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 <rbac>` لأذونات إدارة 2FA، `Authentication Flows <authentication_flows>` لتفاصيل عملية تسجيل الدخول.

الغرض

توفر 2FA:

1. **أمان معزز** — يحمي الحسابات حتى لو تم اختراق كلمات المرور.
2. **امتثال** — يلبي متطلبات الأمان للصناعات المنظمة.
3. **اختيار المستخدم** — اختياري للمستخدمين، يمكن فرضه حسب الدور أو عالميًا.
4. **معيار صناعي** — يستخدم بروتوكول TOTP المتوافق مع Google Authenticator وMicrosoft Authenticator وAuthy وغيرها من التطبيقات القياسية.

كيفية عمل 2FA

عند تمكين 2FA لمستخدم:

1. **الإعداد** — يقوم المستخدم بمسح رمز QR باستخدام تطبيق المصادقة الخاص به أثناء التسجيل.
2. **تسجيل الدخول** — بعد إدخال اسم المستخدم/كلمة المرور، يقدم المستخدم الرمز المكون من 6 أرقام من تطبيقه.
3. **التحقق** — يتحقق النظام من أن الرمز المعتمد على الوقت يتطابق مع القيمة المتوقعة.

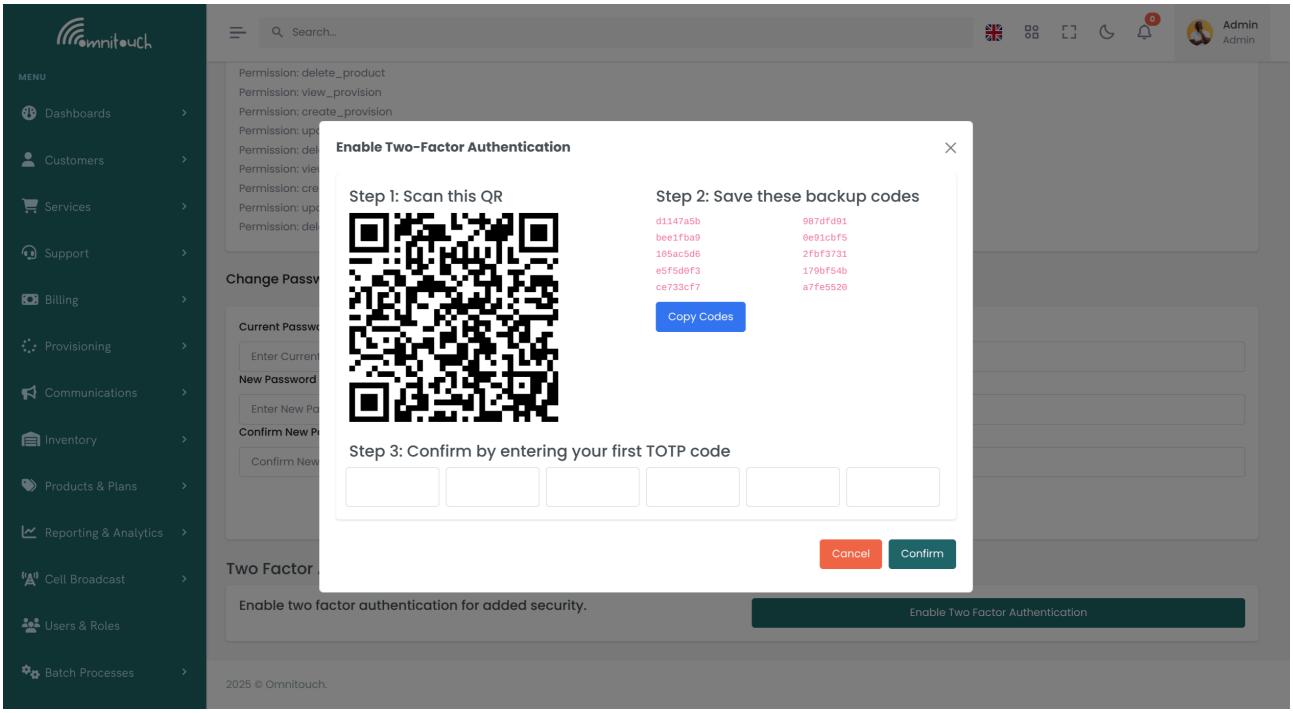
4. يحصل المستخدم على الوصول فقط بعد التحقق من كلا العاملين — **الوصول**.

تمكين 2FA

للمستخدمين الفرديين

يمكن للمستخدمين تمكين 2FA لحساباتهم الخاصة:

1. انتقل إلى **الملف الشخصي** أو **إعدادات المستخدم**
2. اختر **تمكين 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

عند تمكين 2FA، يتلقى المستخدمون رموز نسخ احتياطي يمكن استخدامها إذا كان جهازهم غير متاح:

- كل رمز يستخدم مرة واحدة
- قم بتخزين الرموز بشكل آمن 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 للمستخدم يدويًا عن طريق مسح حقل 2FAلمسؤول لديه وصول إلى قاعدة البيانات إعادة تعيين totp_secret 2 في قاعدة البيانات. يمكن للمستخدم بعد ذلك إعادة تمكين 2FA.

API تكامل

2 لمستخدم تمكينFA

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

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

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

2 تحقق من إعدادFA

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

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

2 أثناء تسجيل الدخول تحقق منFA

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

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

ترجع رمز الوصول، ورمز التحديث، وبيانات المستخدم عند التحقق الناجح.

إعادة إنشاء رموز النسخ الاحتياطي

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

يتطلب المصادقة. ترجع 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 متوافق مع TOTP (Google Authenticator وAuthy وMicrosoft Authenticator 1وPassword، إلخ).

ماذا أفعل إذا فقدت هاتفي؟ استخدم رمز النسخ الاحتياطي أو اتصل بمسؤول 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 لأدوار الموظفين الإداريين والدعم، ولكنها اختيارية لمستخدمي العملاء. لا يفرض النظام 2FA على حسابات العملاء (2FA لا يرى مستخدمو دور العملاء مطالبات تسجيل 2).

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

إدارة مفاتيح API

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

ملاحظة

للحصول على مفاهيم المصادقة العامة على API، راجع `concepts_api` وأمثلة الاستخدام.

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.

الوصول إلى إدارة ��فاتيح API

انتقل إلى:

أو مباشرةً:

الإذن المطلوب: `MANAGE_API_KEYS` (دور المسؤول)

عرض قائمة مفاتيح API

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 انتهاء الصلاحية** - متى سينتهي المفتاح تلقائيًا
- **الإجراءات** - أزرار تعديل، حذف، وإعادة إنشاء

مثال على العرض:

أدوات لوحة التحكم

في أعلى الصفحة، يتم عرض إحصائيات ملخصة:

- **إجمالي مفاتيح API** - عدد جميع مفاتيح API (نشطة وغير نشطة)
- **المفاتيح النشطة** - المفاتيح الصالحة حاليًا
- **ستنتهي قريبًا** - المفاتيح التي ستنته�� في الثلاثين 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 انتهاء الصلاحية: [اختيار التاريخ]
> (اختياري - اتركه فارغًا لعدم انتهاء الصلاحية)
>
> الأذونات: ☐ عرض العملاء ☐ إنشاء العملاء ☐ عرض الخدمات ☐ إنشاء الخدمات ☐ التزويد
> ☐ عرض المخزون ☑ مسؤول (جميع الأذونات)
>
> [توليد المفتاح] [إلغاء]

إرشادات الحقول:

الاسم (مطلوب)

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

الوصف (اختياري)

- شرح أكثر تفصيلاً
- أمثلة: "يستخدم بواسطة خادم تزويد Ansible"، "تزامن فواتير الطرف الثالث"
- يساعد المسؤولين في المستقبل على فهم غرض المفتاح

تاريخ انتهاء الصلاحية (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 الصلاحية لأغراض الأمان (90 يومًا إلى سنة واحدة)

الأذونات

- اختر أذونات محددة أو تحقق من "مسؤول" للحصول على وصول كامل
- يتبع نفس نموذج الأذونات المعتمد على الدور مثل حسابات المستخدمين
- **أفضل ممارسة:** تعيين الحد الأدنى من الأذونات اللازمة

الخطوة 3: توليد ونسخ المفتاح

بعد النقر على **"توليد 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]

تحذير

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 توليده**، مما يلغي المفتاح القديم وقد يتسبب في كسر 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
- الطابع الزمني للإنشاء
- اسم مستخدم المنشئ

- الأذونات المرتبطة
- إحصائيات الاستخدام (إذا تم تنفيذها)
- سجلات الوصول الأخيرة

إعادة توليد مفتاح API

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

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

تحذير

إعادة التوليد تلغي المفتاح القديم على الفور.

ستتوقف أي تطبيقات تستخدم المفتاح القديم عن العمل. قم بتحديث جميع التكاملات بالمفتاح الجديد قبل إعادة التوليد.

ماذا يحدث:

- يتم إلغاء المفتاح القديم
- يتم توليد مفتاح جديد بنفس الأذونات
- يتم عرض المفتاح الجديد (انسخه على الفور)
- الاسم، الوصف، والأذونات تبقى 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 التي تستخدم 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

تعديل تفاصيل مفتاح API

لتعديل تفاصيل مفتاح API:

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

الحقول القابلة للتعديل:

- الاسم
- الوصف
- تاريخ الانتهاء
- الأذونات

غير القابلة للتعديل:

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

حالة مفتا API

عدة حالات API يمكن أن تكون لمفاتيح:

نشط

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

ستنتهي قريبًا

- نشط ولكنه سينتهي في الثلاثين يومًا القادمة
- يظهر بشارة برتقالية/تحذيرية
- يُفضل تدويره قبل انتهاء الصلاحية

منتهي

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

ملغى

- تم حذفه/تعطيله يدويًا
- غير صالح بشكل دائم
- لم يعد يظهر في القائمة النشطة

التصفية والبحث

تدعم قائمة مفاتيح API:

البحث:

ابحث بالاسم، الوصف، أو جزء من المفتاح:

تصفية حسب الحالة:

قائمة منسدلة للتصفية لعرض:

- جميع المفاتيح

- المفاتيح النشطة فقط
- المفاتيح التي ستنتهي قريبًا (في الثلاثين يومًا القادمة)
- لمفاتيح المنتهية��

الفرز:

انقر على رؤوس الأعمدة للفرز حسب:

- الاسم
- تاريخ الإنشاء
- تاريخ انتهاء الصلاحية
- أنشأ بواسطة

أفضل ممارسات الأمان

تولید مفتاح API

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

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 بإدخال المفاتيح في التحكم في الإصدارات

- قم بتد��ير المفاتيح بشكل دوري (90-365 يومًا)

إدارة الأذونات

- **مبدأ أقل امتياز** - امنح الأذونات الضرورية فقط
- تجنب إنشاء مفاتيح المسؤول ما لم يكن ذلك ضروريًا للغاية
- استخدم مفاتيح منفصلة للأنظمة/الأغراض المختلفة
- راجع الأذونات بانتظام

المراقبة والتدقيق

- راقب استخدام مفتاح API عبر سجلات النشاط
- قم بإعداد تنبيهات لأنماط الوصول غير العادية
- راجع الطوابع الزمنية "آخر استخدام" بانتظام
- أزل المفاتيح غير المستخدمة

تدوير المفاتيح

أنشئ 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 الأذونات المطلوبة
  - تأكد من استخدام رأس `X-API-KEY` (ليس `Authorization`)

بعد الإنشاء "API لم يتم العثور على مفتاح"

- **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 عن المشكلة للمسؤول

مفتاح API سينتهي قريبًا

- **السبب:** تاريخ انتهاء الصلاحية يقترب (في غضون 30 يومًا)
- **الإصلاح:**
  - أنشئ مفتاحًا جديدًا مع تمديد انتهاء الصلاحية
  - قم بتحديث التطبيقات لاستخدام المفتاح الجديد
  - ألغِ المفتاح القديم بعد الترحيل

مفتاح API لا يمكن حذف

- **السبب:** قد يكون محميًا أو قيد الاستخدام
- **الإصلاح:**
  - تأكد من أن لدي�� 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": "تزامن فواتير الطرف الثالث",
  "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 → انتقل إلى مفاتيح API
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 `crm_config.yaml` الخاص بـ Ansible

حالة الاستخدام 2: تكامل الفواتير للطرف الثالث

إنشاء مفتاح للقراءة فقط لتصدير الفواتير:

1. الاسم: "تزامن 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 Grafana في مصدر بيانات

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 الخلفية للبوابة

الوثائق ذات الصلة

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

سمات العملاء

سمات العملاء هي أزواج مرنة من المفتاح والقيمة يمكن إرفاقها بأي سجل عميل لتخزين بيانات وصفية مخصصة، أو إعدادات تكوين، أو معلومات محددة للأعمال لا تناسب الحقول القياسية للعملاء.

للتصنيف البصري للعملاء والروابط القابلة للنقر، راجع `Customer Tags <administration_tags>` . لمعلومات العملاء الأساسية، راجع `Customers, Contacts, Sites & Services <basics_customers>` .

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، أو معلمات التكامل، أو علامات منطق الأعمال المخصصة.

الغرض وحالات الاستخدام

تشمل حالات الاستخدام الشائعة لسمات العملاء:

1. بيانات التكامل

تخزين معرفات الأنظمة الخارجية أو مفاتيح API المحددة لهذا 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 العلامات أو الإعدادات التي تتحكم في سلوك محدد للعميل:

- `billing_method` = "quarterly" (تجاوز الفوترة الشهرية الافتراضية)
- `auto_provision` = "true" (تمكين التوفير التلقائي للخدمات)
- `support_tier` = "premium" (مستوى دعم مخصص)
- `credit_limit` = "10000" (حد ائتماني محدد للعميل)

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 الاتصال)
- `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 الخاصة بالتكامل
- تصميم ميزات جديدة قبل إضافة حقول قاعدة البيانات
- لا تحتاج البيانات إلى استعلامات معقدة أو عمليات انضمام

استخدم الحقول القياسية عندما:

- تكون البيانات أساسية لنموذج العميل (الاسم، البريد الإلكتروني، العنوان)

- يتطلب الأمر البحث المتكرر أو التصفية أو التقارير
- تحتوي البيانات على قيود تكامل مرجعي
- 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 **Attributes**
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 **Attributes**
3. انقر على زر **Add Attribute**
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 **Create Attribute**

اتفاقيات التسمية:

- استخدم الحروف الصغيرة مع الشرطات السفلية: `external_system_id` ✓
- تجنب المسافات: `external system id` ✗

- ا��عل الأسماء وصفية ولكن مختصرة
- استخدم تسمية متسقة عبر العملاء لنفس أنواع السمات

تحرير سمة

لتحرير سمة موجودة:

1. انتقل إلى صفحة نظرة عامة العميل
2. انقر على علامة **Attributes**
3. ابحث عن السمة التي تريد تحريرها في الجدول
4. انقر على زر **Edit** (قلم)
5. عدل اسم السمة أو القيمة
6. انقر على **Update Attribute**

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 **Attributes**
3. ابحث عن السمة التي تريد حذفها في الجدول
4. انقر على زر **Delete** (سلة المهملات)
5. أكد الحذف في النافذة المنبثقة

تحذير

حذف السمات المستخدمة من قبل التكاملات، أو سير العمل الخاص بالتوفير، أو منطق الفوترة قد يتسبب في حدوث أخطاء. 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/`

الإذن المطلوب: `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}`

الإذن المطلوب: `update_customer_attribute`

جسم الطلب:

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

الحصول على سمة بواسطة ID

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

الإذن المطلوب: `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 بواسطة العميل ID

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

الإذن المطلوب: `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}`

الإذن المطلوب: `delete_customer_attribute`

الاستجابة:


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


عمليات السمات بالجملة

إدارة سمات متعددة

لتعيين سمات متعددة لعميل مرة واحدة (مثل، 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 بواسطة السمة

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

```
# 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 معينة
    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 القائمة على السمات، ضع في اعتبارك إضافة حقول قاعدة بيانات مفهرسة أو تنفيذ نقطة بحث مخصصة.

أفضل الممارسات

1. اتفاقيات التسمية

- استخدم snake_case: `external_system_id` ✓
- كن وصفياً: `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 البوليانية، 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 التط��يق قبل الحفظ
- استخدم 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 بتدقيق 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 م��ين:

```
# تعيين الفوترة ربع السنوية
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

راجع `rbac` لتكوين التحكم في الوصول المستند إلى الأدوار.

استكشاف الأخطاء وإصلاحها

السمة غير ظاهرة في واجهة المستخدم

- تحقق من أنه تم إنشاء السمة (تحقق 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 بتدقيق تبعيات السمة قبل الحذف

تم اقتطاع قيمة السمة

- 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 البصري، وأفضل الممارسات للنشر.

نظرة عامة

تستخدم OmniCRM نظامين رئيسيين للتكوين:

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:

- **الملف:** `OmniCRM-UI/.env`
- **التنسيق:** متغيرات البيئة
- **يتطلب:** إعادة بناء واجهة المستخدم بعد التغييرات
- **يستخدم لـ:** العلامة التجارية، الميزات، الخدمات الخارجية

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 نظام التحكم في الإصدارات. استخدم تكوينات محددة للبيئة أو إدارة الأسرار.

ملاحظة: يجب أن يتطابق هذا مع بيانات اعتماد قاعدة البيانات في `.env`. على النشر المعتمد على الحاويات، يكون الخادم عادةً `db` (اسم خدمة Docker).

أنواع الخدمات

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

الغرض: يحدد أنواع الخدمات الصالحة لنشرك.

تستخدم هذه الأنواع في جميع أنحاء النظام لـ:

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

أضف أنواع خدمات مخصصة هنا لحالات الاستخدام الخاصة بك.

تكامل HSS (خادم المشترك المنزلي)

للمشغلين في الشبكات المحمولة مع تكامل HSS:

```
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 معرفات APN مفصولة بفواصل المتاحة للتزويد

يستخدم لـ: تزويد الشبكات المحمولة ومصادقة المشتركين.

تكوين قالب البريد الإلكتروني Mailjet

تستخدم OmniCRM Mailjet للبريد الإلكتروني المعاملاتي. كل نوع بريد إلكتروني له 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"

  # التحقق 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"
```

إنشاء قوالب Mailjet:

1. سجل الدخول إلى لوحة معلومات Mailjet (<https://app.mailjet.com>)
2. انتقل إلى **Transactional** → **Templates**
3. أنشئ قالبًا جديدًا أو استنسخ قالبًا موجودًا
4. لاحظ **معرف القالب** (قيمة عددية)
5. أضف 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 التحقق من البريد الإلكتروني
- `{{balance}}` - الرصيد الحالي للحساب
- `{{expiry_date}}` - تاريخ انتهاء الرصيد أو الخدمة

تكوين التزويد

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

الغرض:

- `failure_list` - عناوين البريد الإلكتروني التي يتم إبلاغها عندما يفشل تزويد Ansible
- تتضمن الإشعارات اسم ��لبرنامج النصي، تفاصيل الخطأ، ومعلومات العميل
- يسمح لفريق العمليات بالاستجابة بسرعة لمشكلات التزويد

تكوين الفاتورة

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

الغرض:

يحدد قالب HTML Jinja2 الذي سيتم استخدامه لإنشاء فاتورة PDF.

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

الفاتورة. أدناه للحصول على تفاصيل حول إنشاء قوالب مخصصة **PDF** انظر قسم **إنشاء**

عنوان URL الأساسي لـ CRM

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

الغرض:

- النصية لإجراء استدعاءات API يستخدمه برامج Ansible
- يستخدم في قوالب البريد الإلكتروني لإنشاء روابط إلى CRM
- الخاص بك (ليس أسماء الحاويات API المتاح للجمهور لـ URL يجب أن يكون عنوان الداخلية)

أمثلة:

- التطوير: `http://localhost:5000`
- الإنتاج: `https://api.yourcompany.com`
- Docker: `http://omnicrm-api:5000` (الاتصال الداخلي للحاويات)

مهم:

- لا تقم بتضمين شرطة مائلة في النها��ة
- المتاح للجمهور إذا كانت البرامج النصية تعمل على خوادم مختلفة URL استخدم عنوان
- تحديث عند النشر في الإنتاج

تكوين OCS و CGRates

```
ocs:
  ocsApi: 'http://10.179.2.142:8080/api'
  ocsTenant: 'mnc380.mcc313.3gppnetwork.org'
  cgrates: 'localhost:2080'
```

التكوين:

- `ocsApi` - نقطة نهاية API لـ OCS لإدارة المشتركين

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

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

تكوين SMSC (بوابة SMS)

```
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` - المصادقة لـ SMSC API

تكوين بث الخلايا

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

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

انظر `features_cell_broadcast` للحصول على تفاصيل الاستخدام.

مفتاح السر JWT

```
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 سري فريد
- لا تشارك أو تلتزم بنظام التحكم في الإصدارات

إدارة الأسرار التلقائية (نشر Ansible):

عند النشر باستخدام Ansible، يتم إدارة مفتاح JWT تلقائيًا:

1. **النشر الأول:** يقوم Ansible بإنشاء سلسلة عشوائية من 64 حرفًا (256 بت من الانتروبيا)
2. **النشرات اللاحقة:** يحتفظ Ansible بالمفتاح الموجود من الخادم
3. **يحصل كل خادم على مفتاح فريد خاص به** - لا تعمل الجلسات عبر البيئات

هذا يضمن:

- عدم وجود أسرار مشفرة في مستودعات git
- كل بيئة م��زولة تشفيرياً
- تستمر الأسرار عبر النشرات (لا يتم إبطال الجلسات عند إعادة النشر)
- إعادة بناء الخادم يولد مفتاحًا جديدًا (تُبطل الجلسات بشكل صحيح)

توليد السر اليدوي (نشرات غير Ansible):

إذا كنت تنشر بدون Ansible، يجب عليك توليد مفتاح 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 عبر 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 ائتمان العملاء

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 على التعرف على الرسوم
- يستخدم أيضًا في أسماء ملفات PDF للفواتير (مثل `YOURCOMPANY_12345.pdf`)

للحصول على تفاصيل `<integrations_payment_vendors>` انظر `تكاملات بائعي الدفع` الإعداد.

مفاتيح API وقائمة IP البيضاء

حدد مفاتيح API وقيود IP مع وصول قائم على الأدوار:

```yaml
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 البيضاء (بدون مفتاح، مستقلة API)
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
- الأدوار المخصصة المحددة في نظام RBAC

أفضل ممارسات الأمان:

- استخدم مفاتيح API (حد أدنى 32 حرفًا) طويلة وعشوائية
- قيد IPs إلى المصادر المعروفة فقط
- منح الحد الأدنى من الأدوار الضرورية
- بانتظام API تدوير مفاتيح
- راقب استخدام مفاتيح API في السجلات

تحذير أمني:

- استخدم قائمة IP البيضاء فقط للشبكات الداخلية الموثوقة
- يجب ألا تستخدم IPs المحلية (127.0.0.1، ::1)
- استخدم مفاتيح API بدلاً من ذلك للوصول الخارجي
- اعتبر قواعد جدار الحماية كحماية إضافية

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

تكوين الواجهة الأمامية (.env)

يتم تكوين واجهة React باستخدام متغيرات البيئة في `OmniCRM-UI/.env`.

الموقع: `/OmniCRM/OmniCRM-UI/.env`

مفاتيح API والخدمات الخارجية

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

# بوابة الدفع Stripe
VITE_STRIPE_PUBLISHABLE_KEY=pk_test_xxxxx

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

يجب أن تتطابق:

يجب أن تتطابق `VITE_STRIPE_PUBLISHABLE_KEY` مع `stripe.publishable_key` في `crm_config.yaml`.

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 (لون العلامة التجارية الرئيسي)
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` - لون العلامة التجارية الرئيسي (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` - النصوص الداكنة (#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
# النظام: النظام (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 تجميع الخطوط مع التطبيق
- **متوافقة مع الحديقة 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 تبعيات خارجية
- **الأمان** - ا 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:

- النظام - 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. إلى **أضف ملفات الخط** `OmniCRM-UI/src/assets/fonts/your-font-name/`
    - استخدم تنسيق WOFF2 للحصول على أفضل ضغط ودعم المتصفح
    - تضمين أوزان متعددة (300، 400، 500، 600، 700) للتقديم الصحيح
    - تسمية 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 والنصوص القوية) غامق

روابط سريعة لتطبيق الويب

قم بتكوين ما يصل إلى 6 تطبيقات ويب سريعة الوصول التي تظهر في لوحة معلومات الإدارة:

```
# 1 تطبيق الويب: 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"

# 2 تطبيق الويب: 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
# (تكوين 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 في `/grafana` لإنشاء لوحات معلومات مخصصة، ويمكن تضمين تلك اللوحات في واجهة OmniCRM.

```
# تكامل لوحة معلومات Grafana
# معرفات اللوحات (مفصولة بفواصل) - احصل عليها 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 لوحة 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 تكوينها، 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`، فإن:

- 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 تعريفه
- 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
```

روابط لموارد الدعم الخارجية تظهر في واجهة المستخدم **الغرض:**.

