم العميل**: مع رابط إلى ملف العميل
- **البريد الإلكتروني للاتصال**: البريد الإلكتروني الرئيسي الذي تُرسل إليه الردود
- **تعديل العميل**: تعديل سريع لتعيين العميل

معلومات الخدمة

عند ربط خدمة:

- **معرف الخدمة**: معرف فريد
- **اسم الخدمة**: اسم وصفي
- **تواريخ الخدمة**: تواريخ البدء والانتهاء
- **تجديد تلقائي**: حالة التجديد

- **حالة الخدمة**: نشطة، غير نشطة، معلقة
- **نوع الخدمة**: موبايل، VoIP، إنترنت، IPTV
- **إجراءات سريعة:**
  - **إضافة**: إضافة حزم/ميزا�ا إلى الخدمة
  - **حذف**: إزالة الخدمة (مع تأكيد)
  - **تعديل الخدمة**: نموذج إدارة الخدمة الكامل

بيانات التذكرة

- **رقم التذكرة**: مرجع فريد (مثل، TKT-2026-00042)
- **تاريخ الإنشاء**: متى تم فتح التذكرة
- **آخر تحديث**: طابع زمني لأحدث نشاط
- **المعين إلى**: مالك التذكرة الحالي
- **مستوى الأولوية**: مع ترميز الألوان
- **الحالة الحالية**: مع شارة الحالة
- **العلامات**: تسميات التصنيف

المشتركين

قائمة بالمستخدمين وجهات الاتصال التي تتلقى الإشعارات:

- **الاشتراك التلقائي**: المنشئ والمعين
- **الاشتراكات اليدوية**: موظفون أو جهات اتصال أخرى
- **إضافة/إزالة**: إدارة قائمة المشتركين

تكامل البريد الإلكتروني

معالجة البريد الإلكتروني الوارد

يعالج النظام رسائل البريد الإلكتروني الواردة للدعم:

1. **استلام البريد الإلكتروني**: يصل البريد الإلكتروني إلى عنوان الدعم
2. **تحليل البريد الإلكتروني**: استخراج المرسل، الموضوع، الجسم، المرفقات
3. **تحديد العميل**: مطابقة بريد المرسل مع جهة الاتصال

4. **إنشاء أو تحديث**: إنشاء تذكرة جديدة أو إضافة إلى سلسلة موجودة
5. **إخطار الموظفين**: تنبيه الموظف المعين أو الفريق
6. **إرسال تأكيد**: تأكيد الاستلام للعميل

رؤوس البريد الإلكتروني المستخدمة:

- `From`: عنوان البريد الإلكتروني للعميل
- `Subject`: عنوان التذكرة (مع رقم التذكرة إذا كانت ردًا)
- `In-Reply-To`: تتبع السلسلة للردود
- `Message-ID`: معرف الرسالة الفريد

الإشعارات الصادرة

تحديثات الموظفين تؤدي إلى إشعارات بريد إلكتروني إلى:

- **العميل**: إذا تم وضع علامة على التحديث بأنه مرئي للعميل
- **المشتركين**: جميع الموظفين وجهات الاتصال المشتركين
- **المعين**: عندما يتم تعيين التذكرة أو إعادة تعيينها

متغيرات قالب البريد الإلكتروني:

- `{{ticket_number}}`: مرجع التذكرة
- `{{customer_name}}`: اسم العميل
- `{{update_text}}`: محتوى التحديث
- `{{staff_name}}`: اسم الموظف الذي ��ضاف التحديث
- `{{ticket_url}}`: رابط مباشر للتذكرة

الوصول إلى بوابة العملاء

يمكن للعملاء عرض تذاكرهم من خلال بوابة الرعاية الذاتية:

1. **تسجيل الدخول**: المصادقة على بوابة العملاء
2. **التنقل**: الذهاب إلى "الدعم" أو "تذاكري"
3. **عرض التذاكر**: رؤية جميع تذاكرهم وحالتها
4. **عرض التفاصيل**: قراءة تاريخ التذكرة الكامل (التحديثات المرئية للعملاء فقط)
5. **الرد**: إضافة ردود على التذاكر المفتوحة