تسجيل الدخول الاجتماعي

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

الخيارات:

- `yes` - تمكين أزرار تسجيل الدخول الاجتماعي (Google، Facebook، إلخ.)
- `no` - تعطيل تسجيل الدخول الاجتماعي

يجب تكوين موفري تسجيل الدخول الاجتماعي بشكل منفصل **ملاحظة:**.

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
```

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

التنسيقات المدعومة:

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

كيف يعمل:

يكتشف النظام تلقائيًا خدمة التتبع 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:**

   Google Analytics 4:

   - سجل الدخول إلى Google Analytics
   - أنشئ أو اختر خاصية GA4
   - انتقل إلى **الإدارة** → **تدفقات البيانات**
   - اختر تدفق الويب الخاص بك
   - انسخ **معرف القياس** (التنسيق: `G-XXXXXXXXXX`)

   Google Tag Manager:

   - سجل الدخول إلى Google Tag Manager
   - أنشئ أو اختر حاوية
   - انسخ **معرف الحاوية** (التنسيق: `GTM-XXXXXXX`)

2. **أضف إلى ملف .env:**

```
VITE_GOOGLE_TAG_ID=G-XXXXXXXXXX
```

3. **أعد 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 منطق الدفع
};
```

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 للحوار، علامات التبويب، إلخ.)
trackPageView('/virtual/page-path', 'Page Title');
```

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 للاختيار
- **CCPA (كاليفورنيا)** - السماح للمستخدمين بالانسحاب من جمع البيانات
- **سياسة الخصوصية** - افصح عن استخدام Google Analytics في سياسة الخصوصية الخاصة بك
- **موافقة ملفات تعريف الارتباط** - تنفيذ لافتة موافقة ملفات تعريف الارتباط إذا كانت مطلوبة بموجب القوانين المحلية
- **احتفاظ البيانات** - تكوين فترات احتفاظ بالبيانات المناسبة في إعدادات Google Analytics

استكشاف الأخطاء وإصلاحها:

المشكلة	السبب المحتمل	الحل
ا توجد بيانات في�� لوحة معلومات GA	معرف التتبع غير صحيح	تحقق من أن تنسيق المعرف يتطابق مع G-XXXXXXXXXX أو GTM-XXXXXXX
أخطاء في وحدة التحكم عند تحميل الصفحة	معرف التتبع غير صالح	تحقق من الأخطاء المطبعية في ملف .env
فشل اتصال قاعدة البيانات	بيانات اعتماد خاطئة أو الخادم غير قابل للوصول	تحقق من أن قاعدة البيانات تعمل، وأن بيانات الاعتماد صحيحة

اعتبارات الأداء:

- يتم تحميل برامج التتبع بشكل غير متزامن ولا تعيق تحميل الصفحة
- تأثير الأداء 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`
- خطاف تتبع الصفحة: `OmniCRM-UI/src/utils/usePageTracking.js`
- التهيئة: تلقائية عند بدء التطبيق في `App.js` أو `index.js`

تكوين ال��حن 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 هي الدولار الأمريكي، يدفع العملاء 10 دولارات أمريكية لكل يوم من الخدمة.

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

العلامة التجارية والتخصيص البصري

تسمح لك OmniCRM باستبدال الصور الافتراضية للعلامة التجارية بشعار شركتك وشاشات البداية دون تعديل التعليمات البرمجية.

ملفات الشعار ونظام التراجع

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

الشعارات الافتراضية (موجودة دائمًا):

- `DefaultLogoDark.png` - شعار الوضع الداكن (يستخدم على الخلفيات الفاتحة)
- `DefaultLogoLight.png` - شعار الوضع الفاتح (يستخدم على الخلفيات الداكنة)

الشعارات المخصصة (اختيارية، تأخذ الأولوية عند وجودها):

- `logoSm.png` - شعار صغير لشريط الجانب المنهار (100 بكسل x100 موصى به)
- `logoDark.png` - شعار داكن بالحجم الكامل للرؤوس (200 بكسل x50 موصى به)
- `logoLight.png` - شعار فاتح بالحجم الكامل لشاشات المصادقة (موصى به: 200x50 بكسل)

كيف يعمل تراجع الشعار:

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، العروض الصغيرة للرؤوس
- **logoDark.png** - شريط الرأس الرئيسي (الوضع الفاتح)، رأس لوحة معلومات الإدارة
- **logoLight.png** - شاشات تسجيل الدخول/التسجيل، الخلفيات الداكنة، الدوار المصادقة

استبدال الشعارات:

1. **أنشئ ملفات الشعار الخاصة بك:**
    - استخدم تنسيق PNG لدعم الشفافية
    - تطابق الأبعاد الموصى بها أعلاه
    - تأكد من وضوح الشعارات عند الدقة العادية وعالية الدقة
2. **أضف 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. **أعد 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 الجانبي 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 بخلفيات شفافة
- **عالية الدقة** - 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
<!-- ![شعار الشركة](../absolute/path/to/your-logo.png) -->
        <h1>الفاتورة</h1>
    </div>

    <div class="invoice-details">
        <p><strong>رقم الفاتورة:</strong> {{ invoice_number }}</p>
        <p><strong>التاريخ:</strong> {{ date }}</p>
        <p><strong>تاريخ الاستحقاق:</strong> {{ due_date }}</p>
        <p><strong>فترة الفوترة:</strong> {{ start_date }} إلى {{
end_date }}</p>
    </div>

    <div class="customer-details">
        <h3>الفاتورة إلى:</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>الوصف</th>
                <th>التاريخ</th>
                <th>المبلغ</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>إجمالي المبلغ المستحق: ${{ "%.2f"|format(total_amount)
}}</p>
    </div>

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

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

أفضل ممارسات القالب:

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

إنشاء قالب فاتورة مخصص

1. **انسخ قالب المثال:**

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

2. **تحرير القالب:**
   - استبدل اسم الشركة، الشعار، معلومات الاتصال
   - ضبط التنسيق (الألوان، الخطوط، التخطيط) ليتناسب مع العلامة التجارية
   - إضافة أو إزالة الأقسام حسب الحاجة (تفاصيل الضرائب، تعليمات الدفع، إلخ.)

3. **تحديث التكوين:**

   تحرير `crm_config.yaml`:

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

4. **اختبار إنشاء الفاتورة:**
   - أنشئ فاتورة اختبار في CRM
   - قم بتنزيل PDF وتحقق من التنسيق
   - تحقق من `invoice_templates/rendered/{invoice_id}.html` لأخطاء التقديم

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

   إذا قمت بتغيير القالب وترغب في إعادة إنشاء الفواتير الحالية:

```
-- امسح جميع ملفات PDF المخزنة مؤقتًا (يؤدي إلى إعادة التوليد)
DELETE FROM Invoice_PDF_Cache;
```

نظام التخزين المؤقت PDF

لتحسين الأداء، تقوم OmniCRM بتخزين PDF التي تم إنشاؤها مؤقتًا:

سلوك التخزين المؤقت:

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

تخزين التخزين المؤقت:

- يتم تخزين PDF المشفرة بتنسيق Base64 في جدول `Invoice_PDF_Cache`
- تجزئة المحتوى SHA256 للتحقق من سلامة البيانات
- يتضمن اسم الملف، تاريخ الإنشاء، تاريخ آخر وصول

إدارة التخزين المؤقت اليدوي:

```
# 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:

إنشاء/تنزيل PDF الفاتورة:

GET /invoice/pdf/{invoice_id}

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

رؤوس التخزين المؤقت:

- الطلب الأول: استجابة أبطأ (وقت التوليد)

- الطلبات المخزنة مؤقتًا: استجابة فورية
- يعتبر نجاح/فشل التخزين المؤقت 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:

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

أفضل ممارسات التكوين

الأمان

- **لا تقم أبداً بإدخال الأسرار** - استخدم `.gitignore` لملفات التكوين التي تحتوي 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 قوائم IP البيضاء وقواعد جدار الحماية
- **استخدم تكوينات محددة للبيئة** - فصل تكوينات التطوير/الاختبار/الإنتاج
- **تقييد أذونات مفتاح API** - منح الحد الأدنى من الأدوار الضرورية
- **استخدم قائمة IP البيضاء بحذر** - يفضل استخدام مفاتيح API لأمان أفضل

الصيانة

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

الأداء

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

العلامة التجارية

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

استكشاف الأخطاء وإصلاحها

المشكلات الشائعة

التغييرات لم تُطبق

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

أخطاء في بناء جملة YAML

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

فشل اتصال قاعدة البيانات

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

مدفوعات Stripe لا تعمل

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

البريد الإلكتروني لا يُرسل

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

فشل إنشاء PDF:

- تحقق من تثبيت WeasyPrint: `pip install weasyprint`
- تحقق من تطابق اسم القالب 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 (PNG و JPEG تعمل بشكل أفضل)
- تحقق من أن مسارات الصور لا تحتوي على أحرف خاصة

مشكلات التنسيق:

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

التخزين المؤقت لا يتم إبطاله:

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

قائمة التحقق من ا��تكوين

استخدم هذه القائمة عند نشر OmniCRM:

تكوين الخلفية

- [ ] انسخ `.env.example` إلى `.env`
- [ ] تعيين كلمات مرور قوية لقاعدة البيانات
- [ ] تكوين بيانات اعتماد CGRates
- [ ] تحديث `crm_config.yaml` بإعداداتك:
  - [ ] اتصال قاعدة البيانات
  - [ ] أنواع الخدمات
  - [ ] مفاتيح Mailjet API ومعرفات القوالب
  - [ ] عناوين البريد الإلكتروني لإخطار فشل التزويد
  - [ ] اسم ملف قالب الفاتورة
  - [ ] عنوان URL الأساسي لـ CRM (متاح للجمهور)

- [ ] نقاط نهاية OCS/CGRates
- [ ] تكوين SMSC
- [ ] سري جديد JWT توليد مفتاح
- [ ] مفاتيح Stripe (حقيقية، ليست تجريبية)
- [ ] البيضاء IP وقائمة API مفاتيح

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 إنشاء PDF الفاتورة

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

الأمان

- [ ] تغيير جميع كلمات المرور الافتراضية
- [ ] توليد JWT سري فريد
- [ ] استخدام مفاتيح Stripe (ليس مفاتيح الاختبار) الإنتاجية
- [ ] تدوير مفاتيح Mailjet API
- [ ] تمكين قواعد جدار الحماية
- [ ] تكوين قائمة IP البيضاء 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 العملاء
- [ ] اختبار تزويد الخدمة من البداية إلى النهاية
- [ ] التحقق من إرسال إشعارات البريد الإلكتروني بشكل صحيح
- [ ] اختبار إنشاء الفاتورة وتنزيل PDF
- [ ] التحقق من معالجة الدفع (Stripe)
- [ ] التحقق من مصادقة المستخدم و2FA
- [ ] اختبار الانتحال وتسجيل التدقيق
- [ ] التحقق من مزامنة بيانات الاستخدام من OCS
- [ ] اختبار إنشاء خطة العمل وتجديدها
- [ ] تأكيد تخصيص المخزون بشكل صحيح

النشر

- [ ] بناء صور Docker أو نشرها على الخوادم
- [ ] بدء حاويات قاعدة البيانات (MySQL، PostgreSQL)
- [ ] بدء CGRates

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

الوثائق ذات الصلة

- `RBAC وإدارة المستخدمين </rbac>`
- `المنتجات والخدمات </concepts_products_and_services>`
- `تزويد Ansible </concepts_ansible>`
- `إدارة المخزون </administration_inventory>`
- `فواتير العملاء </payments_invoices>`
- `2/> المصادقة الثنائية fa>`
- `رعاية العملاء والانتحال </customer_care>`
- `تكاملات بائعي الدفع <integrations_payment_vendors>`
- `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 **المخزون** في قبل مشغلي الشبكات والعملاء.

هذا يعني أنه يمكننا تتبع جميع أنواع العناصر، مثل المودمات، أرقام الهواتف، كتل عناوين IP، أو حتى الأجهزة المادية مثل GPON ONTs أو CPEs اللاسلكية الثابتة.

لتخزين بيانات `Customer Attributes <administration_attributes>` انظر أيضًا: التعريف المخصصة، و `Customer Tags <administration_tags>` للتصنيف المرئي.

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، ولأولئك الذين يعملون في شبكة لاسلكية ثابتة، يمكنن CPEs المنتشرة في الميدان مع مواقعها تتبع.

ملاحظة

ترتبط عناصر المخزون بالمنتجات أثناء الت Provisioning من خلال حقل `inventory_items_list`. للحصول على شرح كامل حول كيفية تكامل المخزون مع توفير المنتجات، بما في ذلك واجهة مستخدم اختيار المخزون وتكامل كتاب Ansible، انظر `Complete Product Lifecycle Guide - Inventory Requirements <guide_product_lifecycle>`.

الغرض

يخدم مخزون OmniCRM عدة أغراض رئيسية:

1. **توفير الخدمات**: عندما يسجل عميل للحصول على خدمة، قد تحتاج عناصر مثل المودمات، بطاقات SIM، أو أرقام الهواتف إلى تخصيصها. 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 معلومات حول بصمة الشبكة، مثل لاسلكي ثابت تم نشره، 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 قالب مخزون لبطاقات SIM من Omnitouch. تم طلب دفعة من 1000 بطاقة SIM (إذا لم يكن موجودًا بالفعل) ويتم تحميل بطاقات SIM في حالة النقل المطلوبة.

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

للعميل ويتم SIM عندما يسجل عميل للحصول على خدمة موبايل في المتجر، يتم تخصيص بطاقة للعميل، وتحديث الحالة إلى قيد SIM تغيير الحالة إلى مخصصة. ثم يتم تعيين عنصر مخزون بطاقة الاستخدام.

كـ تم إلغاء SIM إذا ألغى العميل الخدمة أو أصبحت الخدمة غير نشطة، يتم وضع علامة على بطاقة الخدمة.

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.

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

تشغيل تأهيل الخدمة تلقائيًا ضد OmniCRM عندما يسجل عميل ��لحصول على خدمة، يمكن لـ عنوان العميل، لمعرفة ما إذا كان العنوان في مخزون المنازل المارة وما الخدمات التي يمكن

تقديمها.

قوالب المخزون

تتيح ميزة **InventoryTemplate** إنشاء أي عدد من أنواع العناصر مع حقول محددة مسبقًا. تعمل هذه القوالب كخرائط 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 رئيسي (`itemtext1`) وموقع جغرافي (`itemtext2`).
- **الموارد الافتراضية** مثل كتل عناوين IP، مع معرفات تم تعيينها من خلال القالب.

يحدد كل قالب 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 قا��ب المودم `itemtext1_label` إلى "عنوان MAC" و `itemtext2_label` إلى "رقم تسلسلي".

يمكن للمشغلين تخصيص الحقول لكل نوع عنصر باستخدام **InventoryTemplates**. تسمح هذه القوالب بتصنيف العناصر وإدارتها بطريقة منظمة، مما يضمن الاتساق في كيفية تتبع العناصر.

الربط بالمنتجات:

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

مثال: منتج مع `inventory_items_list: "['SIM Card', 'Mobile Number']"` يتطلب وجود قوالبي مخزون يحملان بالضبط أسماء القوالب "Mobile Number" و "SIM Card". حساسة لحالة الأحرف.

للحصول على تفاصيل كاملة حول كيفية ارتباط قوالب المخزون بتوفير المنتجات، انظر `Product Lifecycle - Inventory Requirements <guide_product_lifecycle>`.

إنشاء قوالب المخزون عبر ��اجهة المستخدم

لإنشاء قالب مخزون جديد:

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

   معلومات أساسية:

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

   معلومات التكلفة (مطلوبة):

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

   ملاحظة

   تعتبر تكاليف الجملة والتجزئة المحددة هنا قيم افتراضية عند إنشاء عناصر مخزون جديدة من هذا القالب. يمكن أن تحتوي عناصر المخزون الفردية على تكاليف مختلفة إذا لزم الأ��ر.

تسميات الحقول:

- **تسمية النص 1 للعنصر** (مطلوبة): تسمية للحقل القابل للتخصيص الأول (تكون افتراضية إلى "رقم الموديل")
  - أمثلة شائعة: "ICCID" لبطاقات SIM، "عنوان MAC" للمودمات، "رقم الهاتف" للأرقام
- **تسمية النص 2 للعنصر** (مطلوبة): تسمية للحقل القابل للتخصيص الثاني (تكون افتراضية إلى "رقم تسلسلي")
  - أمثلة شائعة: "IMSI" لبطاقات SIM، "رقم تسلسلي" للأجهزة، "المنطقة الجغرافية" للأرقام
- **تسميات النص 3-20 للعنصر** (اختياري): تسميات حقول إضافية 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 النوع من المخزون (بوابة العميل)

4. انقر على **حفظ** لإنشاء القالب

تحرير قوالب المخزون

لتحرير قالب موجود:

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

تحذير

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

تحذير

أسماء القوالب المشار إليها في حقول `inventory_items_list` للمنتجات حساسة لحالة الأحرف. سيؤدي إعادة تسمية قالب إلى كسر الرابط بالمنتجات التي تستخدم الاسم القديم.

أمثلة شائعة للقوالب

قالب بطاقة SIM:

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

قالب رقم الهاتف المحمول:

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

قالب CPE اللاسلكية الثابتة:

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

قالب GPON ONT:

- العنصر: "GPON ONT"
- تكلفة الجملة: 45.00
- تكلفة التجزئة: 0.00 (مضمنة مع الخدمة)
- تسمية النص 1 للعنصر: "رقم تسلسلي"
- تسمية النص 2 لـ"عنصر: "عنوان MAC"
- تسمية النص 3 للعنصر: "موقع PON"
- تسمية النص 4 للعنصر: "الموديل"

إنشاء وإدارة عناصر المخزون

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

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

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 وتاريخه Ansible بمجرد تخصيص عنصر المخزون لعميل، يمكن أن تقوم لتعكس التخصيص. هذا يضمن أن 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 عناصر المخزون

تحتوي كل عنصر مخزو�� على معلومات شاملة منظمة في عدة فئات:

معلومات العنصر الأساسية

- **inventory_id** - معرف فريد لعنصر المخزون (يتم إنشاؤه تلقائيًا)
- **item** - نوع العنصر (مثل، "Modem"، "SIM Card" يتطابق مع اسم قالب المخزون، "Phone Number")
- **inventory_template_id** - رابط إلى قالب المخزون الذي يحدد هذا النوع من العنصر
- **customer_id** - إذا تم تخصيصه لعميل، معرف العميل (يمكن أن يكون فارغًا)
- **service_id** - إذا كان مرتبطًا بخدمة معينة، معرف الخدمة (يمكن أن يكون فارغًا)
- **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** - الحالة الحالية لعنصر المخزون (قيم محددة):
  - **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 إصلاح أو التخلص منه
  - **خارج الخدمة** - غير متاح مؤقتًا
  - **مفقود** - لا يمكن تحديد موقع العنصر
  - **مسروق** - تم سرقة العنصر

حقول العنصر القابلة للتخصيص (من القالب)

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

- **itemtext1** - أول حقل قابل للتخصيص (مطلوب، التسمية محددة بواسطة `itemtext1_label` للقالب)
  - 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 القالب)
    - مثال: itemtext2 للمودمات قد يكون "رقم تسلسلي"
    - مثال: itemtext2 لبطاقا�� SIM قد يكون "IMSI"
    - مثال: itemtext3 للمودمات قد يكون "إصدار البرنامج الثابت"

يحدد كل قالب 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 الفيزيائي (لأجهزة الشبكة والمواقع)

يستخدم لتتبع مواقع النشر الفيزيائية، خاصة لأجهزة الشبكة الثابتة (المودمات، ONTs، CPEs) أو 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/ZIP
- **country** - اسم الدولة

الموقع الجغرافي (يتم ملؤه تلقائيًا من واجهة الويب)

عند إنشاء عناصر المخزون عبر واجهة الويب مع إكمال عنوان، 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 فنيي الميدان

إدارة الأجهزة وروابط الوصول

يوفر حقل `management_url` وصولاً سريعًا إلى واجهات الأجهزة وروابط الت Provisioning:

- **management_url** - رابط الوصول لعنصر المخزون
  - **أجهزة الشبكة**: رابط واجهة الويب (مثل، `https://192.168.1.1` للمودمات، المحولات، 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

  توليد رمز QR

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

  - **عرض تفاصيل عنصر المخزون**: رمز QR 128 بحجم 128x معروض بجانب الرابط
  - **جدول مخزون الخدمة**: رمز QR 64 بحجم 64x يظهر 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 الجهاز دون كتابة عناوين IP
  - **تفعيل eSIM**: يقوم العملاء بمسح رمز QR من CRM لتثبيت ملف تعريف eSIM على جهازهم
  - **خدمة العملاء الذاتية**: توفير وصول سهل إلى تكوين الجهاز أو بوابات العملاء

- **management_username** - اسم المستخدم الإداري للوصول إلى الجهاز

- **management_password** - كلمة مرور الإدارة للوصول إلى ��لجهاز (مشفرة في حالة السكون)

إدارة التكوين

بالنسبة للأجهزة التي تحتوي على ملفات تكوين:

- **config_content** - محتوى ملف التكوين الكامل (يتم تخزينه كنص)
  - مفيد للنسخ الاحتياطي، وإصدار النسخ، والتعافي من الكوارث
  - يمكن تخزين تكوينات المودمات، تكوينات المحولات، 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
  - خصائص 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 استخدام 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 المحمول


```
{
  "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"
}
```


يحتوي على عنوان QR هذا، تعرض واجهة المستخدم رمز eSIM عند عرض عنصر مخزون LPA. هذا باستخدام جهازهم المحمول لتثبيت ملف تعريف QR يقوم العملاء بمسح رمز 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 واجهة ،(مثل الأمثلة 2، 3، و4 أعلاه) `management_url` عند عرض عناصر المخزون مع المستخدم تلقائيًا:

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

حالات الاستخدام:

- **تفعيل eSIM** (المثال 2): يقوم العملاء بمسح رمز QR لتثبيت ملف تعريف eSIM على جهازهم
- **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 الطبيعي عند التProvisioning لعميل
- **مخصص** → **مستخدم** - بعد إلغاء الخدمة، يمكن إعادة استخدام العنصر
- **جديد** → **استخدام داخلي** - مخصص للاختبار، العروض، أو استخدام الموظفين
- **مخصص** → **تالف** - فشل العنصر، يحتاج إلى RMA أو التخلص منه
- **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 (العناصر الجديدة)
- 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 جدًا" أو "مزعج".

لتخزين البيانات الوصفية المنظمة وبيانات المفتاح والقيمة المخصصة، راجع `Customer Attributes <administration_attributes>`.

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

استخدام شائع هو 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 **Tags**
3. سترى قائمة بجميع العلامات النشطة للعميل، تظهر:
    - معاينة العلامة مع اللون المكون
    - نص العلامة
    - تاريخ التفعيل (عندما تصبح العلامة مرئية)
    - تاريخ إلغاء التفعيل (عندما ستختفي العلامة)
    - رابط (إذا تم تكوينه)

إنشاء علامة جديدة

لإنشاء علامة جديدة لعميل:

1. انتقل إلى صفحة نظرة عامة العميل
2. انقر على علامة **Tags**
3. انقر على زر **Add Tag**
4. املأ الحقول المطلوبة:
    - **Tag Text** (مطلوب): النص الذي سيتم عرضه على العلامة
    - **Tag Color** (مطلوب): اختر لونًا باستخدام أداة اختيار اللون أو أدخ�رمز سداسي
    - **Tag Link** (اختياري): URL الذي سيفتح عند النقر على العلامة
    - **Active Date** (مطلوب): التاريخ الذي يجب أن تبدأ فيه العلامة بالظهور
    - **Deactivate Date** (مطلوب): التاريخ الذي يجب أن تتوقف فيه العلامة عن الظهور (الافتراضي هو 01-01-2099)
5. معاينة علامتك في قسم المعاينة
6. انقر على **Create Tag**

تعديل علامة

لتعديل علامة موجودة:

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

حذف علامة

لحذف علامة:

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

مرجع حقل العلامة

تكامل API

يمكن أيضًا إدارة العلامات برمجيًا عبر API:

إنشاء علامة:

```
PUT /crm/tag/
{
  "tag_text": "VIP Customer",
  "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": "Updated Tag Text",
  "tag_hex_color": "FF0000"
}
```

الحصول على العلامات حسب العميل:

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

حذف علامة:

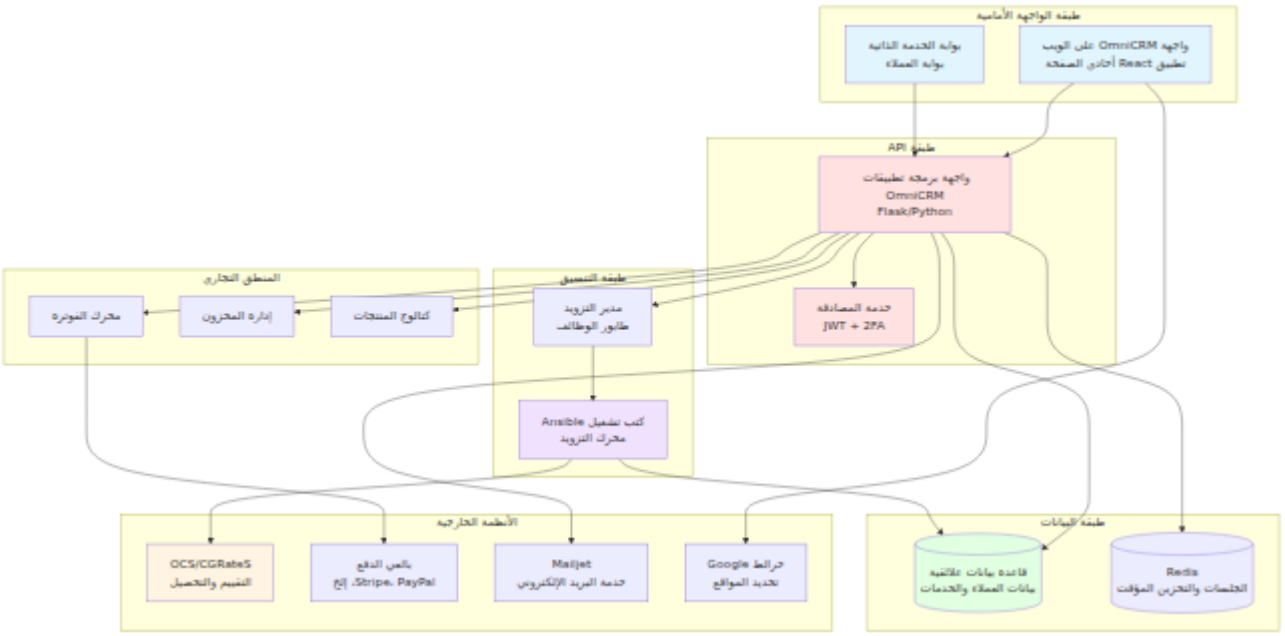
```
DELETE /crm/tag/tag_id/{tag_id}
```

هيكل نظام OmniCRM

يوفر هذا المستند نظرة عامة على هيكل 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 كاملة للخدمات لمزودي الاتصالات.



المكونات الأساسية

1. تطبيقات الواجهة الأمامية

واجهة OmniCRM على الويب

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

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

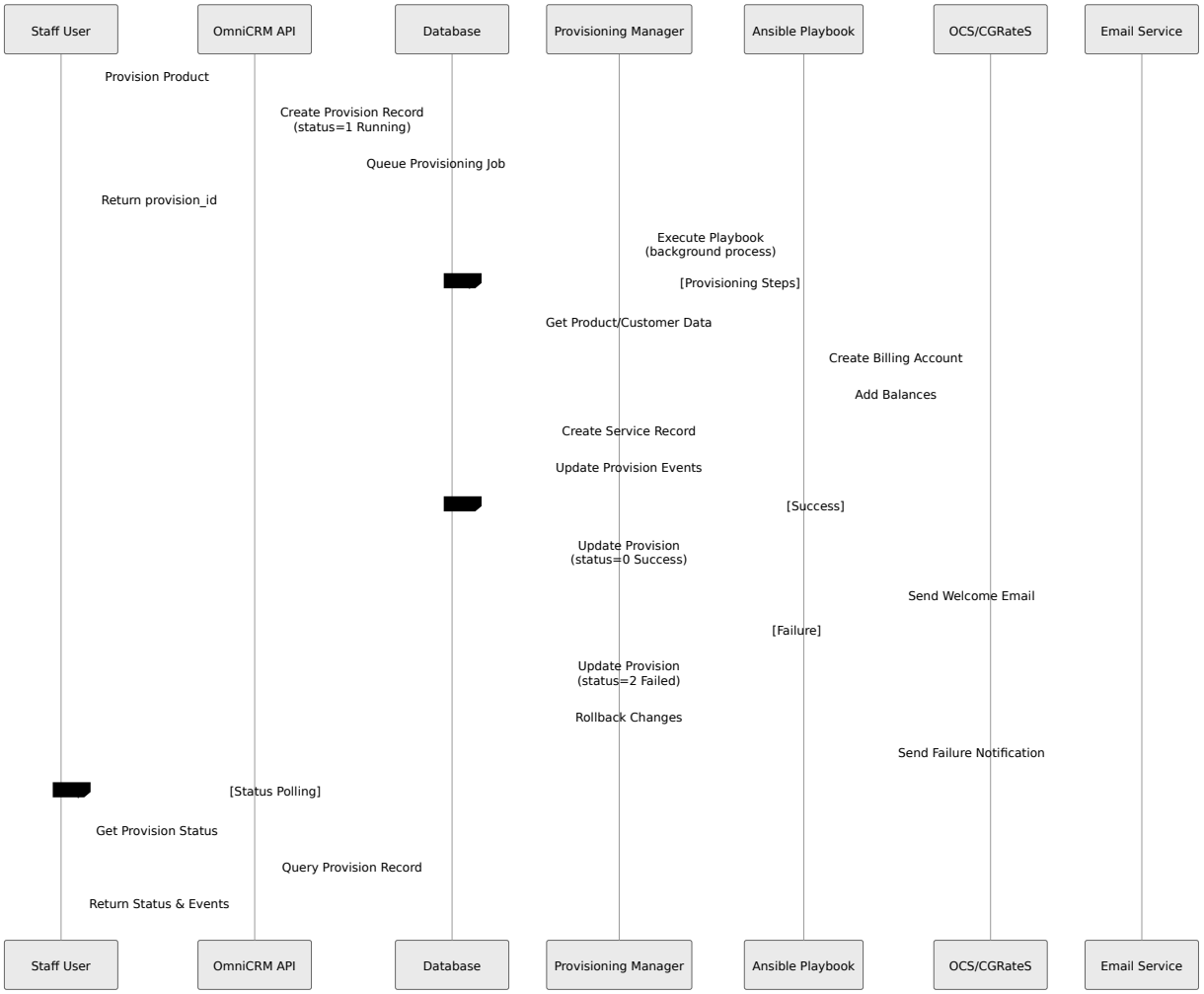
واجهة برمجة تطبيقات OmniCRM (Flask/Python)

- واجهة برمجة تطبيقات RESTful لجميع العمليات
- وثائق OpenAPI/Swagger
- مصادقة قائمة على JWT
- تحديد معدل الطلبات والتخزين المؤقت
- دعم WebSocket للتحديثات في الوقت الفعلي

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

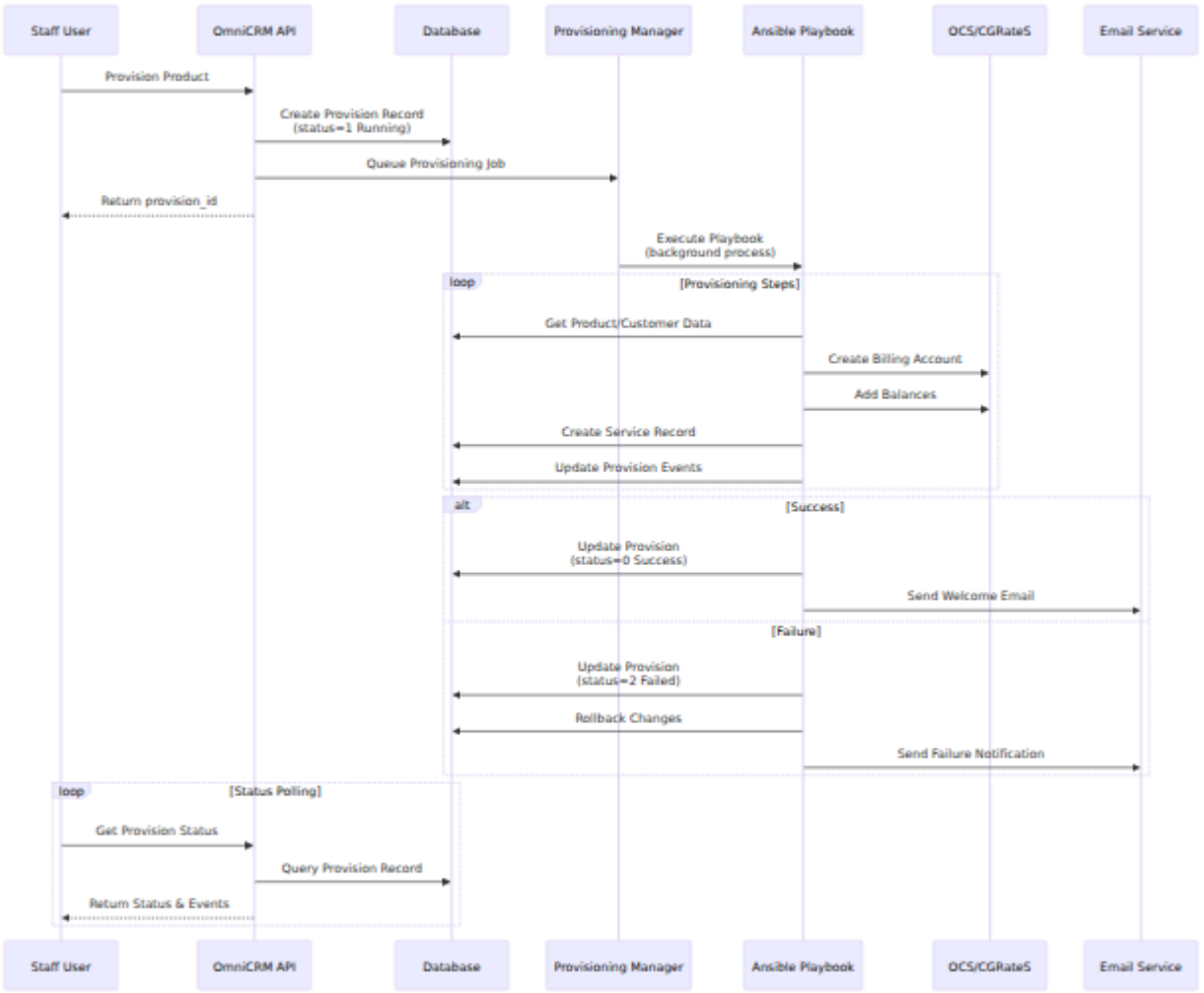


4. الفوترة والتقييم

تكامل OCS/CGRateS

- التحصيل والتقييم في الوقت الف��لي
- إدارة الحسابات
- تتبع الرصيد (صوت، بيانات، نقدي، SMS)
- خطط العمل للرسوم المتكررة
- إشعارات قائمة على العتبات

سير عمل الفوترة:



5. نموذج البيانات

يستخدم OmniCRM قاعدة بيانات علائقية مع النماذج الأساسية التالية. لمخططات العلاقة الكيانية المرئية، راجع العملاء، جهات الاتصال، المواقع والخدمات.

العميل والكيانات ذات الصلة

العميل - الكيان المركزي الذي يمثل شركة أو فرد

الحقل	النوع	الوصف
customer_id	Integer (PK)	معرف فريد
customer_name	String	اسم الشركة أو الفرد
customer_account_type	Enum	'فردي' أو 'تجاري'
customer_status	Enum	'مفتوح'، 'مغلق'، 'معلق'، 'أرشييف'
customer_payment_type	String	'مدفوع مسبقًا' أو 'مدفوع لاحقًا'
customer_enabled	Boolean	هل الحساب نشط
tax_identifier	String	رقم VAT/GST
contract_start_date	DateTime	بداية العقد
contract_end_date	DateTime	نهاية العقد

جهة الاتصال - الأشخاص المرتبطون بالعميل

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

الموقع - مواقع تسليم الخدمة الفعلية

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

نماذج الخدمة والمنتج

الخدمة - حالات الخدمة النشطة

الحقل	النوع	الوصف
service_id	Integer (PK)	معرف فريد
customer_id	Integer (FK)	العميل الرئيسي
product_id	Integer (FK)	قالب المنتج
site_id	Integer (FK)	موقع الخدمة
service_name	String	الاسم المعروض
service_uuid	String	معرف نظام الفوترة
service_status	Enum	الحالة الحالية
service_billed	Boolean	توليد المعاملات
wholesale_cost	Float	تكلفة المزود
retail_cost	Float	سعر العميل
bundled_parent	Integer (FK)	الخدمة الرئيسية للحزم

المنتج - قوالب عروض الخدمة

الحقل	النوع	الوصف
product_id	Integer (PK)	معر�� فريد
product_name	String	الاسم المعروض
product_slug	String	اسم مناسب لـ URL
category	Enum	'مستقل'، 'حزمة'، 'إضافة'، 'عرض'
provisioning_play	String	اسم كتاب تشغيل Ansible
provisioning_json_vars	JSON	متغيرات كتاب التشغيل
inventory_items_list	String	المخزون المطلوب
retail_cost	Float	السعر الشهري
retail_setup_cost	Float	رسوم لمرة واحدة
enabled	Boolean	متاح للبيع

نماذج الفوترة

المعاملة - رسوم/ائتمانات فردية

الحقل	النوع	الوصف
transaction_id	Integer (PK)	معرف فريد
customer_id	Integer (FK)	العميل الرئيسي
invoice_id	Integer (FK)	الفاتورة الرئيسية (اختياري)
service_id	Integer (FK)	الخدمة ذات الصلة
title	String	وصف المعاملة
retail_cost	Float	المبلغ
tax_percentage	Float	معدل الضريبة
tax_amount	Float	الضريبة المحسوبة
void	Boolean	معاملة ملغاة

الفاتورة - معاملات مجمعة للفوترة

الحقل	النوع	الوصف
invoice_id	Integer (PK)	معرف فريد
customer_id	Integer (FK)	العميل الرئيسي
paid	Boolean	تم استلام الدفع
void	Boolean	فاتورة ملغاة
payment_reference	String	معرف معاملة Stripe
start_date, end_date	Date	فترة الفوترة
due_date	Date	الموعد النهائي للدفع
retail_cost	Float	المبلغ الإجمالي

نماذج المخزون

المخزون - الأصول المادية والافتراضية

الحقل	النوع	الوصف
inventory_id	Integer (PK)	معرف فريد
customer_id	Integer (FK)	العميل المعين
service_id	Integer (FK)	الخدمة المرتبطة
inventory_template_id	Integer (FK)	قالب نوع العنصر
item	String	نوع العنصر (بطاقة SIM، إلخ، جهاز توجيه.)
item_state	Enum	'جديد'، 'مخصص'، 'مستخدم'، إلخ.
itemtext1-20	String	حقول مرنة
management_url	String	عنوان إدارة المعدات
config_content	Text	ملف التكوين

Inventory_Template - يحدد هيكل عنصر المخزون

الحقل	النوع	الوصف
inventory_template_id	Integer (PK)	معرف فريد
item	String	اسم القالب
itemtext1_label	String	تسمية لحقل itemtext1
itemtext2_label	String	تسمية لحقل itemtext2

نماذج التزويد

Provision - تتبع وظائف التزويد

الحقل	النوع	الوصف
provision_id	Integer (PK)	معرف فريد
product_id	Integer (FK)	المنتج الذي يتم تزويده
customer_id	Integer (FK)	العميل المستهدف
service_id	Integer (FK)	الخدمة التي تم إنشاؤها/تعديلها
provisioning_play	String	اسم كتاب تشغيل Ansible
provisioning_status	Integer	نجاح، 1=قيد التشغيل، 2=فشل=0

Provision_Event - خطوات التزويد الفردية

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

نماذج المستخدم والأمان

المستخدم - حسابات المستخدمين

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

الدور - أدوار المستخدمين

الحقل	النوع	الوصف
id	Integer (PK)	معرف فريد
name	String	اسم الدور
description	String	وصف الدور

الإذن - أذونات دقيقة

الحقل	النوع	الوصف
id	Integer (PK)	معرف فريد
name	String	اسم الإذن (resource.action)
description	String	وصف الإذن

العلاقات:

- لدى المستخدمين العديد من الأدوار (علاقة متعددة إلى متعددة)
- لدى الأدوار العديد من الأذونات (علاقة متعددة إلى متعد��ة)
- يمكن للمستخدمين الارتباط بجهة اتصال واحدة (للوصول إلى بوابة العملاء)

نقاط التكامل

Stripe بوابة الدفع

- طرق الدفع المرمزة
- تخزين بطاقات متوافق مع PCI
- دفع الفواتير تلقائيًا
- معالجة الاسترداد
- إشعارات انتهاء صلاحية البطاقة

Mailjet خدمة البريد الإلكتروني

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

Google خرائط

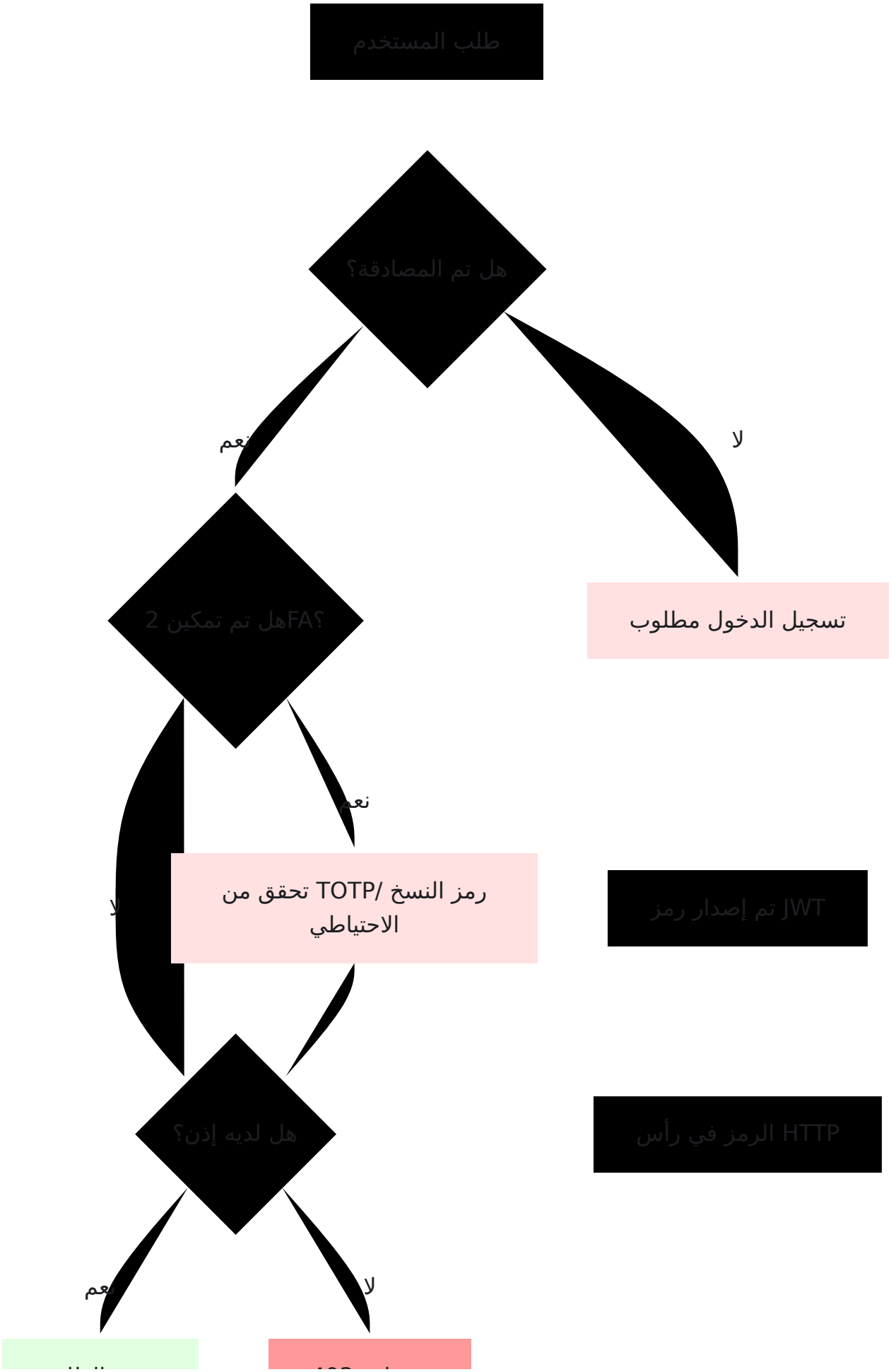
- إكمال العنوان
- تحديد المواقع وعكس تحديد المواقع
- توليد رمز زائد
- رسم خرائط مواقع المواقع

OCS/CGRateS

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

- خطط العمل والجداول الزمنية

هيكل الأمان


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

تنفيذ الطلب

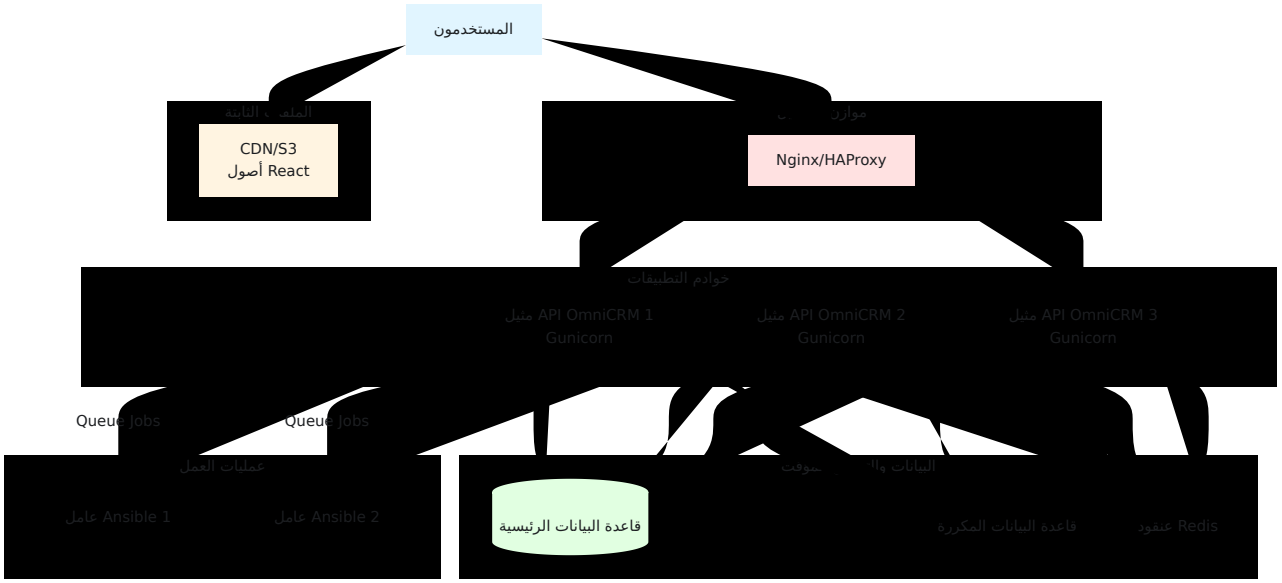
محظور 403

ميزات الأمان:

- مصادقة قائمة على JWT
- مصادقة ثنائية (TOTP)
- التحكم في الوصول بناءً على الأدوار (RBAC)
- الوصول إلى الموارد بناءً على الأذونات
- إدارة الجلسات باستخدام Redis
- تجزئة كلمات المرور (bcrypt)
- التحقق من البريد الإلكتروني
- تسجيل التدقيق عبر سجل النشاط

هيكل النشر

إعداد الإنتاج الموصى به:



مجموعة التكنولوجيا

الجهة الخلفية:

- Python 3.x

- إطار عمل الويب Flask
- SQLAlchemy ORM
- ترحيلات Alembic
- للتشغيل الآلي 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 (تحديد المواقع)

اعتبارات القابلية للتوسع

التوسع الأفقي:

- تصميم API بدون حالة يسمح بوجود عدة مثيلات
- موازن التحميل يوزع الطلبات
- لحالة الجلسة المشتركة Redis

توسع 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 قائمة على Prometheus للحصول لمراقبة جميع جوانب النظام. على تفاصيل كاملة، راجع المراقبة والمقاييس.