6. **المرفقات**: عرض وتنزيل المرفقات
7. **الحالة**: رؤية الحالة الحالية للتذكرة والأولوية

قيود عرض العملاء:

- يرى العملاء تذاكرهم فقط
- التحديثات الداخلية فقط مخفية
- المرفقات الخاصة بالموظفين غير متاحة
- التغييرات في النظام (التعيينات، الحالة) ملخصة

المشتركين والإشعارات

الاشتراك التلقائي

يتم الاشتراك تلقائيًا عند:

- **إنشاء تذكرة**: منشئ التذكرة
- **التعيين إلى التذكرة**: المعين الحالي
- **الذكر** @username: الموظف المذكور في التحديثات باستخدام صيغة

الاشتراك اليدوي

إضافة مشتركين:

1. افتح التذكرة
2. انقر على قسم **المشتركين**
3. انقر على **إضافة مشترك**
4. اختر مستخدمًا أو جهة اتصال
5. انقر على **إضافة**

إزالة المشتركين:

1. انقر على X بجانب اسم المشترك
2. أكد الإزالة
3. لن يتلقوا إشعارات بعد الآن

تفضيلات الإشعارات

يتلقى الموظفون إشعارات عبر قنوات متعددة:

- **إشعارات داخل التطبيق**: إشعارات في الوقت الحقيقي في قائمة الإشعارات المنسدلة
- **إشعارات البريد الإلكتروني**: إشعارات بريد إلكتروني قابلة للتكوين لتحديثات التذاكر
- **إشعارات @الإشارة**: تنبيهات فورية عند ذكرهم في تحديثات التذاكر
- **التكرار**: فوري، ملخص كل ساعة، ملخص يومي
- **الأنواع**: الأحداث التي تؤدي إلى إشعارات (تذاكر جديدة، تحديثات، تعيينات، إش��رات)

البحث والتصفية

البحث السريع

يبحث شريط البحث العالمي في:

- عناوين التذاكر ووصفها
- أرقام التذاكر
- أسماء العملاء
- محتوى التحديث
- أسماء المرفقات

التصفية المتقدمة

تصفية التذاكر حسب:

- **الحالة**: مفتوحة، قيد التقدم، تم الحل، إلخ.
- **الأولوية**: عاجلة، عالية، متوسطة، منخفضة
- **المعين إلى**: موظف محدد أو غير معين
- **العميل**: عميل محدد
- **الخدمة**: خدمة محددة
- **العلامات**: علامة واحدة أو أكثر
- **نطاق التاريخ**: تم إنشاؤها أو تحديثها ضمن تواريخ

- **تحتوي على مرفقات**: تذاكر تحتوي/لا تحتوي على ملفات

الفلاتر المحفوظة

إنشاء وحفظ مجموعات فلاتر مخصصة:

1. **تطبيق الفلاتر**: تعيين الفلاتر المطلوبة
2. **حفظ الفلتر**: انقر على زر "حفظ الفلتر"
3. **تسمية الفلتر**: أعطه اسمًا وصفيًا
4. **الوصول السريع**: يظهر في قائمة الفلاتر المنسدلة

الفلاتر المحفوظة الشائعة:

- تذاكري المفتوحة": المعينة لي، الحالة مفتوحة أو قيد التقدم"
- قائمة العاجل": الأولوية عاجلة، الحالة مفتوحة"
- تحتاج إلى رد": الحالة معلقة، تنتظر العميل"
- مغلقة مؤخرًا": مغلقة في آخر 7 أيام"

التقارير والتحليلات

مقاييس التذاكر

تتبع أداء الدعم:

- **إجمالي التذاكر**: تم إنشاؤها، تم حلها، مغلقة
- **متوسط وقت الحل**: الوقت المستغرق للحل حسب الأولوية
- **وقت الاستجابة الأول**: الوقت حتى أول تحديث من الموظف
- **التذاكر حسب الحالة**: توزيع عبر الحالات
- **التذاكر حسب الأولوية**: عاجلة مقابل عالية مقابل متوسطة مقابل منخفضة
- **التذاكر حسب الموظف**: عبء العمل لكل موظف
- **التذاكر حسب العلامة**: فئات المشكلات الشائعة

الالتزام بـ SLA

مراقبة الالتزام باتفاقية مستوى الخدمة:

- **ضمن SLA**: التذاكر التي تم حلها ضمن الهدف
- **تم انتهاك SLA**: التذاكر الت�� تجاوزت الأوقات المستهدفة
- **في خطر** SLA: التذاكر التي تقترب من موعد النهائي
- **متوسط الاستجابة**: الأوقات الفعلية مقابل الأوقات المستهدفة

رؤى العملاء

تحليل أنماط دعم العملاء:

- **أعلى منشئي التذاكر**: العملاء الذين لديهم أكبر عدد من التذاكر
- **مشكلات متكررة**: العملاء الذين لديهم مشاكل متكررة
- **متعلقة بالخدمة**: تذاكر حسب نوع الخدمة
- **معدل نجاح الحل**: معدل الحل من أول اتصال

التكامل مع الخدمات

ربط الخدمات بالتذاكر

عندما تتعلق تذكرة بخدمة محددة:

1. **اختيار الخدمة**: اختر من خدمات العميل
2. **سياق الخدمة**: عرض تفاصيل الخدمة في الشريط الجانبي
3. **إجراءات سريعة**: إدارة الخدمة من التذكرة
4. **تاريخ الخدمة**: رؤية التذاكر والتغييرات ذات الصلة

إجراءات الخدمة السريعة

من الشريط الجانبي للتذكرة:

الإضافات:

1. انقر على زر **إضافة**
2. تفتح نافذة الإضافات
3. اختر الحزم لإضافتها
4. أكد وطبق

5. يتم تحديث الخدمة تلقائيًا

حذف الخدمة:

1. انقر على زر **حذف**
2. تظهر نافذة تأكيد
3. أكد الحذف
4. تتم إزالة الخدمة
5. يتم تحديث التذكرة بالإجراء

تعديل الخدمة:

1. انقر على زر **تعديل الخدمة**
2. تفتح نافذة الخدمة الكاملة
3. قم بإجراء تغييرات على تكوين الخدمة
4. احفظ التغييرات
5. عد إلى التذكرة

أفضل الممارسات

إنشاء التذاكر

1. **عناوين وصفية**: ملخص واضح وموجز للمشكلة
2. **وصف مفصل**: تضمين جميع المعلومات ذات الصلة
3. **الأولوية الصحيحة**: تعيينها بناءً على التأثير الفعلي
4. **ربط الخدمة**: دائمًا ربطها بالخدمة المتأثرة إذا كان ذلك ممكنًا
5. **إضافة علامات**: استخدام علامات متسقة للتصنيف

إدارة التذاكر

1. "**تحديثات في الوقت المناسب**: تحديث التذاكر بانتظام، حتى لو كان "لا تقدم
2. **دقة الحالة**: الحفاظ على الحالة الحالية (لا تترك التذاكر المحلولة مفتوحة)
3. **تواصل مع العملاء**: إبقاء العملاء على اطلاع بالتقدم
4. **ملاحظات داخلية**: توثيق خطوات استكشاف الأخطاء داخليًا
5. **توثيق الحلول**: بيان بوضوح ما الذي أصلح المشكلة

نقل المعرفة

1. **ربط الأسئلة الشائعة**: الإشارة إلى أو إنشاء أسئلة شائعة للمشكلات الشائعة
2. **توثيق الحلول**: إضافة تفاصيل الحلول للرجوع إليها في المستقبل
3. **توسيم بشكل مناسب**: يجعل من السهل العثور على التذاكر المماثلة
4. **مشاركة الدروس المستفادة**: مناقشة الحلول المعقدة مع الفريق