المقاييس الرئيسية:

- أوقات استجابة API ومعدلات الطلبات
- 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 (سجل التدقيق)
- تتبع الأخطاء والتنبيهات

نقطة نهاية المقاييس: `/crm/metrics` بتنسيق Prometheus. جميع المقاييس متاحة في راجع المراقبة والمقاييس لتكوين السحب، وقواعد التنبيه، وأمثلة لوحة المعلومات.

الوثائق ذات الصلة

- كتب تشغيل Ansible - التشغيل الآلي للتزويد
- نظام التزويد - تفاصيل سير العمل

- المنتجات والخدمات - هيكل المنتج
- وثائق API - مرجع API
- RBAC - الأمان والأذونات

تدفقات المصادقة والتحكمات الإدارية

يوفر OmniCRM ميزات مصادقة شاملة بما في ذلك تسجيل الدخول، المصادقة الثنائية (2FA)، إدارة كلمات المرور، والتحكمات الإدارية لإدارة أمان المستخدم. يركز هذا الدليل على سير العمل في واجهة المستخدم لكلا من المستخدمين النهائيين والمديرين.

راجع أيضًا: `Self-Care Portal <self_care_portal>` لتسجيل دخول العملاء و الوصول إلى البوابة، `RBAC <rbac>` لأذونات 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** - ا��ترداد كلمة المرور عبر البريد الإلكتروني
- **التحكمات الإدارية** - أدوات إدارية لإعادة تعيين 2FA و كلمات المرور
- **تسجيل الدخول الاجتماعي** - تكامل اختياري مع Google وApple وFacebook (إذا تم تمكينه)
- **التنقل القائم على الدور** - توجيه تلقائي بناءً على أدوار المستخدم

تدفق تسجيل الدخول

تعد 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:

- يتم تسجيل دخول المستخدم على الفور، ويتم توجيهه بناءً على الدور **بدون 2FA:**
  - **العملاء** ← بوابة الخدمة الذاتية (`/self-care`)
  - **الموظفون/المدي��ون** ← لوحة معلومات العملاء (`/customers`)
  - **CBC وضع** ← واجهة بث الخلايا (`/create-cell-broadcast`)
- **مع تمكين 2FA:** يتم إعادة توجيه المستخدم 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 تحدي 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، يتم تسجيل دخول المستخدم وتوجيهه إلى لوحة المعلومات المناسبة

استخدام رمز الاسترداد:

إذا كان تطبيق المصادقة غير متاح:

1. انقر على علامة التبويب **"رمز الاسترداد"**
2. أدخل أحد رموز النسخ الاحتياطي المحفوظة لديك (على سبيل المثال، `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)

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

> [تسجيل الدخول باستخدام Google] [● تسجيل الدخول باستخدام Apple] [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 رابط **"نسيت كلمة المرور؟"** لبدء تدفق إعادة تعيي 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:

مطلوبة فقط للموظفين 2FA. عادةً ما تكون 2FA لا يرى مستخدمو دور العملاء 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 QR

[تأكيد] [إلغاء]

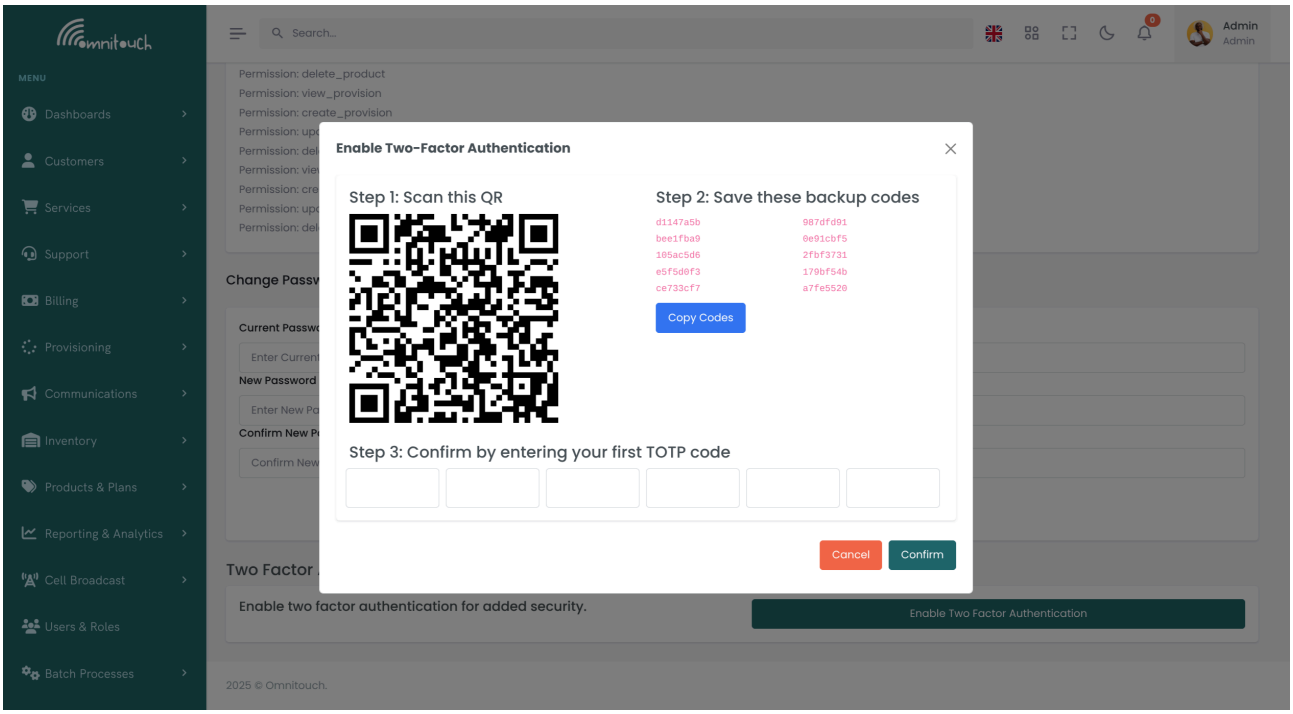
تعليمات الإعداد:

1. **قم بتنزيل تطبيق المصادقة** (إذا لم يكن لديك واحد):
   - iOS: متجر تطبيقات Apple → "Google Authenticator"
   - Android: متجر Google Play → "Google Authenticator"
   - بدائل: Authy، Microsoft Authenticator، 1Password
2. **امسح رمز QR:**
   - افتح تطبيق المصادقة
   - اضغط على "+" أو "إضافة حساب"
   - اختر "امسح رمز QR"
   - وجه الكاميرا نحو رمز QR على الشاشة
   - يضيف التطبيق إدخال "OmniCRM" مع الرمز المكون من 6 أرقام
3. **احفظ رموز النسخ الاحتياطي:**
   - **حرج:** قم بتدوين أو نسخ هذه الرموز الثمانية
   - احتفظ بها في مكان آمن (مدير كلمات المرور، خزنة، إلخ)
   - كل رمز للاستخدام 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 المر��ر (الخدمة الذاتية)

يمكن للمستخدمين الذين ينسون كلمة المرور 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 تنتهي صلاحيته بعد ساعة واحدة)
- تظهر رسالة النجاح: "تم إرسال تعليمات إعادة التعيين إلى بريدك الإلكتروني"

إذا لم يتم العثور على البريد الإلكتروني:

رسالة خطأ: "لم يتم العثور على حساب بهذا البريد الإلكت��وني"

الخطوة 2: تحقق من البريد الإلكتروني

يتلقى المستخدم بريدًا إلكترونيًا بعنوان مثل:

مرحبًا [الاسم]،

لقد طلبت إعادة تعيين كلمة المرور لحسابك في OmniCRM.

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

تنتهي صلاحية هذا الرابط خلال ساعة واحدة.

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 من الطول (عادةً 8+ أحرف)
- يجب أن ت��طابق 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]

التحكمات الإدارية 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... ✏ 🔑 🗑 ✉

أيقونات 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 من قبل)

المسؤول: إعادة تعيين كلمة مرور المستخدم

عندما ينسى المستخدم كلمة المرور ويحتاج المسؤول إلى المساعدة:

الخطوة 1: انقر على أيقونة 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 مؤقتة 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 • سيتم إجبار المستخدم على تغيير كلمة المرور عند تسجيل الدخول التالي

[إغلاق]

إجراء المسؤول:

- نسخ كلمة المرور المؤقتة
- الاتصال بالمستخدم أو التواصل عبر طريقة آمنة
- تقديم كلمة المرور المؤقتة شف��يًا
- توجيه المستخدم لتسجيل 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

للمستخدم FAالمسؤول: إعادة تعيين 2

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

الخطوة 1: انقر على أيقونة إعادة تعيين 2FA ()

تظهر فقط للمستخدمين الذين تم تمكين 2FA لديهم 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 يتطلب رمز 2FA)
- تختفي أيقونة درع 2FA () من صف المستخدم في جدول الإدارة
- يمكن للمستخدم إعادة تمكين 2FA طواعية من إعداداته

ملاحظة أمنية هامة:

يجب على المسؤول، قبل إعادة تعيين 2FA:

1. التحقق من هوية المستخدم بوسائل 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 الدخول من قبل** (`login_count = 0`).

ان��ر على أيقونة إرسال بريد الترحيب (✉)

> إرسال بريد الترحيب إلى:
>
> المستخدم: بوب ويلسون (<bob@example.com>)
>
> سيشمل 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"

رسالة النجاح:

تم إرسال البريد الإلكتروني عبر Mailjet:

يستخدم القالب: `api_crmCommunicationUserWelcome`

المسؤول: تعديل المستخدم

انقر على **أيقونة 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 **أيقونة 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
- احتفظ بالرموز 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 أنماط تسجيل الدخول غير العادية

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 تغيير المستخدم لكلمة المرور عند أول تسجيل دخول
- لا تعيد تعيين كلمات المرور دون داع - استخدم تدفق إعادة تعيين البريد الإلكتروني عندما يكون ذلك ممكنًا

إعادة تعيين 2FA:

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

إدارة المستخدم:

- راجع بانتظام قائمة المستخدمين للحسابات غير النشطة
- أزل المستخدمين الذين غادروا المؤسسة
- تأكد من تعيين الأدوار المناسبة
- راقب المستخدمين الذين لم يسجلوا الدخول من قبل
- قم بتدقيق أذونات المستخدمين ربع سنويًا

استكشاف الأخطاء وإصلاحه

خطأ "البريد الإلكتروني أو كلمة المرور غير صحيحة"

- **السبب:** بيانات الاعتماد غير صحيحة
- **الإصلاح:**
  - تحقق من صحة عنوان البريد الإلكتروني
  - تحقق من إيقاف تشغيل مفتاح Caps Lock
  - حاول إعادة تعيين كلمة المرور إذا نسيتها
  - اتصل بالمسؤول إذا تم قفل الحساب

رمز 2FA غير مقبول

- **السبب:** مشكلة مزامنة الوقت أو رمز غير صحيح
- **الإصلاح:**
  - تأكد من أن وقت الجهاز صحيح (الإعدادات → التاريخ والوقت → تلقائي)
  - انتظر حتى يتم تحديث الرمز (تتغير الرموز كل 30 ثانية)
  - حاول الرمز التالي الذي يظهر
  - استخدم رمز النسخ الاحتياطي إذا لم يعمل تطبيق المصادقة
  - إذا فشلت كل الخيارات 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:** لم يتم إرسال البريد الإلكتروني، فلتر البريد العشوائي، أو البريد الإلكتروني غير صحيح
- **الإصلاح:**
  - تحقق من مجلد البريد العشوائي/غير المرغوب فيه
  - تحقق من صحة عنوان البريد الإلكتروني
  - 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 ساعة واحدة)
- **الإصلاح:**
  - اطلب إعادة تعيين كلمة مرور جديدة
  - تحقق من البريد الإلكتروني وانقر على الرابط بسرعة
  - اتصل بالمسؤول إذا كانت هناك مشكلات متكررة

لا يمكن تمكين 2FA (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

فقدت 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
  - تسجيل الدخول باستخدام كلمة المرور فقط بعد إعادة التعيين
  - إعادة تمكين 2FA واحفظ رموز النسخ الاحتياطي هذه المرة

خطأ "فشل في إعادة تعيين 2FA" المسؤول:

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

المسؤول: لم يتم إنشاء كلمة مرور مؤقتة

- **السبب:** خطأ في API أو مشكلة في الأذونات
- **الإصلاح:**
  - قم بتحديث الصفحة وحاول مرة أخرى
  - تحقق من أذونات المسؤول
  - تحقق من سجلات API للخطأ
  - تأكد من أن قاعدة البيانات قابلة للوصول

اعتبارات أمنية

إدارة الجلسات:

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

أمان كلمة المرور:

- يتم تشفير 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 QR على جانب العميل عندما يكون ذلك ممكنًا
- يتم تشفير رموز النسخ الاحتياطي قبل التخزين
- كل رمز نسخ احتياطي للاستخدام 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 لـ
- `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 الخدمة 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** و عدة **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 تكون هي نفسها 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، مكتبًا، أو موقعًا آخر. وهذا يسمح لنا بوجود مواقع متعددة لعميل واحد، على سبيل المثال، عميل لديه مكاتب متعددة، ومعرفة الخدمات المرتبطة بكل موقع.

الخدمة هي شيء نقوم بفوترة 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.

`سجل النشاط` كما أن العملاء لديهم `<csa_activity_log>`، والذي هو سجل لجميع التغييرات `علامات` التي تم إجراؤها، `<administration_tags>`، `سمات` `<administration_attributes>` لتخزين بيانات التعريف المخصصة، `عناصر الجرد` `<administration_inventory>` ومعلومات مالية مثل `المعاملات` `<payments_transaction>`، `الفواتير` `<payments_invoices>` و `طرق الدفع` `<basics_payment>`.

`إضافة خدمة` لذلك العميل، وهي `<csa_add_service>` بمجرد أن ننشئ عميلًا، يمكننا بعد ذلك الشيء الذي نقوم بفوترته.

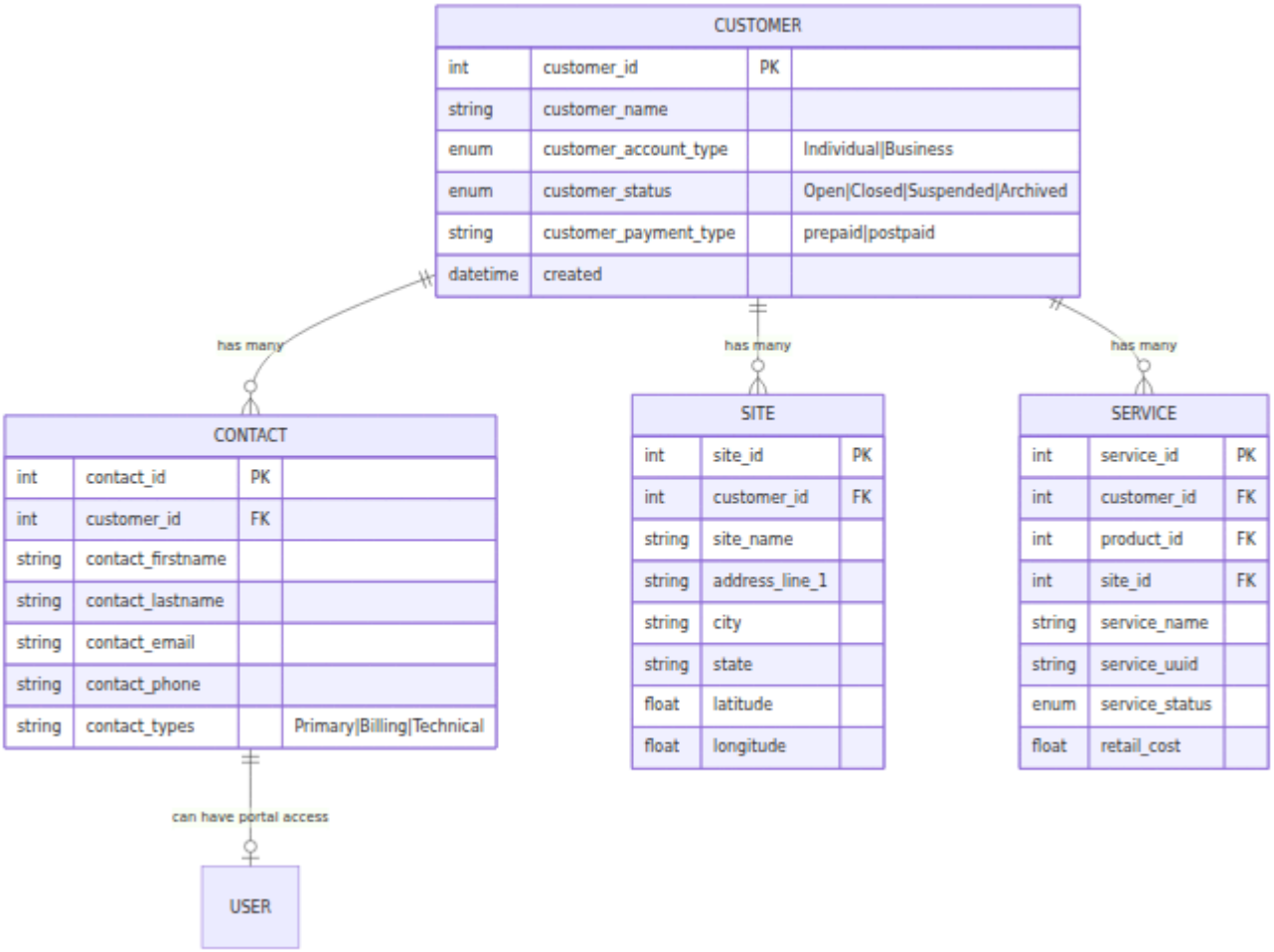
إنشاء عميل للحصول على معلومات حول إنشاء عميل، راجع.

نظرة عامة على نموذج البيانات

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

العلاقات الأساسية للعميل

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

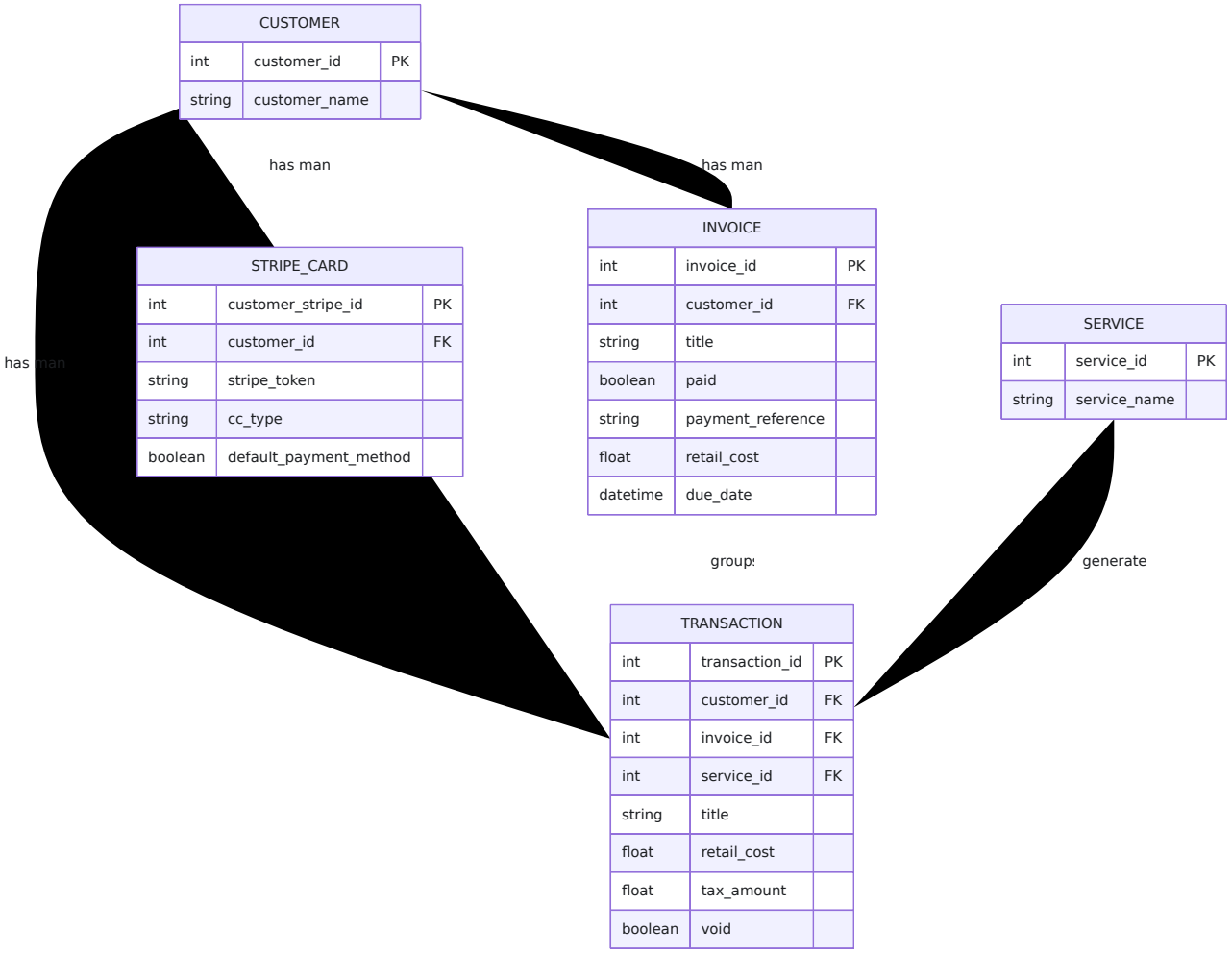


النقاط الرئيسية:

- 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

بيانات الفوترة والمالية

تتبع المعاملات والفواتير جميع الأنشطة المالية.

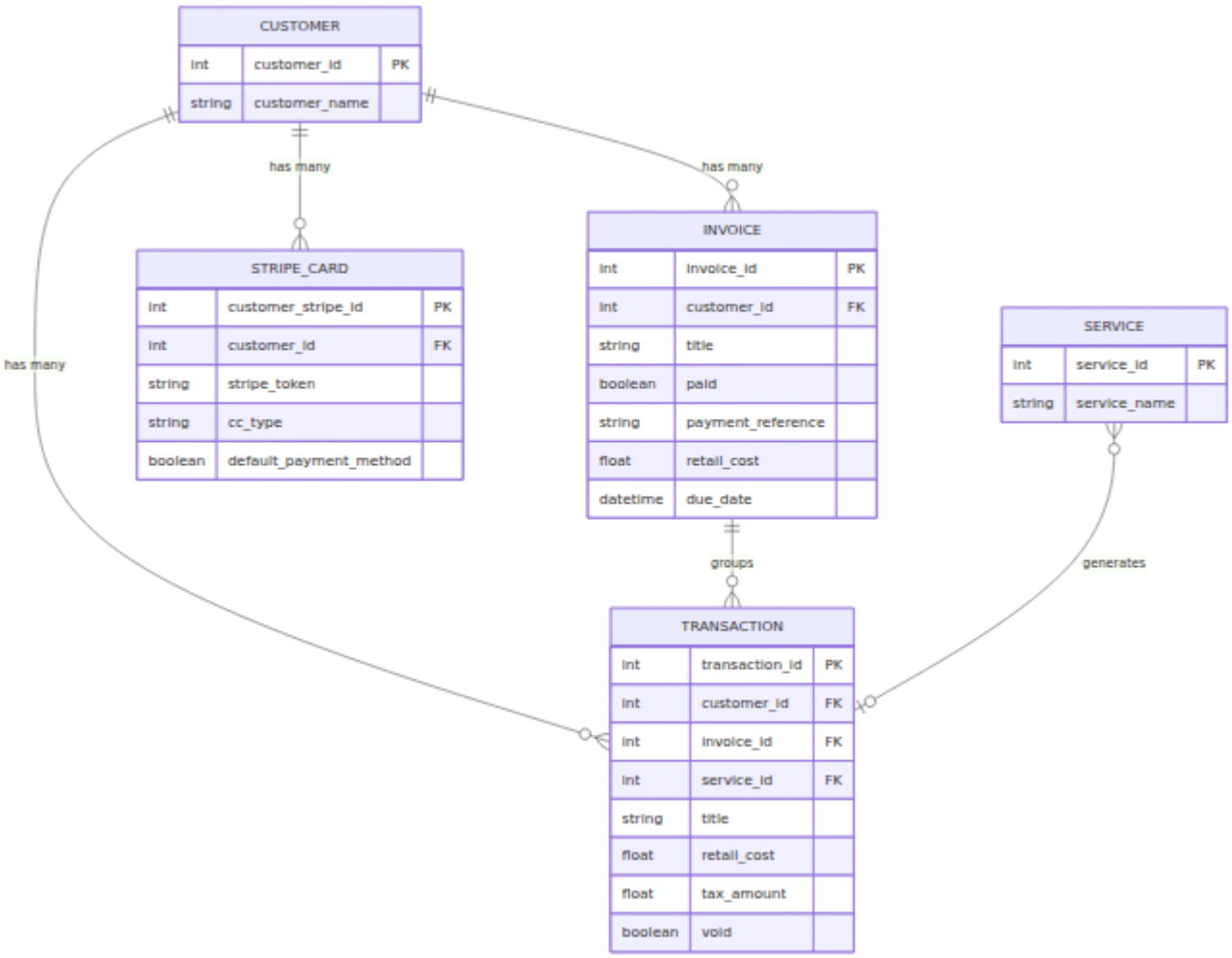


النقاط الرئيسية:

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

المنتج والتوفير

تحدد المنتجات عروض الخدمة؛ التوفير ينشئ خدمات فعلية.

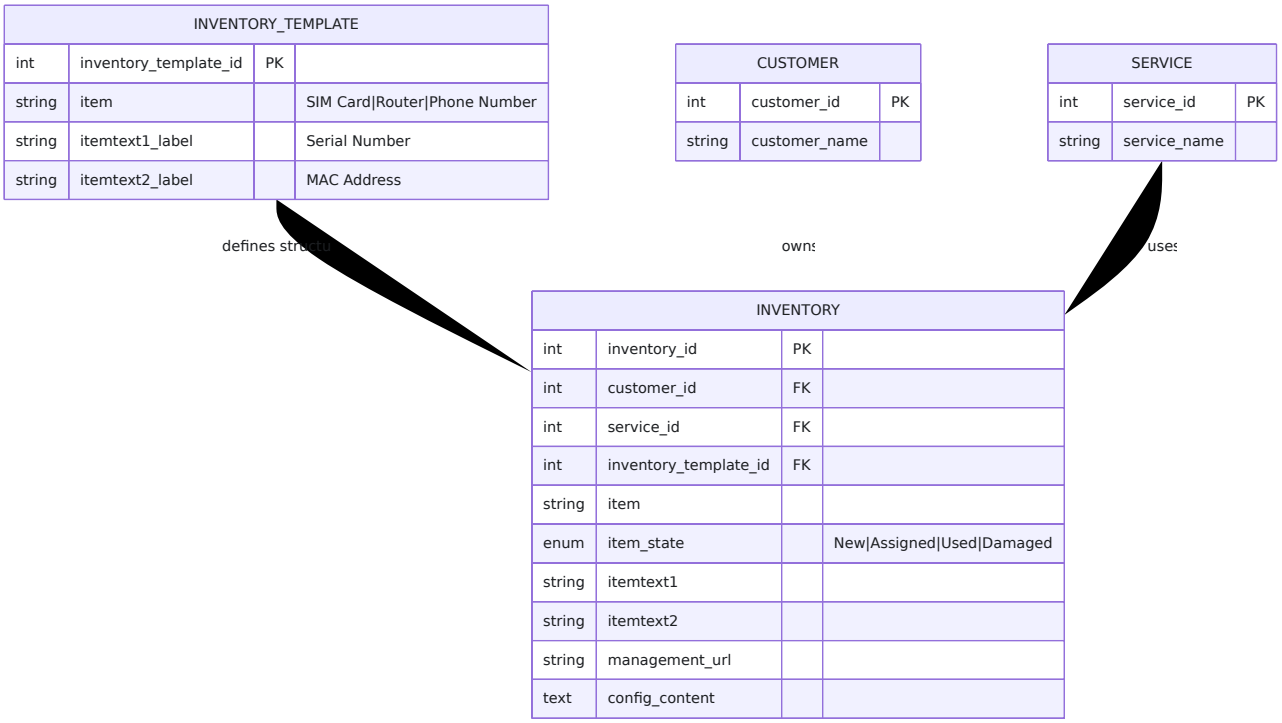


النقاط الرئيسية:

- المنتجات هي قوالب؛ الخدمات هي حالات نشطة
- ينشئ التوفير أو يعدل الخدمات عبر دفاتر اللعب Ansible
- تحتوي كل وظيفة توفير على عدة أحداث لتتبع التقدم
- يمكن أن ينشئ توفير منتج واحد خدمات متعددة (حزم)

نظام الجرد

تتبع الأصول المادية والافتراضية المخصصة للعملاء.

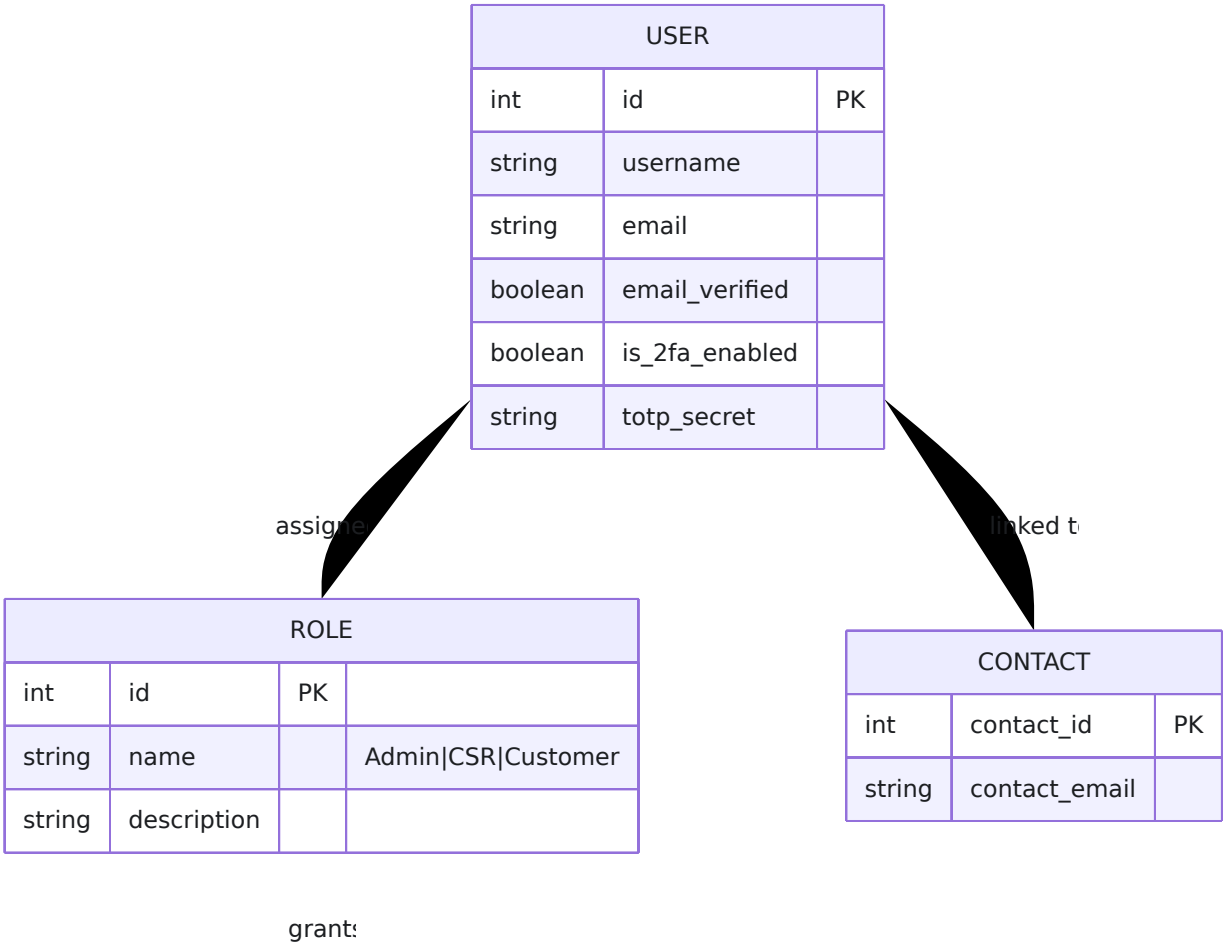


النقاط الرئيسية:

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

المستخدم والأمان

حسابات المستخدمين مع التحكم في الوصول القائم على الأدوار.



PERMISSION			
int	id	PK	
string	name		customer.read|service.create
string	description		

النقاط الرئيسية:

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

قائمة العملاء

توفر قائمة العملاء جدولًا قابلًا للبحث والتصفية لجميع العملاء في النظام.

الميزات:

- **بحث** - تصفية العملاء حسب الاسم أو المعرف
- **إجراءات جماعية** - تحديد عدة عملاء لعمليات الدفعة
- **تصفح 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 حول كيفية مقارنة هذا العميل بمتوسط ARPU لجميع العملاء في النظام.

تكون خيارات حالة العميل مصممة بناءً على الاحتياجات المحددة لعملك، ولكن عادةً ما تشمل خيارات مثل نشط، غير نشط، معلق، إ��خ، 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 إجراء المدفوعات قبل أن تتمكن من حذف العميل، مما سيؤدي بدوره إلى أرشفة العميل وجميع البيانات المرتبطة به، 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 مواقع متعددة 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 وتوليد بيانات الموقع.

كيف يعمل تشفير العنوان:

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

1. **بحث العنوان (موصى به)**
    - استخدم شريط البحث في أعلى النموذج
    - اكتب عنوانًا وتقترح خرائط Google التلقائية المطابقة
    - اختر العنوان الصحيح من القائمة المنسدلة
    - يقوم النظام تلقائيًا بملء:
        - اسم المكان من خرائط Google - **اسم الموقع**
        - رقم الشارع واسم الشارع - **عنوان السطر 1**
        - وحدة/رقم الشقة - **عنوان السطر 2**
        - المحلية - **المدينة**
        - المنطقة الإدارية - **الولاية/المنطقة**
        - الرمز البريدي - **الرمز البريدي**
        - اسم الدولة - **الدولة**
        - الإحداثيات الدقيقة - **خط العرض وخط ا��طول**
        - رمز الموقع المفتوح المكون من 11 حرفًا - **رمز Plus** (مثل، `8C3MFJV8+2F`)
        - معرف المكان الفريد - **معرف مكان خرائط Google**
2. **علامة خريطة قابلة للسحب (يدوي)**
    - اسحب الدبوس على الخريطة إلى الموقع الدقيق
    - يقوم النظام بإجراء **تشفير جغرافي عكسي** للحصول على العنوان من الإحداثيات
    - يتم ملء نفس الحقول تلقائيًا استنادًا إلى موقع الدبوس
    - مفيد للمناطق الريفية أو عندما يكون العنوان غير دقيق

تو‌ليد رمز Plus:

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

مثال:

> عنوان السطر 1: "123 Main Street" ✓ الحقول المملوءة تلقائيًا: ✓ اسم الموقع: "123 Main Street" ✓ المدينة: "لندن" ✓ الولاية: "Greater London" ✓ الدولة: "المملكة المتحدة" ✓ الرمز البريدي: "SW1A 1AA" ✓ خط الطول: 0.1278- ✓ خط العرض: 51.5074 ✓ معرف مكان Google: "ChIJdd4hrwug2EcRmSrV3Vo6llI" ✓ رمز Plus: "9C3XGPHC+3Q"

متطلبات التحقق:

قبل حفظ الموقع، يتحقق النظام من:

- **خط العرض وخط الطول** يجب أن يتم تعيينهما (إما عبر البحث أو سحب العلامة)
- **الدولة** يجب أن تكون مملوءة (تكون القيمة الافتراضية هي `REACT_APP_DEFAULT_COUNTRY` إذا لم يتم تعيينها)
- **رمز Plus** يجب أن يكون 12 حرفًا (11 + 1 للتعبئة)

إذا فشل التحقق، سترى خطأ:

التغذية الراجعة المرئية:

يوفر الواجهة تغذية راجعة في الوقت الحقيقي:

أو

كيف يتم استخدام بيانات الموقع

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

1. تقديم الخدمات والتركيب

- **الفنيون الميدانيون** - الوصول إلى رمز Plus للتنقل إلى الموقع الدقيق للموقع
- **جدولة التركيب** - تعيين الفنيين بناءً على القرب ال��غرافي
- **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 <integrations_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** - الإيرادات حسب المدينة، الولاية، المنطقة
- **خرائط كثافة الخدمة** - خريطة حرارية لمواقع الخدمة
- **تخطيط التوسع** - تحديد المناطق غير المخدومة

4. العملاء التجاريون متعدد المواقع

- **إدارة المواقع** - تتبع الخدمات في أي المواقع
- **فواتير منفصلة** - فوتره 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**
   - قم بالتكبير إلى المنطقة التقريبية
   - اسحب الدبوس إلى العقار/المبنى الدقيق
   - يقوم النظام بإنشاء رمز Plus لذلك الموقع الدقيق
2. **مشاركة رمز Plus**
   - شارك رمز Plus مع العميل (مثل، `C3MFJV8+2F` `8`)
   - يمكن للعميل إدخال هذا في خرائط Google لرؤية الموقع الدقيق
   - يستخدم الفنيون الميدانيون رمز Plus للتنقل
3. **ملاحظات العنوان**
   - استخدم حقل "ملاحظة العنوان" للحصول على توجيهات إضافية
   - مثال: "انعطف يسارًا عند الحظيرة الحمراء، 500 متر بعد شبكة الماشية"
   - الملاحظات مرئية لفرق التركيب

نصيحة

يمكنك سحب الدبوس على الخريطة إلى الموقع الصحيح إذا كان العنوان غير دقيق. سيقوم النظام بتشفير الموقع عكسيًا و ملء جميع حقول العنوان تلقائيًا.

جهة الاتصال - التفاصيل

جهات الاتصال هم الأشخاص المرتبطون بالعميل. يمكن أن تكون 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 لديها 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 الفواتير، ونوع جهة الاتصال يسمح لنا بالتحكم في ذلك.

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

يتم مزامنة جهات الاتصال مع تكامل `Mailjet <integrations_mailjet>` ، مما يسمح لنا بإرسال حملات بريد إلكتروني مستهدفة بناءً على نوع جهة الاتصال، الموقع، الإنفاق الشهري، أو الخدمات المشتراة، وإدارة جميع قوالب البريد الإلكتروني المستخدمة للتواصلات المعاملات.

التنقل

تم تصميم OmniCRM من الألف إلى الياء ليكون استجابة وبديهيًا.

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، ستجد قائمة المستخدم، التي تتيح لك تسجيل الخروج، تغيير كلمة المرور، 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 مؤسست��).

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

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

إذا كنت أمير الظلام، يمكنك التبديل إلى وضع الظلام 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 ائتمانًا، يتم إنشاء معاملة لمبلغ الائتمان، أو إذا تم فرض رسوم تركيب، يتم إنشاء معاملة لرسوم التركيب.

تُجمع المعاملات معًا لتشكيل `<payments_invoices>`، والتي تُرسل إلى العميل `الـفوا تـير` 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 المعاملات بالتجزئة (على مر الزمن)
- **إجمالي المعاملات غير المفوترة** - مجموع المعاملات التي لم يتم تضمينها بعد في فاتورة
- **إجمالي المعاملات هذا الشهر** - مجموع المعاملات الت�� تم إنشاؤها هذا الشهر الميلادي
- **إجمالي المعاملات الشهر الماضي** - مجموع المعاملات التي تم إنشاؤها الشهر الماضي الميلادي

تنسيق القيم:

- القيم التي تزيد عن 1,000: عرضها كلاحقة "k" (1.5$، على سبيل المثال k)
- القيم التي تزيد عن 1,000,000: عرضها كلاحقة "M" (2.3$، على سبيل المثال M)
- القيم التي تزيد عن 1,000,000,000: عرضها كلاحقة "B" (1.1$، على سبيل المثال 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 تم فوترتها، شرطة إذا لم يتم فوترتها بعد

الإجراءات لكل صف:

كل صف لديه 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 للموصلات، أجهزة التوجيه، بطاقات SIM
- **رسوم الدفع المتأخر** - غرامات على الفواتير المتأخرة
- **رسوم يدوية** - رسوم مخصصة يضيفها الموظفون

معاملات الائتمان (المدفوعات/الاستردادات)

مبالغ سلبية تقلل من رصي�� 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 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

ماذا يحدث:

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

البحث وت��فية المعاملات

البحث

استخدم شريط البحث للعثور على المعاملات. يبحث عبر:

- معرف المعاملة
- العنوان
- الوصف
- معرف الفاتورة

الفلاتر

تطبيق الفلاتر لتضييق قائمة المعاملات:

الفلاتر المتاحة:

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

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. حدد 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 توليد الفاتورة
- يمكن عرضها عن طريق تصفية "المعاملات الملغاة"
- يتم خصمها من إجمالي "المعاملات غير المفوترة"

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 تشمل المعاملات الضريبة، 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 الضريبة 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

- 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 لتوليد الفاتورة التالية
- يمكن إلغاؤها
- تُحسب 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

تولید الفاتورة:

عند توليد فاتورة لعميل:

1. يقوم النظام بالبحث عن 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 توليد الفواتير.

سير العمل الشائعة

سير العمل 1: ائتمان يدوي لعدم توفر الخدمة

1. يتصل العميل: "كانت الخدمة متوقفة لمدة يومين"
2. يقرر الموظف منح ائتمان بقيمة 10£
3. انتقل إلى علامة التبويب **الفوترة** للعميل
4. انقر على "**+ إضافة معاملة**"
5. اختر نوع المعاملة **الائتمان**
6. اختر نوع الائتمان **الدفع النقدي**
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 **الخصم**
6. أدخل العنوان: "رسوم التركيب"
7. أدخل الوصف: "زيارة الفني الميداني لتركيب الألياف"
8. أدخل تكلفة التجزئة: 75.00
9. أدخل تكلفة الجملة: 45.00 (اختياري، لتتبع الهامش)
10. اختر الخدمة المثبتة
11. اختر الموقع الذي تم التركيب فيه
12. انقر على "**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 الإجمالي غير المفوتر وفقًا لذلك

سير العمل 4: العثور على المعاملات لفاتورة

1. تحتاج إلى توليد 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 لتوليد 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 المعاملة

- **السبب:** تم فوترتها بالفعل
- **الإصلاح:** المعاملة جزء من تاريخ الفواتير. إذا كانت هناك حاجة إلى استرداد، قم بإنشاء معاملة ائتمان بدلاً من ذلك.

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.

لا تظهر المعاملة 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".

لا يتطابق الإجمالي غير المفوتر مع المتوقع

- **السبب:** تم فوترت بعض المعاملات بالفعل، أو تم استبعاد المعاملات الملغاة
- **الإصلاح:** تطبيق الفلتر "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 لل��ميل
- **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` - توليد 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 `المعاملات </payments_transaction>` للدفع.

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 توليد 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.

ويمكن تخصيصها `Mailjet <integrations_mailjet>` تكون فواتير العملاء مُعدة بالكامل مع لتشمل شعار الشركة، والعنوان، 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، والعلامة التجارية، والألوان، والتخطيط بالكامل.