تجربة العملاء

1. **الاتصال الأول**: الرد بسرعة للاعتراف بالاستلام
2. **تحديد التوقعات**: إخبار العميل متى يتوقع الحل
3. **نبرة احترافية**: الحفاظ على المجاملة والاحترافية
4. **لغة واضحة**: تجنب المصطلحات التقنية مع العملاء
5. **تأكيد الحل**: التأكد من رضا العميل قبل الإغلاق

تكامل نظام الإشعارات:

- إضافة دعم الإشعارات داخل التطبيق لأحداث التذاكر
- وظيفة @الإشارة لإخطار موظفين محددين
- التكامل مع واجهة برمجة التطبيقات للإشعارات للأنظمة الخارجية
- تسليم الإشعارات في الوقت الحقيقي

التعاون في الوقت الحقيقي:

- تتبع المشاهدين النشطين - رؤية من يقوم حاليًا بعرض كل تذكرة
- طوابع زمنية لآخر نشاط للتذاكر
- تحديثات في الوقت الحقيقي عندما يعمل عدة مستخدمين على نفس التذكرة

تحسين التواصل:

- الإشارات تشترك تلقائيًا المستخدمين المذكورين@
- تحسين تفضيلات الإشعارات مع قنوات متعددة
- تحسين التكامل بين البريد الإلكتروني والإشعارات داخل التطبيق

انظر أيضًا: واجهة برمجة تطبيقات الإشعارات للحصول على تفاصيل حول إرسال الإشعارات من الأنظمة الخارجية.

مرجع API

إنشاء تذكرة

```
POST /crm/ticket/
```

الإذن المطلوب: `create_ticket`

الحصول على تذكرة

```
GET /crm/ticket/ticket_id/{ticket_id}
```

الإذن المطلوب: `view_ticket`

تحديث تذكرة

```
PATCH /crm/ticket/ticket_id/{ticket_id}
```

الإذن المطلوب: `update_ticket`

حذف تذكرة

```
DELETE /crm/ticket/ticket_id/{ticket_id}
```

الإذن المطلوب: `delete_ticket`

إضافة تحديث تذكرة

```
POST /crm/ticket/ticket_id/{ticket_id}/updates
```

تحميل مرفق

```
POST /crm/ticket/ticket_id/{ticket_id}/attachments
```

الإذن المطلوب: `upload_attachment`

الاشتراك في تذكرة

```
POST /crm/ticket/ticket_id/{ticket_id}/subscribe
```

الإذن المطلوب: `subscribe_ticket`

استكشاف الأخطاء وإصلاحها

المشكلة: عدم إنشاء التذاكر من البريد الإلكتروني

- الحل: تحقق من تكوين تكامل البريد الإلكتروني
- الحل: تحقق من أن بريد المرسل يتطابق مع جهة اتصال العميل
- الحل: مراجعة سجلات معالجة البريد الإلكتروني

المشكلة: عدم استلام العميل للإشعار��ت

- الحل: تحقق من أن البريد الإلكتروني للاتصال صحيح
- "الحل: تحقق من أن التحديث محدد بأنه "مرئي للعميل
- الحل: تأكد من عدم حظر البريد الإلكتروني/البريد العشوائي

المشكلة: لا يمكن تحميل المرفقات

- الحل: تحقق من حجم الملف (حد أقصى 10 ميغابايت)
- الحل: تحقق من أن نوع الملف مسموح به
- الحل: تحقق من أذونات دليل التحميلات

المشكلة: إجراءات الخدمة السريعة لا تعمل

- الحل: تحقق من أن الخدمة مرتبطة بشكل صحيح بالتذكرة

- الحل: تحقق من أن المستخدم لديه الأذونات المطلوبة للخدمة
- الحل: تأكد من تحميل بيانات الخدمة بنجاح