موقع قالب الفاتورة

يتم تخزين قوالب الفواتير في `OmniCRM-API/invoice_templates/`

القوالب الافتراضية:

- `norfone_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: 'norfone_invoice_template.html'
```

المتغيرا�ت المتاحة في القالب

تتمتع قوالب الفواتير بالوصول إلى المتغيرات التالية من Jinja2:

معلومات الفاتورة:

- `{{ invoice_number }}` - معرف الفاتورة الفريد (على سبيل المثال، `INV-2025-001234`)
- `{{ date }}` - تاريخ إصدار الفاتورة (بتنسيق ISO: `2025-01-10T12:00:00`)
- `{{ due_date }}` - تاريخ استحقاق الدفع (على سبيل المثال، `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 }}` - 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 }}` - البلد

عناصر خط المعاملة:

قم بالتكرار 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 خط المعاملة
- `sub_transaction.tax_percentage` - نسبة الضريبة المطبقة (على سبيل المثال، 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 norfone_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 بعنوان 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 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، البلد</p>
    <p>الهاتف: +1-555-123-4567 | البريد الإلكتروني:
billing@yourcompany.com</p>
    <p>رقم التعريف الضريبي: 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>رقم الحساب: 987654321</li>
        <li>المرجع: {{ invoice_number }}</li>
    </ul>
</div>
```

5. **الشروط والأحكام**

```
<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 توليد الفاتورة

1. انتقل إلى عميل لديه معاملات
2. توليد فاتورة اختبار
3. تنزيل PDF للتحقق من التنسيق
4. إرسال الفاتورة إلى نفسك لاختبار تسليم البريد الإلكتروني

التخصيص المتقدم

المحتوى الشرطي:

استخدم شروط Jinja2 لإظهار/إخفاء المحتوى:

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

{% if client.address.country == "Australia" %}
    <p>GST مشمول: ${{ "%.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 وتف��صيل 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.

رمز الاستجابة السريعة 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 الممارسات لتنسيق PDF

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

CSS المدعوم:

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

غير المدعوم:

- 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. **التفتيش 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
    - مجموعات المعاملات صحيحة
    - تم تنسيق التواريخ بشكل صحيح
    - يتم استبدال جميع المتغيرات بشكل صحيح
3. **جودة PDF:**
    - حجم الملف معقول (<5MB)
    - الصور واضحة وحادة
    - لا يوجد نص مقطوع أو فائض
    - تظهر فواصل الصفحات في الأماكن المناسبة
4. **الفواتير متعددة الصفحات:**
    - تتكرر الرؤوس في كل صفحة
    - تظهر أرقام الصفحات
    - قوائم المعاملات الطويلة تتجزأ 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
- **الإصلاح:** تحقق من التهجئة بالضبط (حساسة لحالة الأحرف)، تحقق من وجود البيانات في قاعدة البيانات

تخطيط PDF معطل:

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

الصور لا تظهر:

- **السبب:** عناوين URL نسبية أو موارد خارجية محجوبة
- **الإصلاح:** استخدم عناوين URL مطلقة HTTPS، 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 توليد PDF:

- **السبب:** أخطاء في بناء HTML أو تعطل WeasyPrint
- **الإصلاح:** تحقق من سجلات WeasyPrint، قم بتبسيط التخطيطات المعقدة

تخزين PDF للفواتير

نظام تخزين PDF المكرر، يتضمن OmniCRM لتحسين الأداء وتقليل توليد PDF للفواتير. عندما يتم توليد PDF فاتورة لأول مرة، يتم تخزينه في قاعدة البيانات للطلبات اللاحقة.

كيفية عمل تخزين PDF:

1. **الطلب الأول** - عندما يتم طلب PDF فاتورة (تنزيل أو بريد إلكتروني)، يقوم النظام:
   - بتوليد 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 (Base64 مشفرة بـ)
     - اسم الملف
     - تجزئة المحتوى (للتحقق من السلامة)
     - الطابع الزمني للإنشاء
2. **الطلبات اللاحقة** - عندما يتم طلب نفس الفاتورة مرة أخرى:
   - يتحقق النظام من وجود PDF مخزن بواسطة invoice_id
   - 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 يدويًا

الفوائد:

- **الأداء** - تسليم PDF فوري للطلبات المتكررة (لا يوجد 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 التحميل أثناء التوليد الأولي، الطلبات اللاحقة تكون فورية

إدارة التخزين:

للفواتير تلقائيًا بواسطة النظام. يمكن حذف إدخالات التخزين 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 في 30 يومًا)
- حدود التخزين (تطبيق حدود حجم التخزين إذا لزم الأمر)

سلوك API:

أو واجهة المستخدم API عند تنزيل فاتورة عبر:

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

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

```
-- مسح جميع PDF الفواتير المخزنة (في تشغيل 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 العميل]**
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 بالتجزئة (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 إنشاؤ��ا الشهر الماضي الميلادي مع العدد

تنسيق القيم:

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

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

يعرض 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 است��قاق 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 لم يتم دفعها)
- **(البريد الإلكتروني)** ✉ - إرسال بريد إلكتروني بالفاتورة إلى العميل
- **(استرداد)** - استرداد دفعة Stripe (فقط للفواتير المدفوعة عبر Stripe)

توليد فاتورة

انقر على **"+ توليد فاتورة برو forma"** لإنشاء فاتورة جديدة.

وصف الحقول:

- **البحث عن العملاء** - اختر العميل (يظهر فقط في العرض على مستوى النظام، يتم ملؤه 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 نطاق التاريخ مع القدرة على تضمين/استبعاد معاملات محددة

اختيار المعاملات:

- **(علامة زائد خضراء)** ✓ - انقر لاستبعاد معاملة من الفاتورة
- **(حمراء X علامة)** × - انقر لتضمين معاملة تم استبعادها مسبقًا

- **تحديد الكل** - تضمين جميع المعاملات المعروضة
- **مسح الكل** - استبعاد جميع المعاملات
- تظهر المعاملات المستبعدة بلون رمادي مع نص مشطوب
- يتم تحديث الإجماليات في الوقت الحقيقي أثناء تحديد/إلغاء تحديد المعاملات

ماذا يحدث:

1. يقوم ��لنظام بالبحث عن جميع المعاملات غير المفوترة للعميل ضمن نطاق التاريخ
2. يعرض معاينة المعاملة مع القدرة على تضمين/استبعاد المعاملات الفردية
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 كـ مفوترة (تم ملؤه `invoice_id`)
7. تبقى المعاملات المستبعدة غير مفوترة ومتاحة للفواتير المستقبلية
8. تظهر الفاتورة في القائمة مع حالة "غير مدفوعة"

حالات الاستخدام النموذجية:

الفوترة الشهرية: تعيين تاريخ البدء إلى الأول من الشهر، تاريخ الانتهاء إلى آخر يوم من الشهر، تظهر المعاينة جميع المعاملات غير المفوترة من تلك الفترة. حدد الكل أو استبعد يدويًا معاملات محددة.

فاتورة محددة للخدمة: استخدم نفس نطاق التاريخ، ثم استبعد يدويًا المعاملات غير المرغوب فيها (على سبيل المثال، استبعاد المعاملات غير المحمولة لإنشاء فاتورة خاصة بالمحمول 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، الإجماليات، والإجراءات المتاحة.

نافذة تفاصيل الفاتورة:

- **معلومات الفاتورة** - تظهر معرف الفاتورة، العنوان، التواريخ، حالة الدفع، وحالة الإلغاء
- **قائمة المعاملات** - تعرض جميع المعاملات المضمنة في الفاتورة مع:
  - تاريخ المعاملة
  - العنوان والوصف
  - تكلفة التجزئة
  - مبلغ الضريبة ونسبتها (مُنسقة كـ `$10.00 (%10)`)
  - تظهر المعاملات المعفاة من الضريبة كـ "-" 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 (متاحة دائمًا) الفاتورة
  - **إرسال بريد إلكتروني** - إرسال الفاتورة إلى العميل (الفواتير غير الملغاة)
  - **دفع الفاتورة** - معالجة الدفع (فقط للفواتير غير المدفوعة وغير الملغاة)
  - **استرداد** - استرداد دفعة Stripe (فقط للفواتير المدفوعة عبر Stripe)
  - **حذف** - إلغاء الفاتورة (فقط للفواتير غير المدفوعة وغير الملغاة)

تنزيل PDF الفواتير

في نافذة تفاصيل الفاتورة **"PDF** انقر على **أيقونة التنزيل (⬇)** في الجدول أو زر **"تنزيل** لتنزيل فاتورة كملف PDF.

عملية التنزيل:

1. انقر على أيقونة التنزيل بجانب 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 تخزين PDF:

- من القالب، ويتم تخزينه في قاعدة البيانات (قد PDF **التنزيل الأول** - يتم توليد 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:

- معطلاً API **المؤشر لا يتوقف أبدًا** - تحقق من وحدة التحكم في المتصفح، قد يكون
- **فارغ أو تالف PDF** - تحقق من قالب الفاتورة بحثًا عن أخطاء بناء
- **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 الفاتورة غير المدفوعة
2. تفتح نافذة الدفع مع عرض تفاصيل الفاتورة
3. اختر طريقة الدفع:
    - **معاملة Stripe** - فرض رسوم على بطاقة الائتمان المحفوظة (متاحة لجميع المستخدمين)
    - **نقدي** - دفع نقدي يدوي (موظف فقط)
    - **استرداد** - تطبيق استرداد كدفع (موظف فقط)
    - **معاملة POS** - محطة نقاط البيع (موظف فقط)
    - **تحويل بنكي** - تحويل بنكي يدوي (موظف 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** - يفرض رسومًا على البطاقة عبر API Stripe
    - **طرق أخرى** - ينشئ معاملة سلبية لمبلغ الدفع
8. تتغير حالة الفاتورة إلى "مدفوعة"
9. يتم عرض إشعار النجاح

البوابة الذاتية مقابل دفع الموظف:

:doc:`بوابة الرعاية الذاتية <self_care_portal>` (العملاء):

- متاحة فقط دفع Stripe
- يجب أن يكون لديك طريقة دفع محفوظة
- يظهر تحذير إذا لم توجد طرق دفع
- يتم توفير رابط لإضافة طريقة دفع

بوابة الموظف (المسؤولون):

- جميع طرق الدفع متاحة
- يمكنهم وضع علامة على الفاتورة كمدفوعة يدويًا (نقدًا، POS، تحويل بنكي)
- يمكنهم إدخال أرقام مرجعية للتتبع

تحذير طريقة الدفع:

إذا لم يكن لدى العميل أي طرق دفع محفوظة، يتم عرض تحذير يحثهم على إضافة طريقة دفع قبل أن يتمكنوا من دفع الفواتير.

إرسال الفواتير عبر البريد الإلكتروني

انقر على **أيقونة البريد الإلكتروني (✉)** لإرسال الفاتورة إلى العميل.

ماذا يحدث:

1. انقر على أيقونة البريد الإلكتروني بجانب الفاتورة

2. يسترجع النظام PDF (أو يولدها إذا لم تكن مخزنة) الفاتورة من ��لتخزين
3. يرسل البريد الإلكتروني عبر `Mailjet <integrations_mailjet>` باستخدام قالب `api_crmCommunicationCustomerInvoice`
4. يتضمن البريد الإلكتروني:
    - الفاتورة كمرفق PDF
    - 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 Mailjet API في `crm_config.yaml`
- **العميل لا يتلقى** - تحقق من عناوين البريد الإلكتروني لجهات الاتصال الخاصة بالعميل
- **لا يتم إرفاقه PDF** - تحقق من نجاح توليد PDF (جرب تنزيله أولاً)

إلغاء الفواتير

انقر على **أيقونة الحذف (🗑)** لإلغاء فاتورة.

المتطلبات:

- يجب أن تكون الفاتورة **غير مدفوعة**
- لا يمكن إلغاء الفواتير المدفوعة (يجب استردادها بدلاً من ذلك)

كيفية الإلغاء:

1. حدد الفاتورة غير المدفوعة في القائمة
2. انقر على أيقونة الحذف (🗑)
3. أكد في النافذة:

ماذا 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 "ملغاة:" في العنوان
- يتم تعطيل إجراءات الفاتورة (لا يوجد تنزيل، دفع، أو بريد إلكتروني)
- يمكن عرضها عن طريق تصفية "الفواتير الملغاة"

ملاحظات مهمة:

- الإلغاء ليس هو نفسه الاسترداد
- **الإلغاء** = "يجب ألا تكون هذه الفاتورة موجودة أبدًا" (خطأ في الفوترة، مكرر)
- **الاسترداد** = "عكس فاتورة مدفوعة صالحة" (إرجاع الأموال 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 فقط لمستخدمي الموظفين (ليس للرعاية الذاتية)

كيفية الاسترداد:

1. حدد فاتورة Stripe المدفوعة
2. انقر على ��يقونة 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

ماذا يحدث بعد الاسترداد:

- تبقى الفاتورة في النظام (غير ملغاة)
- تظهر الحالة "مستردة"
- تبقى المعاملات مرتبطة بالفاتورة
- يتلقى العميل استردادًا إلى طريقة الدفع الأصلية (3-7 أيام عمل)
- يظهر استرداد المعاملة في لوحة معلومات Stripe

قيود الاسترداد:

- لا يمكن استرداد الفواتير المدفوعة نقدًا، أو عبر POS، (يتطلب عكس يدوي)
- لا يمكن الاسترداد الجزئي (مبلغ الفاتورة بالكامل فقط)
- لا يمكن الاسترداد مرتين

البحث وتصفية الفواتير

البحث

استخدم شريط البحث للعثور على الفواتير. يبحث 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 الفلتر:

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

الفرز

انقر على أي رأس عمود للفرز:

- **ID** - فرز حسب معرف الفاتورة (الأحدث/الأقدم)
- **العنوان** - فرز أبجديًا
- **تاريخ الاستحقاق** - فرز حسب تاريخ الاستحقاق
- **تاريخ الإنشاء** - فرز حسب تاريخ الإنشاء
- **المبلغ** - فرز حسب تكلفة التجزئة (الأعلى/الأدنى)
- **الحالة** - فرز حسب حالة الدفع (المدفوعة أولاً أو غير المدفوعة أولاً)

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

الصفحات

انتقل عبر قوائم الفواتير الكبيرة مع عناصر التحكم في الصفحة التي تعرض الصفحة الحالية، إجمالي الصفحات، ومحدد العناصر لكل صفحة (10، 25، 50، أو 100 عنصر).

سير العمل الشائعة للفواتير

سير العمل 1: الفوترة الشهرية مع معاينة المعاملة

1. يصل نهاية الشهر (على سبيل المثال، 31 يناير)
2. انتقل إلى **الفوترة** → **الفواتير**
3. انقر على **"+ توليد فاتورة برو forma"**
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. يظهر قسم **معاينة المعاملة** مع جميع المعاملات غير المفوترة من يناير
7. مراجعة المعاينة:
    - يتم تضمين جميع المعاملات افتراضيًا
    - تحقق من الإجماليات: المجموع الفرعي، الضريبة، و��جمالي الفاتورة
    - تحقق من صحة جميع الرسوم
8. انقر على **"توليد الفاتورة"** (يظهر 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. لدى العميل خدمات متعددة (محمول + إنترنت) ورسوم متنوعة
2. يريد فواتير منفصلة لكل خدمة
3. **توليد الفاتورة الأولى (خدمات المحمول):**
    - انقر على **"+ توليد فاتورة برو forma"**
    - العنوان: "خدمات المحمول - يناير 2025"
    - تاريخ البدء/الانتهاء: 1-31 يناير
    - تاريخ الاستحقاق: 15 فبراير
    - في معاينة المعاملة، **استبعد** جميع المعاملات غير المحمولة:
        - بجانب معاملات الإنترنت انقر على زر **X**
        - بجانب الرسوم المتنوعة انقر على زر **X**
        - تبقى فقط معاملات خدمة المحمول محددة
    - تحقق من أن الإجماليات تعكس فقط خدمات المحمول
    - انقر على **"توليد الفاتورة"** (يظهر عدد معاملات المحمول)
4. **توليد الفاتورة الثانية (خدمات الإنترنت):**

- مرة أخرى انقر على **"+ توليد فاتورة برو forma"**
- العنوان: "خدمات الإنترنت - يناير 2025"
- تاريخ البدء/الانتهاء: 1-31 يناير (نفس الفترة)
- في معاينة المعاملة:
  - تم فوترت معاملات المحمول بالفعل (لا تظهر)
  - استبعد الرسوم المتنوعة باستخدام زر **X**
  - تبقى فقط معاملات خدمة الإنترنت
- انقر على **"توليد الفاتورة"**

5. **توليد الفاتورة الثالثة (الرسوم الإضافية):**
   - مرة أخرى انقر على **"+ توليد فاتورة برو forma"**
   - العنوان: "الرسوم الإضافية - يناير 2025"
   - تظهر فقط الرسوم المتنوعة غير المفوترة في المعاينة
   - انقر على **"تحديد الكل"** لتضمين الكل
   - انقر على **"توليد الفاتورة"**
6. أرسل جميع الفواتير الثلاثة إلى العميل

سير العمل 3: استبعاد المعاملات المتنازع عليها أو المعلقة

1. يصل نهاية فترة الفوترة
2. انتقل إلى علامة التبويب **الفوترة** للعميل
3. انقر على **"+ توليد فاتورة برو forma"**
4. تعيين تواريخ فترة الفوترة
5. تظهر معاينة المعاملة 20 معاملة
6. لدى العميل معاملة متنازع عليها وأخرى معلقة قيد التحقيق
7. في معاينة المعاملة:
   - حدد المعاملة المتنازع عليها (على سبيل المثال، "رسوم تجاوز البيانات")
   - انقر على زر **X** لاستبعادها
   - حدد المعاملة المعلقة (على سبيل المثال، "رسوم التركيب")
   - انقر على زر **X** لاستبعادها
   - يتم تحديث عدد المعاملات: "تم اختيار 18 معاملة"
   - يتم إعادة حساب الإجماليات تلقائيًا
8. مراجعة الإجماليات المحدثة (تستبعد المبالغ المتنازع عليها)

9. انقر على **"توليد الفاتورة (18)"**
10. تم إنشاء الفاتورة مع المعاملات المعتمدة فقط
11. تبقى المعاملات المتنازع عليها/المعلقة غير مفوترة للدورة الفوترة التالية

سير العمل 4: مراجعة سريعة للفواتير وتعديلها

1. يقوم الموظف بتوليد فاتورة شهرية
2. تظهر معاينة المعاملة إجماليًا مرتفعًا غير متوقع
3. مراجعة كل معاملة في المعاينة:
    - لاحظ رسومًا مكررة لنفس الخدمة
    - انقر على **X** لاستبعاد المكررة
    - لاحظ معاملة اختبار لا ينبغي فوترتها
    - انقر على **X** لاستبعاد معاملة الاختبار
4. يتم تحديث الإجماليات في الوقت الحقيقي
5. تحقق من أن الإجمالي الجديد يتطابق مع المبلغ المتوقع
6. انقر على **"توليد الفاتورة"** 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. يكمل الفني الميداني التركيب
2. يضيف الموظف معاملة التركيب يدويًا
3. انتقل إلى علامة التبويب **الفوترة** للعميل
4. انقر على **"+ توليد فاتورة برو forma"**
5. تعيين التواريخ:
    - تاريخ البدء: اليوم
    - تاريخ الانتهاء: اليوم
    - تاريخ الاستحقاق: اليوم + 7 أيام
    - العنوان: "خدمات التركيب"
6. تظهر معاينة المعاملة فقط معاملات اليوم
7. تحقق من ظهور رسوم التركيب
8. استبعد أي رسوم متكررة باستخدام زر **X** (إذا كانت موجودة)
9. انقر على **"توليد الفاتورة"**

10. أرسل إلى العميل على الفور
11. يدفع العميل عبر الإنترنت عبر Stripe

سير العمل 6: مراجعة الفاتورة قبل الاتصال ب��لعميل

1. يتصل العميل بسؤال حول الفوترة
2. يتنقل الموظف إلى قائمة فواتير العميل
3. **انقر على صف الفاتورة** لفتح نافذة تفاصيل الفاتورة
4. مراجعة معلومات الفاتورة:
    - معرف الفاتورة، التواريخ، الحالة
    - جميع المعاملات المضمنة مع الوصف
    - 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 الفوترة

1. يبلغ العميل عن رسوم غير صحيحة
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. توليد فاتورة جديدة مع المعاملات المصححة:
    - استخدم معاينة المعاملة لاستبعاد المعاملة المصححة إذا لزم الأمر
    - تضمين الرسوم الصالحة فقط

11. أرسل الفاتورة المصححة إلى العميل

سير العمل 8: معالجة مدفوعات متعددة

1. يحضر العميل نقدًا لدفع فواتير متعددة
2. انتقل إلى علامة التبويب **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 **"نقدي"**
8. أدخل المرجع: "تم الدفع نقدًا 15-01-2025"
9. انقر على **"دفع الفاتورة"**
10. تغلق النافذة، وتُعَلَّم الفاتورة كـ "مدفوعة"
11. كرر ذلك لبقية الفواتير
12. الآن جميع الفواتير مُعَلَّمَة كـ "مدفوعة"

سير العمل 9: التعامل مع طلب استرداد

1. يطلب العميل استردادًا لمبلغ زائد
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 توليد 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. اضبط نطاق التاريخ أو أزل الفلتر.

فشل توليد PDF الفاتورة

- **السبب:** خطأ في بناء القالب، تعطل WeasyPrint، أو بيانات العميل مفقودة
- **الإصلاح:** تحقق من قالب الفاتورة بحثًا عن الأخطاء، تحقق من ملء حقول HTML عنوان العميل، راجع سجلات API.

فشل الدفع مع خطأ Stripe

- **السبب:** تم رفض البطاقة، أو عدم كفاية الأموال، أو انتهاء صلاحية البطاقة، 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 كانت هناك حاجة إلى استرداد، استخدم وظيفة الاسترداد لفواتير Stripe أو قم بإنشاء معاملة ائتمان يدويًا.

البريد الإلكتروني للفواتير لا يُرسل

- **السبب:** بيانات اعتماد Mailjet API غير صحيحة، أو العميل ليس لديه جهة اتصال فوترة، أو القالب مفقود
- **الإصلاح:** تحقق من تكوين Mailjet في `crm_config.yaml`، تحقق من جهات اتصال العميل، تحقق من وجود قالب بريد الفاتورة.

زر الاسترداد لا يظهر

- **السبب:** تم دفع الفاتورة نقدًا/POS/تحويل بنكي (ليس عبر Stripe)، أو لم يتم دفع الفاتورة

- **الإصلاح:** يظهر زر الاسترداد فقط للمدفوعات عبر Stripe. بالنسبة لطرق الدفع الأخرى، قم بإنشاء معاملة ائتمان يدويًا.

يظهر PDF قديم في التنزيل

- **السبب:** تم تخزين PDF قبل تحديث القالب
- **الإصلاح:** امسح تخزين PDF الفواتير: `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 الرعاية الذاتية
- **الإصلاح:** يجب على العميل إضافة بطاقة ائتمان في صفحة **طرق الدفع** قبل دفع الفواتير.

تم توليد فواتير متعددة لنفس الفترة

- **السبب:** قام الموظف بتوليد الفاتورة مرتين، أو تتداخل نطاقات التواريخ
- **الإصلاح:** إلغاء الفاتورة المكررة. اضبط نطاقات التواريخ لمنع التداخل. استخدم معاينة المعاملة للتأكد من مجموعات المعاملات الفريدة.

لا تظهر المعاملات في المعاينة

- **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 الفاتورة التي تحتوي على المعاملات.

لا يمكن استبعاد المعاملة من توليد الفاتورة

- **السبب:** تم فوترتها بالفعل أو مشكلة في المتصفح
- **الإصلاح:** تحقق من أن المعاملة تظهر في المعاينة مع علامة تحقق. قم بتحديث الصفحة وحاول مرة أخرى. امسح ذاكرة التخزين المؤقت للمتصفح إذا استمرت المشكلة.

الإجمالي 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 لكل معاملة والضريبة. تحقق من أن المعاملات المستبعدة تظهر بلون رمادي. تحقق من شارة عدد المعاملات على زر

تحقق من أن تضمين معاملة واحدة على الأقل (ليست مستبعدة). تحقق من أن تاريخ البدء قبل تاريخ الانتهاء. تحقق من تعيين تاريخ الاستحقاق. تأكد من تضمين معاملة واحدة على الأقل (ليست مستبعدة). تحقق من أن تاريخ البدء قبل تاريخ الانتهاء. تحقق من تعيين تاريخ الاستحقاق. **الإصلاح:** تولید الفاتورة.

زر توليد الفاتورة معطل

- **السبب:** لم يتم اختيار أي معاملات أو نطاق التاريخ غير صالح
- **الإصلاح:** تأكد من تضمين معاملة واحدة على الأقل (ليست 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
- **الإصلاح:** قم بتحديث الصفحة. تحقق من وحدة التحكم في المتصفح بحثًا عن الأخطاء. جرب متصفحًا مختلفًا. تحقق من اتصال الإنترنت.

الضريبة على المعاملات لا تظهر في تفاصيل الفاتورة

- **السبب:** المعاملة تحتوي على ضريبة 0% أو `tax_amount` فارغ
- **الإصلاح:** تحقق من أن المعاملة تحتوي على `tax_percentage` محددة. تحقق من أنه عند إنشاء المعاملة. قم بتحديث المعاملة إذا لزم الأمر `tax_amount` تم حساب.

تفتقر أزرار الإجراءات في نافذة تفاصيل الفاتورة

- **السبب:** تم إلغاء الفاتورة أو يفتقر المستخدم إلى الأذونات
- **الإصلاح:** تظهر الفواتير الملغاة فقط زر تنزيل PDF. تحقق من تحقق من حالة الفاتورة. تحقق من دور المستخدم والأذونات.

الوثائق ذات الصلة

- `integrations_mailjet` - تسليم البريد الإلكتروني للفواتير والقوالب
- `administration_configuration` - تكوين قالب الفاتورة
- `payments_transaction` - إنشاء المعاملات التي تظهر على الفواتير
- `payments_process` - معالجة المدفوعات باستخدام طرق الدفع المخزنة
- `payment_system_guide` - مرجع API الدفع وتكوين البائع

إدارة طرق الدفع

يسمح نظام طرق الدفع في OmniCRM للعملاء والموظفين بإدارة بطاقات الدفع بشكل آمن. تمكّن طرق الدفع من **معالجة الدفع متعددة البائعين** باستخدام (Stripe، PayPal، إلخ). الفوترة التلقائية للخدمات، والمدفوعات لمرة واحدة، والرسوم المتكررة دون تخزين بيانات البطاقة الحساسة في OmniCRM.

انظر أيضًا: `دليل نظام الدفع <payment_system_guide>`، `نظرة عامة على الفوترة <billing_overview>`، `معالجة الدفع <payments_process>`، `الفواتير <payments_invoices>`.

نظرة عامة

يوفر نظام طرق الدفع:

- **تخزين آمن للبطاقات** - يتم توكين بطاقات الدفع بواسطة البائعين (Stripe، PayPal)، ولا يتم تخزينها أبدًا في OmniCRM
- **دعم متعدد البائعين** - يتم دعم 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 المنتهية
- **خدمة ذاتية** - يمكن للعملاء إدارة بطاقاتهم الخاصة عبر `بوابة الرعاية الذاتية <self_care_portal>`
- **إدارة الموظفين** - يمكن لموظفي الدعم إضافة/إزالة بطاقات نيابة عن العملاء

طرق الدفع المدعومة:

- بطاقات الائتمان (Visa، Mastercard، American Express، Discover)
- بطاقات الخصم
- بطاقات مسبقة الدفع (إذا كانت مدعومة من شبكة البطاقة)

غير المخزنة في OmniCRM:

يتم توكين تفاصيل البطاقة بواسطة بائعي الدفع وتخزينها بشكل آمن. 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 العميل]**
2. انقر على علامة التبويب **الفوترة**
3. قم بالتمرير إلى قسم **طرق الدفع**

أو مباشرة:

من لوحة معلومات البطاقات المنتهية:

عرض جميع العملاء الذين لديهم بطاقات منتهية:

يعرض هذا قائمة على مستوى النظام للبطاقات التي ستنتهي خلال 60 يومًا القادمة.

قائمة طرق الدفع

يعرض جدول طرق الدفع جميع البطاقات المخزنة لعميل:

وصف الأعمدة:

- الاسم الودي للبطاقة - **اللقب** (على سبيل المثال، "البطاقة الشخصية"، "عمل" Visa")
- علامة البطاقة وآخر 4 أرقام - **المصدر**
- شهر/سنة انتهاء الصلاحية - **الانتهاء** (بتنسيق MM/YYYY)
- تاريخ إضافة البطاقة إلى الحساب - **تمت الإضافة**
- علامة تحقق تشير إلى طريقة الدفع الافتراضية للرسوم التلقائية - **افتراضي**

الإجراءات لكل بطاقة:

كل صف لديه قائمة إجراءات (⋮) مع الخيارات:

- تعيين**

نظرة عامة على علامة الفواتير

توفر علامة الفواتير عرضًا موحدًا لجميع المعلومات المالية للعميل، حيث تجمع بين طرق الدفع والمعاملات والفواتير في واجهة واحدة لإدارة الفواتير بكفاءة.

المستندات ذات الصلة: `Payment Methods <payment_methods>`, `Transactions <payments_transaction>`, `Invoices <payments_invoices>`, `Payment Processing <payments_process>`.

الوصول إلى علامة الفواتير

عرض لكل عميل:

1. انتقل إلى **Customers → [Select Customer]**
2. انقر على علامة **Billing**
3. عرض الأقسام الثلاثة: طرق الدفع، المعاملات، والفواتير

عرض على مستوى النظام:

يمكن الوصول إلى بيانات الفواتير على مستوى النظام بشكل منفصل:

- **Billing → Transactions** - جميع المعاملات عبر جميع العملاء
- **Billing → Invoices** - جميع الفواتير عبر جميع ا��عملاء

بوابة الخدمة الذاتية:

يرى العملاء الذين يصلون إلى `Self-Care Portal <self_care_portal>` نفس هيكل علامة الفواتير:

- عرض وإدارة طرق الدفع الخاصة بهم
- عرض تاريخ المعاملات
- عرض ودفع الفواتير عبر الإنترنت

هيكل علامة الفواتير

تنظم علامة الفواتير 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 (الإجم��لي، 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 المفوترة مقابل غير المفوترة

المستندات: `payments_transaction`

القسم 3: الفواتير

الغرض: تجميع المعاملات في فواتير ليقوم العميل بدفعها

الميزات الرئيسية:

- عرض إحصائيات الفواتير (الإجمالي، غير المدفوعة، هذا الشهر، الشهر الماضي)
- قائمة بجميع الفواتير مع تصفية حسب حالة المدفوعة/المُلغاة
- 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 رد الفواتير

المستندات: `payments_invoices`

تدفق البيانات بين الأقسام

فهم كيفية تدفق البيانات بين الأقسام الثلاثة أمر حاسم لإدارة الفواتير بشكل فعال.

مخطط التدفق

علاقة المعاملة بالفاتورة

1. إنشاء المعاملة:

عندما يتم توفير خدمة أو إضافة رسوم يدوية:

- يتم إنشاء معاملة في قسم **Transactions**
- حالة المعاملة: **غير مفوترة**

- حقل `invoice_id` للمعاملة هو `null`

مثال:

2. إنشاء الفاتورة:

عندما يقوم الموظف بإنشاء فاتورة:

- يتم إنشاء فاتورة في قسم **Invoices**
- يتم تجميع جميع المعاملات غير المفوترة ضمن نطاق التاريخ في الفاتورة
- يتم ملء حقل `invoice_id` للمعاملة
- تتغير حالة المعاملة إلى: **مفوترة**

مثال:

3. تحديث إحصائيات المعاملات:

- **المعاملات غير المفوترة** ينخفض إجمالي
- **إجمالي الفواتير** تزيد إحصائية
- **الفواتير غير المدفوعة** يزيد إجمالي

علاقة الفاتورة بالدفع

1. دفع الفاتورة:

عندما يدفع العميل الفاتورة:

- **طريقة الدفع المحفوظة** تتم معالجة الدفع باستخدام (Stripe بطاقة)
- 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$ (سلبي، يمنح العميل ائتمانًا)

- 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 **Payment Methods**
2. يتم ترميز البطاقة عبر Stripe، وتخزينها بأمان
3. عند دفع الفاتورة، يختار العميل ا��بطاقة المحفوظة
4. يقوم Stripe بخصم البطاقة
5. يتم وضع علامة على الفاتورة كمدفوعة
6. يتم ملء حقل `payment_reference` من Stripe بمعرف نية الدفع

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 للفواتير

توضح هذه العمليات كيفية عمل الأقسام الثلاثة معًا لإنجاز المهام الشائعة.

سير العمل 1: إعداد عميل جديد والفاتورة الأولى

الهدف: إعداد الفواتير للعميل الجديد وجمع أول دفعة

1. **إضافة طريقة الدفع:**

- انتقل إلى العميل → علامة **Billing**
- قسم **Payment Methods** → انقر على **"Add Payment Method"**
- يضيف العميل بطاقة ائتمان عبر Stripe
- يتم حفظ البطاقة كطريقة الدفع الافتراضية

2. **التحقق من المعاملات:**
   - يظهر قسم **Transactions** المعاملات غير المفوترة:
     - رسوم إعداد الخدمة: 50.00$
     - خدمة الشهر الأول: 45.00$
     - إجمالي غير مفوتر: 95.00$
3. **إنشاء فاتورة:**
   - قسم **Invoices** → انقر على **"Generate Proforma Invoice"**
   - تعيين نطاق التاريخ ليشمل الإعداد والشهر الأول
   - انقر على **"Generate Invoice"**
   - تم إنشاء الفاتورة #INV-2025-001234 بمبلغ 95.00$
4. **تحديث المعاملات:**
   - فاتورة #INV-2025-001234 تظهر كلا المعاملتين الآن:
   - 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. **العميل يدفع عبر الإنترنت:**
   - ينقر العميل على رابط الدفع في البريد الإلكتروني
   - يتم توجيهه إلى بوابة الخدمة الذاتية
   - انقر على زر **"Pay Invoice"**
   - اختر طريقة الدفع الافتراضية
   - انقر على **"Pay Invoice"**
   - يقوم Stripe بخصم البطاقة
7. **تحديث الفاتورة:**
   - تتغير حالة الفاتورة إلى **"مدفوعة"**
   - ينخفض إجمالي **الفواتير غير المدفوعة** بمبلغ 95.00$

النتيجة: تم إعداد العميل بالكامل مع طريقة الدفع، وتم دفع الفاتورة الأولى.

سير العمل 2: الفواتير الشهرية المتكررة

الهدف: فواتير جميع العملاء مقابل الخدمة الشهرية في نهاية الشهر

1. **الرسوم التلقائية للخدمات:**
   - يصل نهاية الشهر (31 يناير)
   - يقوم نظام الفواتير تلقائيًا بإنشاء معاملات لجميع الخدمات المتكررة
   - يظهر قسم **Transactions** معاملات جديدة غير مفوترة
2. **مراجعة المعاملات غير المفوترة:**
   - انتقل إلى قسم **Transactions**
   - تصفية: **Invoice Status: Not Invoiced**
   - مراجعة قائمة جميع المعاملات الجاهزة للفوترة
   - التحقق من صحة المبالغ والأوصاف
3. **إنشاء الفواتير:**
   - انتقل إلى **Billing → Invoices** (على مستوى النظام)
   - لكل عميل (أو استخدم عملية الدفعة):
     - انقر على **"Generate Proforma Invoice"**
     - اختر العميل
     - تاريخ البدء: 01-01-2025
     - تاريخ الانتهاء: 31-01-2025
     - تاريخ الاستحقاق: 15-02-2025
     - انقر على **"Generate Invoice"**
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 فاتورة، انقر على أيقونة البريد الإلكتروني
   - يتلقى جميع العملاء الفواتير الشهرية
6. **العملاء يدفعون:**
   - العملاء الذين لديهم طرق دفع محفوظة يدفعون 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 فواتير جميع العملاء لشهر يناير، وتم إرسال الفواتير، ومعالجة المدفوعات.

سير العمل 3: معالجة ائتمان مشكلة الخدمة

الهدف: منح العميل ائتمانًا عن انقطاع الخدمة، وتطبيقه على الفاتورة غير المدفوعة

1. **العميل يبلغ عن المشكلة:**
    - كانت الخدمة معطلة لمدة يومين
    - يستحق العميل ائتمانًا بمبلغ $10
2. **إضافة معاملة ائتمان:**
    - علامة → انتقل إلى العميل **Billing** → قسم **Transactions**
    - انقر على **"+ Add Transaction"**
    - نوع المعاملة: **Credit**
    - نوع الائتمان: **Cash Payment** (أو النوع المناسب)
    - العنوان: "ائتمان انقطاع الخدمة"
    - الوصف: "تعويض عن انقطاع لمدة يومين 8-9 يناير"
    - التكلفة التجزئة: 10.00
    - انقر على **"Add Transaction"**
3. **تم إنشاء المعاملة:**
    - **10.00$-** : تظهر المعاملة في القائمة بمبلغ
    - **غير مفوترة** : حالة المعاملة
    - 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 **Invoices** → انقر على **"Generate Proforma Invoice"**
        - تضمين نطاق التاريخ مع معاملة الائتمان
        - تم إنشاء فاتورة مع تطبيق الائتمان:
5. **العميل يدفع:**
    - 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 البطاقة "انتهت صلاحية البطاقة: "رسالة خطأ
    - يفشل ال��فع، وتبقى الفاتورة غير مدفوعة
2. **تحديث طريقة الدفع:**
    - ينتقل العميل إلى علامة **Billing**
    - قسم **Payment Methods** ← انقر على **"Add Payment Method"**
    - أدخل تفاصيل البطاقة الجديدة (تاريخ انتهاء الصلاحية المحدث)
    - تم حفظ البطاقة الجديدة
3. **تعيين كافتراضي:**
    - ينقر العميل على **"Set as Default"** على البطاقة الجديدة
    - تتم إزالة البطاقة القديمة تلقائيًا (إذا رغبت في ذلك)
4. **إعادة المحاولة للدفع:**
    - انتقل إلى قسم **Invoices**
    - حدد الفاتورة غير المدفوعة
    - انقر على أيقونة **"Pay"**
    - تفتح نافذة الدفع مع البطاقة الافتراضية الجديدة محددة مسبقًا
    - انقر على **"Pay Invoice"**
    - يقوم Stripe بخصم البطاقة الجديدة بنجاح
5. **تحديث الفاتورة:**
    - تتغير حالة الفاتورة إلى **"مدفوعة"**
    - يتم ملء حقل `payment_reference` من بمعرف نية الدفع الجديدة Stripe

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 إنشائها بشكل صحيح

1. **اكتشاف الخطأ:**
    - تم إنشاء الفاتورة #INV-2025-001234 مع نطاق تاريخ خاطئ
    - تضمنت معاملات من شهر خاطئ
    - الفاتورة غير مدفوعة
2. **إلغاء الفاتورة:**
    - انتقل إلى علامة **Billing** → قسم **Invoices**
    - حدد الفاتورة غير الصحيحة
    - انقر على أيقونة الحذف (🗑)
    - تأكيد الإلغاء
    - تم إلغاء الفاتورة
3. **إطلاق المعاملات:**
    - انتقل إلى قسم **Transactions**
    - تظهر جميع المعاملات من الفاتورة الملغاة الآن: **غير مفوترة**
    - يزيد إجمالي **المعاملات غير المفوترة**
    - المعاملات متاحة لفاتورة جديدة
4. **إنشاء فاتورة صحيحة:**
    - قسم **Invoices** → انقر على **"Generate Proforma Invoice"**
    - تعيين نطاق التاريخ الصحيح
    - تطبيق الفلتر إذا لزم الأمر (على سبيل المثال، "Mobile" لفاتورة خاصة بالهاتف المحمول)
    - انقر على **"Generate Invoice"**
    - 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 فوتر 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:**
    - أحضر العميل $300 نقدًا لدفع الفواتير المستحقة
    - انتقل إلى العميل → علامة **Billing**
2. **مراجعة الفواتير غير المدفوعة:**
    - قسم **Invoices** → تصفية: **Paid: Not yet Paid**
    - عرض الفواتير غير المدفوعة:
3. **دفع الفاتورة الأولى:**
    - انقر على أيقونة الدفع على الفاتورة #1234
    - تفتح نافذة الدفع
    - اختر طريقة الدفع **"Cash"**
    - أدخل المرجع: "نقد مدفوع 10-02-2025 - إيصال #001"
    - انقر على **"Pay Invoice"**
    - تم وضع علامة على الفاتورة #1234 كـ **"مدفوعة"**
4. **دفع الفواتير المتبقية:**
    - كرر العملية للفاتورة #1235:
        - المرجع: "نقد مدفوع 10-02-2025 - إيصال #001"
    - كرر للفاتورة #1236:
        - المرجع: "نقد مدفوع 10-02-2025 - إيصال #001"
5. **التحقق من المعاملات:**
    - انتقل إلى قسم **Transactions**
    - تم إنشاء ثلاث معاملات دفع جديدة:
    - جميعها مرتبطة بالفواتير المعنية
6. **تحديث الإحصائيات:**
    - قسم **Invoices** → انخفض إجمالي **الفواتير غير المدفوعة** بمقدار 300.00$
    - 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 أن يتم فوترتها

إنشاء الفواتير:

- أنشئ الفواتير على فترات منتظمة (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
- رد الأموال عبر Stripe (أنشئ ائتمانًا يدويًا 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 فورًا إذا ظهرت الفاتورة بشكل غير صحيح

مراجعة المعاملات:

- راجع تاريخ المعاملات بانتظام
- أبلغ عن أي رسوم غير متوقعة على الفور
- افهم أي المعاملات مفوترة مقابل غير مفوترة

للمسؤولين

تكوين النظام:

- قم بتكوين قوالب بريد Mailjet لتسليم الفواتير بشكل احترافي
- قم بتخصيص قوالب PDF للفواتير لتناسب 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 ينخفض 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 تلقائيًا للفوترة المتكررة (إذا كانت متاحة)
- قم بتكوين تذكيرات عبر البريد الإلكتروني للفواتير المتأخرة

المشكلات الشائعة والحلول

المشكلة: العميل لا يستطيع دفع الفاتورة

الأعراض:

- ينقر العميل على زر الدفع لكن لا يحدث شيء
- رسالة خطأ: "لا توجد طرق دفع متاحة"

التشخيص:

1. انتقل إلى العميل ← علامة **Billing** ← قسم **Payment Methods**
2. تحقق مما إذا كان لدى العميل أي طرق دفع محفوظة
3. تحقق مما إذا كانت البطاقات المحفوظة منتهية

الحل:

- يجب على العميل إضافة طريقة دفع صالحة قبل دفع الفواتير
- ارشد العميل إلى صفحة **Payment Methods** لإضافة بطاقة ائتمان
- تحقق من قبول البطاقة (Visa، Mastercard، Amex، إلخ)
- أعد المحاولة للدفع بعد إضافة البطاقة

المشكلة: تم إنشاء فاتورة بمعاملات خاطئة

الأعراض:

- تتضمن الفاتورة معاملات من فترة خاطئة
- الفاتورة مفقودة المعاملات المتوقعة
- إجمالي الفاتورة غير صحيح

التشخيص:

1. افتح الفاتورة في قسم **Invoices**
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 أن المعاملات غير مفوترة، وأعد إنشائها مع نطاق التاريخ الصحيح
- **إذا كانت الفاتورة مدفوعة:** لا يمكن إلغاؤها - أنشئ معاملة ائتمان للمبلغ غير الصحيح، وأنشئ فاتورة مصححة
- **الوقاية:** دائمًا راجع قسم **Transactions** قبل إنشاء الفاتورة للتحقق من أن المعاملات الصحيحة ستتضمن

المشكلة: إجمالي المعاملات غير المفوترة لا ينخفض

الأعراض:

- يظهر عنصر واجهة المستخدم **Uninvoiced Transactions** مبلغًا مرتفعًا
- تظهر قائمة المعاملات العديد من المعاملات غير المفوترة
- تم إنشاء فواتير شهرية بالفعل

التشخيص:

1. قم بتصفية المعاملات حسب **Invoice Status: Not Invoiced**
2. راجع قائمة ال��عاملات 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 في إجمالي 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 يكن كذلك

المشكلة: طريقة الدفع ترفض 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 على أنها مفقودة/مسروقة
    - منع الاحتيال من البنك
    - بطاقة دولية محجوزة

الحل:

- اطلب من العميل الاتصال بالبنك الخاص به 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 الفوترة للبطاقة

اختصارات لوحة المفاتيح والنصائح

التنقل:

- الوصول إلى علامة الفواتير بسرعة: صفحة العميل → مفتاح **B** (إذا كانت اختصارات لوحة المفاتيح مفعلة)

التصفية:

- قبل إنشاء الفواتير لرؤية ما سيتم تضمينه بالضبط **المعاملات:** انقر على فلتر **"Not Invoiced"**

- لرؤية جميع الفواتير المستحقة التي **"Not yet Paid" الفواتير:** انقر على فلتر تتطلب متابعة

العمليات الجماعية:

- حدد فواتير متعددة (مربع الاختيار) لحذف/إلغاء عدة فواتير مرة واحدة (موظف فقط)
- استخدم شريط البحث للعثور بسرعة على معاملة أو فاتورة معينة حسب المعرف

الإجراءات السريعة:

- انقر على عنوان المعاملة/الفاتورة لعرض التفاصيل الكاملة في نافذة
- قائمة إجراءات النقر بزر الماوس الأيمن (⋮) للوصول السريع إلى خيارات التنزيل/البريد الإلكتروني/الدفع

الوثائق ذات الصلة

- `basics_payment` - طرق الدفع وتكامل الدفع متعدد البائعين
- `payment_system_guide` - مرجع واجهة برمجة تطبيقات الدفع وتكوين البائع
- `payments_transaction` - المعاملات بالتفصيل
- `payments_invoices` - الفواتير بالتفصيل
- `integrations_mailjet` - تسليم الفواتير عبر البريد الإلكتروني
- `csa_activity_log` - عرض تاريخ نشاط الفواتير
- Monitoring & Metrics - مقاييس نظام فوترة OCS/CGRateS والمراقبة

سلوكيات CGRateS والإجراءات

تشرح هذه الوثيقة كيفية عمل إجراءات CGRateS داخل OmniCRM، مع التركيز بشكل خاص على إدارة الرصيد، وسلوكيات الشحن، 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 بأرصدة CGRateS عند توفير منتج إضافي أو شحن، فإنك تقوم فعليًا بتنفيذ إجراء Playbooks يمكنك اتباع طرق أخرى أيضًا، مثل إضافة الأرصدة يدويًا إلى الحسابات من الحساب. (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 (مثل "حزمة بيانات"، "دقائق صوت")

الوزن - الأولوية لاستهلاك الرصيد (يتم استهلاك الوزن الأعلى أولاً)

وقت الانتهاء - متى ينتهي الرصيد (تاريخ مطلق أو نسبي مثل "+5 أيام")

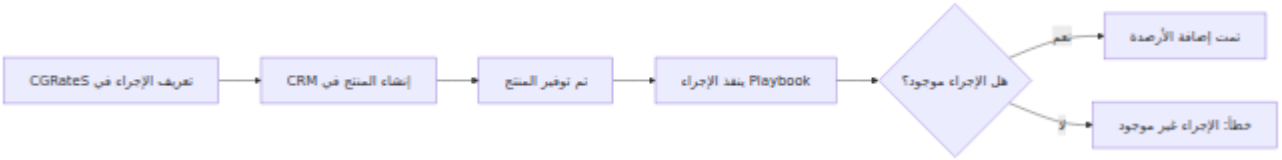
الحاجز - علامة رصيد خاصة تمنع جميع الاستخدامات عندما يصل الرصيد إلى الصفر، حتى لو كانت هناك أرصدة أخرى موجودة (انظر حواجز الرصيد)

مهم: يجب تعريف الإجراءات أولاً

قبل أن تتمكن من تنفيذ إجراء في Playbook باستخدام `ExecuteAction`، يجب أن يكون هذا الإجراء قد تم تعريفه بالفعل في CGRateS. هذه شرط أساسي حاسم غالبًا ما يتم تجاهله.

متى يتم تعريف الإجراءات

عادةً ما يتم تعريف الإجراءات خلال **تكوين النظام الأولي** أو **إعداد المنتج**، وليس أث��اء التوفير. يتم إنشاؤها عادةً عبر نصوص بايثون التي تقوم بتكوين كل من CRM و OCS في وقت واحد.



كيفية ربط الإجراءات بالمنتجات

ترتبط الإجراءات بالمنتجات عبر قاعدة 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 هذا الإجراء موجودًا في CGRateS، فإن التوفير يفشل.

أين يتم تعريف الإجراءات

يجب تعريف الإجراءات في نصوص تكوين المنتج الخاصة بك، عادةً:

1. **أثناء الإعداد الأولي** - عند تكوين النظام لأول مرة
2. **عند ��نشاء منتجات جديدة** - تعريف الإجراء قبل إنشاء المنتج
3. **عبر نصوص التكوين** - نصوص 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:

```
# في Playbook
- name: تنفيذ الإجراء
  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: الإجراء غير موجود"
# 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:
1 الرصيد:
  ID: "Data_5GB_5days_uuid_abc123"
  Type: *data
  Value: 5368709120  # 5GB بالبايت
  Expiry: 2024-12-29
  Weight: 10

2 الرصيد:
  ID: "Data_10GB_30days_uuid_def456"
  Type: *data
  Value: 10737418240  # 10GB بالبايت
  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 CGRateS عندما يكون للأرصدة نفس الوزن وتتطابق مع نفس الوجهة، لا يضمن بناءً على الانتهاء - يعتمد الترتيب على كيفية تخزين الأرصدة واسترجاعها. **استخدم أوزان مختلفة للتحكم في ترتيب الاستهلاك**.

أنواع الإجراءات: *topup مقابل *topup_reset

يحدد نوع الإجراء كيفية تفاعل الأرصدة الجديدة مع الأرصدة الموجودة بنفس المعرف.

*topup - سلوك إضافي

الإجراء `*topup` **يضيف إلى الأرصدة الموجودة** بنفس معرف الرصيد و**يمدد وقت الانتهاء**.

السلوك:

- يجد الرصيد الموجود مع المعرف المطابق
- يضيف القيمة الجديدة إلى القيمة الموجودة

- يحدث وقت الانتهاء إلى وقت الانتهاء الجديد
- يحافظ على الرصيد الموجود (الترحيل)

مثال:

```yaml
# الحالة الأولية:
الرصيد:
  ID: "Data_Package__5368709120"
  Value: 1073741824  # 1GB المتبقي
  Expiry: 2024-12-24 (1 يوم متبقي)

# مع تشغيل إجراء *topup:
الإجراء:
  Identifier: "*topup"
  Balance:
    ID: "Data_Package__5368709120"  # نفس المعرف - يحفز الترحيل
    Value: 5368709120  # 5GB
    ExpiryTime: "+5d"

# النتيجة بعد *topup:
الرصيد:
  ID: "Data_Package__5368709120"
  Value: 6442450944  # 6GB (1GB + 5GB تم ترحيلها)
  OriginalValue: 5368709120  # 5 يزال يظهر 5GB الأصلية
  Value_hr: "6 GB"
  OriginalValue_hr: "5 GB"
  Remaining_hr: "6 GB 5 من GB (1 GB تم ترحيله)"
  Expiry: 2024-12-29 (5 أيام من الآن)
```

حالات الاستخدام:

- مكافآت الولاء (إضافة بيانات إضافية إلى الحزمة الموجودة)
- أرصدة التعويض (إضافة إلى رصيد العميل)
- حزم البيانات القابلة للترحيل
- تمديد فترات السماح

*topup_reset - سلوك إعادة التعيين

الإجراء `*topup_reset` بنفس معرف الرصيد **يستبدل الأرصدة الموجودة**.

السلوك:

- يجد الرصيد الموجود مع المعرف المطابق
- يتجاهل القيمة القديمة (لا ترحيل)
- يحدد الرصيد إلى القيمة الجديدة فقط
- يحدث وقت الانتهاء إلى وقت الانتهاء الجديد

مثال:

```
# الحالة الأولية:
الرصيد:
  ID: "Data_Package__5368709120"
  Value: 1073741824  # 1GB المتبقي
  Expiry: 2024-12-24 (1 يوم متبقي)

# مع تشغيل إجراء *topup_reset:
الإجراء:
  Identifier: "*topup_reset"
  Balance:
    ID: "Data_Package__5368709120"  # نفس المعرف - يحفز إعادة التعيين
    Value: 5368709120  # 5GB
    ExpiryTime: "+5d"

# النتيجة بعد *topup_reset:
الرصيد:
  ID: "Data_Package__5368709120"
  Value: 5368709120  # 5GB (1 تجاهل GBتم القديمة)
  OriginalValue: 5368709120
  Value_hr: "5 GB"
  Remaining_hr: "5 GB 5 من GB"
  PercentUsed: 0
  Expiry: 2024-12-29 (5 أيام من الآن)
```

حالات الاستخدام:

- حزم متكررة شهرية (إعادة تعيين إلى المبلغ الكامل كل شهر)
- شحنات ثابتة الحجم (الحصول دائمًا على المبلغ الدقيق)
- تغييرات الخطة (استبدال رصيد الخطة القديمة بالخطة الجديدة)
- منع الإساءة (لا يمكن تجميع إضافات غير محدودة)

التحكم في سلوك الرصيد باستخدام معرفات الرصيد

معرف الرصيد أمر حاسم لتحديد ما إذا كانت الأرصدة مستقلة أو تتفاعل مع بعضها البعض.

قاعدة تسمية معرف الرصيد وعرضه القابل للقراءة البشرية

يستخدم OmniCRM قاعدة تسمية محددة لمعرفات الرصيد التي ترمز إلى كل من الاسم الوصفي والحجم الأصلي. وهذا 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: "__" (شرطة مزدوجة)
# - الحجم الأصلي: "107374182400" (100GB بالبايت)
# - واجهة المستخدم تظهر: "AU Data Domestic - 100 GB"

# 3000 دقيقة :رصيد الصوت
"AU_Voice_Domestic__180000000000000"
# - الجزء الوصفي: "AU_Voice_Domestic" (ليس
"AU_Voice_Domestic_3000min")
# - الحجم الأصلي: "180000000000000" (3000 دقيقة بالنانوسكند)
# - 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 تتضمن معلومات الحجم في الجزء الوصفي - فهي زائدة عن الحاجة لأن الحجم مشفر بعد `__` و API تقوم بتحويله إلى تنسيق قابل للقراءة البشرية تلقائيًا.

كيف ينشئ API عرضًا قابلًا 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 تم إنشاؤها تلقائيًا:
        "ID_hr": "AU Data Domestic",
        "OriginalValue": 107374182400,
        "OriginalValue_hr": "100 GB",
        "Value_hr": "50 GB",
        "Remaining_hr": "50 GB 100 من GB",
        "PercentUsed": 50,
        "ExpiryTime_hr": "25 (يوم 22) 2025 يناير"
      }
    ]
  }
}
```

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 بمسافات
id_hr = descriptive_name.replace("_", " ")  # "AU Data
Domestic"
```

3. **تحويل الحجم الأصلي إلى وحدات بشرية:**

```
# لأرصدة البيانات (بايت)
if balance_type == "*data":
    original_value_hr = convert_bytes_to_gb(original_size)
# "100 GB"

# لأرصدة الصوت (نانوسكند)
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} من
{original_value_hr}"
# "50 GB 100 من 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 100 من GB" */}
  </Progress>
  <Expiry>{balance.ExpiryTime_hr}</Expiry> {/* "25 22) 2025 يناير (يوم" */}
</BalanceCard>
```

لماذا هذا مهم:

1. **تتبع الحجم الأصلي** - حتى عندما يتم استهلاك الأرصدة جزئيًا، يمكن للواجهة أن تظهر "50 GB من 100 GB" بدلاً من "50 GB المتبقية" فقط
2. **تصور التقدم** - تمكن حسابات النسبة من شريط تقدم دقيق
3. **تسمية متسقة** - تضمن الأسماء الوصفية المستخرجة من معرفات الرصيد التناسق بين الخلفية والواجهة الأمامية
4. **عرض الترحيل** - عند استخدام `*topup` (الترحيل)، 70 GB إذا كان لدى العميل 100 GB المتبقية وقام بشحن:
   - يبقى معرف الرصيد: `"AU_Data_Domestic__107374182400"` (100 GB الأصلية)
   - تصبح القيمة الحالية: 170 GB
   - تظهر الواجهة: "170 GB (70 GB 100 + تم ترحيله GB جديدة)"

أفضل ممارسة - إنشاء معرفات الرصيد:

تأكد دائمًا من تضمين 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 تُشحن إلى أي مبلغ:

```
# رصيد نقدي بدون تشفير الحجم
{
    "BalanceId": "PAYG_Monetary_Balance",
    "BalanceType": "*monetary",
    "Units": 5000  # $50.00
}

# تعرض الواجهة ببساطة الرصيد الحالي دون النسبة
# "50.00$ :الرصيد"
```

الاستراتيجية 1: معرفات فريدة (أرصدة مستقلة)

استخدم معرفات فريدة (مثل UUIDs) لإنشاء أرصدة مستقلة تمامًا لا تتفاعل أبدًا.

تنفيذ المثال:

```
- name: توليد معرف رصيد فريد
  set_fact:
    uuid: "{{ 99999999 | random | to_uuid }}"
    balance_id: "Data_5days__5368709120_{{ uuid[0:8] }}"

- name: إضافة رصيد مستقل مع *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 منفصلًا حتى لو اشترى العميل نفس الإضافة عدة مرات.

```
# العميل يشتري "إضافة بيانات 5 أيام" ثلاث مرات:
1 الرصيد:
  ID: "Data_5days__5368709120_a1b2c3d4"
  Value: 5368709120
  Value_hr: "5 GB"
  OriginalValue_hr: "5 GB"
  Remaining_hr: "5 GB 5 من GB"
  Expiry: 2024-12-29

2 الرصيد:
  ID: "Data_5days__5368709120_e5f6g7h8"
  Value: 5368709120
  Value_hr: "5 GB"
  Remaining_hr: "5 GB 5 من GB"
  Expiry: 2024-12-30

3 الرصيد:
  ID: "Data_5days__5368709120_i9j0k1l2"
  Value: 5368709120
  Value_hr: "5 GB"
  Remaining_hr: "5 GB 5 من GB"
  Expiry: 2024-12-31

# 15 :إجمالي المتاح GB عبر ثلاثة أرصدة منفصلة
# 5 - أيام 5 "بيانات" كـ الواجهة في فردي بشكل منها يظهر GB كل GB 5 من GB"
```

الاستراتيجية 2: معرفات مشتركة مع topup* (الترحيل)

استخدم نفس معرف الرصيد مع إجراء `topup*` للسماح بترحيل الرصيد وتمديد الانتهاء.

تنفيذ المثال:

```
- name: تعيين معرف رصيد ثابت
  set_fact:
    balance_id: "Data_5days__5368709120"

- name: إضافة رصيد مع *topup (الترحيل)
  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 وتطيل مدة الانتهاء.

```yaml
# "اليوم 1: العميل يشتري "إضافة بيانات 5 أيام
الرصيد:
  ID: "Data_5days__5368709120"
  Value: 5368709120
  Value_hr: "5 GB"
  Remaining_hr: "5 GB 5 من GB"
  PercentUsed: 0
  Expiry: 2024-12-29

# ثم اشترى نفس الإضافة مرة أخرى، GBاليوم 3: العميل استخدم 1 GB المتبقية + 4GB 5 جديدة
الرصيد:
  ID: "Data_5days__5368709120"
  Value: 9663676416  # 4GB 5 + المتبقيةGB جديدة
  Value_hr: "9 GB"
  OriginalValue_hr: "5 GB"
  Remaining_hr: "9 GB (4 GB 5 + تم ترحيله 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.

تنفيذ المثال:

```
- name: تعيين معرف رصيد ثابت
  set_fact:
    balance_id: "Monthly_Plan__32212254720"

- name: إضافة رصيد مع *topup_reset (بدون ترحيل)
  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 بالضبط بغض النظر عن مقدار الاستخدام.

```
# 1 اليوم ،1 الشهر:
الرصيد:
  ID: "Monthly_Plan__32212254720"
  Value: 32212254720
  Value_hr: "30 GB"
  Remaining_hr: "30 GB 30 من GB"
  PercentUsed: 0
  Expiry: 2024-12-31

# 28 العميل استخدم :25 اليوم ،1 الشهرGB
الرصيد:
  ID: "Monthly_Plan__32212254720"
  Value: 2147483648
  Value_hr: "2 GB"
  Remaining_hr: "2 GB 30 من GB"
  PercentUsed: 93
  Expiry: 2024-12-31

# 1 اليوم ،2 الشهر: ActionPlan يعمل *topup_reset
الرصيد:
  ID: "Monthly_Plan__32212254720"
  Value: 32212254720  # 2 فقدان تم ،الكامل إلى تعيين إعادةGB المستخدمة غير
  Value_hr: "30 GB"
  Remaining_hr: "30 GB 30 من GB"
  PercentUsed: 0
  Expiry: 2025-01-31
```

حواجز الرصيد

تسمح لك **بمنع أو تقييد الاستخدام** حتى عند CGRateS **حواجز الرصيد** هي ميزة قوية في وجود أرصدة. يتوقف رصيد الحاجز عن الاستهلاك عندما يصل إلى الصفر، مما يمنع المزيد من الاستخدام بغض النظر ع�ن الأرصدة المتاحة الأخرى.

كيف تعمل الحواجز

عندما يحتوي الرصيد على `Blocker: true`:

1. **بينما يحتوي الحاجز على قيمة** ← يُسمح بالاستخدام حتى الحد الأقصى للحاجز

2. **عندما يصل الحاجز 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 **فحص أرصدة الحاجز حتى عندما تكون القيمة صفر** (على عكس الأرصدة العادية التي يتم تخطيها)
- **يتوقف CGRateS** عند مواجهة حاجز مع طلب استخدام متبقي، ويعيد **عن المعالجة** خطأ
- تعمل الحواجز مع جميع أنواع الأرصدة: `*voice`، `*data`، `*sms`، `*monetary`

حالات الاستخدام للحواجز

1. تعليق الحساب

قم بحظر جميع الاستخدامات عندما يتم تعليق الحساب (مثل، فشل الدفع):

```
Action_Suspend_Account = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_suspend-account",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            # إضافة حاجز بقيمة صفرية لمنع جميع الاستخدامات
            {
                "Identifier": "*topup",
                "BalanceId": "Suspension_Blocker",
                "BalanceType": "*monetary",
                "DestinationIDs": "*any",
                "Units": 0,  # قيمة صفرية
                "BalanceWeight": 9999,  # أعلى أولوية - يتم فحصه أولاً
                "Blocker": True,  # يمنع جميع الاستخدامات
                "Weight": 10
            }
        ]
    }]
}

result = OCS_Obj.SendData(Action_Suspend_Account)
```

النتيجة: يتم حظر جميع المكالمات/البيانات/SMS الأخرى بغض النظر عن الأرصدة.

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,  # يتم استهلاكه أولاً
                "Weight": 95
            },
            # (حاجز) 50 دولار حد إضافي قدره
            {
                "Identifier": "*topup_reset",
                "BalanceId": "Overage_Cap",
                "BalanceType": "*monetary",
                "DestinationIDs": "*any",
                "Units": 5000,  # 50.00 دولار كحد أقصى
                "ExpiryTime": "*month",
                "BalanceWeight": 1000,  # يتم استهلاكه بعد المضمن
                "Blocker": True,  # 50 دولار يتوقف عند إنفاق
                "Weight": 90
            }
        ]
    }]
}
```

التدفق:

1. 10 العميل يستخدمGB مجانية ← المضمنة (من Included_Data_10GB)
2. 5 العميل يستخدمGB من يتم خصمها ← إضافية Overage_Cap (أسعار PAYG)

3. عندما يصل Overage_Cap (تم الوصول إلى حد الإنفاق) → 0 إلى **يتم حظر جميع الاستخدامات**

3. فترة تجريبية محدودة زمنياً

توفير استخدام مجاني محدود للحسابات التجريبية:

```
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 الاستنفاد)
            {
                "Identifier": "*topup",
                "BalanceId": f"Trial_Voice__{100 * 60 *
1000000000}",
                "BalanceType": "*voice",
                "DestinationIDs": "Dest_Domestic_All",
                "Units": 100 * 60 * 1000000000,  # 100 دقيقة
                "ExpiryTime": "+720h",  # 30 يوم
                "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 محددة

حظر وجهات ��حددة مع السماح بأخرى:

```
Action_Block_Premium_Numbers = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_block-premium",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            # السماح بالاستخدام العادي
            {
                "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 فحصه أولاً للأرقام المميزة
                "Blocker": True,
                "Weight": 10
            }
        ]
    }]
}
```

التدق:

- تستخدم رصيد → المكالمات المحلية Regular_Usage
- تطابق → المكالمات إلى الأرقام المميزة Premium_Blocker (الوزن 2000 > 1000) → **محظور**

الحاجز مقابل الأرصدة المعطلة

لا تخلط بين `Blocker` و `Disabled`:

الميزة	حاجز	معطل
الغرض	توقف الاستخدام عند استنفاد الرصيد	تعليق رصيد مؤقت
عندما تكون القيمة 0 <	الرصيد قابل للاستخدام بشكل طبيعي	يتم تخطي الرصيد
عندما تكون القيمة 0 =	**يمنع جميع الاستخدامات الإضافية**	يتم تخطي الرصيد (يتم تجربة الرصيد التالي)
حالة الاستخدام	حدود الإنفاق، الحدود، حدود التجربة	تعليق رصيد معين مؤقتًا

```
# 10 يسمح بـ :حاجزGB، ثم يحظر كل شيء
{
    "BalanceId": f"Data_Cap__{10 * 1024 * 1024 * 1024}",
    "Units": 10 * 1024 * 1024 * 1024,
    "Blocker": True  # 10 عند الاستخدام GBيتوقف
}

# (مؤقتًا متوقف) يتجاهل هذا الرصيد تمامًا :معطل
{
    "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

اجمع بين الأرصدة الوحدوية، والنفقات النقدية الزائدة، 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
            },
            # حد إضافي قدره 20 دولار
            {
                "Identifier": "*topup_reset",
                "BalanceId": "Overage_Allowance",
                "BalanceType": "*monetary",
                "DestinationIDs": "*any",
                "Units": 2000,  # 20.00 دولار
                "ExpiryTime": "*month",
                "BalanceWeight": 1000,
                "Weight": 90
            },
            # حد صارم قدره 50 دولار (حاجز)
            {
                "Identifier": "*topup_reset",
                "BalanceId": "Hard_Spending_Cap",
                "BalanceType": "*monetary",
                "DestinationIDs": "*any",
                "Units": 5000,  # 50.00 دولار كحد أقصى
                "ExpiryTime": "*month",
```

```
                "BalanceWeight": 500,  # أقل من الزائد - يستخدم آخر
                "Blocker": True,  # يتوقف عند الوصول إلى الحد
                "Weight": 85
            }
        ]
    }]
}
```

رحلة العميل:

1. Domestic_Voice_500min (مجاني) يستخدم **0-500 دقيقة:**
2. بسعر 0.10 دولار/دقيقة = Overage_Allowance 20 يستخدم **500-700 دقيقة:** دولار (200 دقيقة)
3. بسعر 0.10 دولار/دقيقة = Hard_Spending_Cap 50 يستخدم **عند 700 دقيقة:** دولار (500 دقيقة)
4. Hard_Spending_Cap → **يتم حظر جميع** تم استفاد**** **عند 1200 دقيقة:** **الاستخدامات**

يحصل العميل على 1200 دقيقة إجمالية، مع حد إنفاق أقصى قدره 50 دولار.

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 على الفور
   "Blocker": True
   ```

3. **إخطار العملاء قبل استنفاد الحاجز**

   - لإرسال إشعارات عند 80%، 90%، 100% من ActionTriggers استخدم 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. **اختبر سلوك الحاجز**

   - تحقق من أن الحاجز يعيد خطأ `INSUFFICIENT_CREDIT_BALANCE_BLOCKER`
   - تظهر التكلفة = 1.0- عند الحظر تأكد من أن CDRs
   - اختبر أن الأرصدة الأخرى لا تُستخدم بعد استنفاد الحاجز

استكشاف الأخطاء وإصلاحها للحواجز

المشكلة: الاستخدام غير محظور على الرغم من أن الحاجز عند الصفر

الأسباب المحتملة:

1. وزن الحاجز منخفض جدًا (يتم فحص 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:
# - Blocker: true
# - BalanceWeight (مثل 9999) هو الأعلى
# - DestinationIDs تتضمن وجهة الاستخدام
# - Value = 0
```

المشكلة: الاستخدام محظور بشكل غير متوقع

السبب المحتمل: تم إنشاء رصيد الحاجز عن غير قصد بقيمة منخفضة

الحل: تحقق من جميع الأرصدة لـ `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 الوجهة الأكثر تحديدًا) أولاً. يتم تحديد ذلك بواسطة طول تطابق البادئة.

```
# العميل يتصل بـ +44-20-1234-5678 (لندن، المملكة المتحدة)

الرصيد 1:
  DestinationIDs: "Dest_UK_London"  # بادئة: "4420" (الدقة: 4)
  Value: 100 دقيقة
  Weight: 10

الرصيد 2:
  DestinationIDs: "Dest_UK_All"     # بادئة: "44" (الدقة: 2)
  Value: 200 دقيقة
  Weight: 10

# يتم استهلاك الرصيد 1 أولاً (الدقة 4 > الدقة 2)
# تطابق الوجهة الأكثر تحديدًا يفوز
```

حالة الاستخدام: تفضل الأرصدة الخاصة بالمدينة أو المنطقة على الأرصدة الخاصة بالبلد.

القاعدة 2: أولوية الوزن (نفس الدقة)

عندما تكون دقة الوجهة متساوية، يتم استهلاك الأرصدة ذات **الوزن الأعلى** أولاً.

```
1 الرصيد:
  ID: "Premium_Data"
  Value: 5GB
  Weight: 20

2 الرصيد:
  ID: "Standard_Data"
  Value: 10GB
  Weight: 10

# (10 الوزن < 20 (الوزن 1 أولاً الرصيد استهلاك يتم
# على الرغم من أن الرصيد 2 يحتوي على مزيد من البيانات
```

حالة الاستخدام: يتم استهلاك الأرصدة ذات الأولوية (بيانات إضافية يتم استهلاكها قبل البيانات العادية).

القاعدة 3: الأقدم أولاً

عندما يتطابق الوزن، يتم استخدام الرصيد الأقدم أولاً.

```
1 الرصيد:
  ID: "Data_Package_A"
  Value: 5GB
  Expiry: 2024-12-25
  Weight: 10

2 الرصيد:
  ID: "Data_Package_B"
  Value: 10GB
  Expiry: 2025-01-15
  Weight: 10
```

للتحكم في ترتيب الاستهلاك، استخدم أوزانًا مختلفة:

```
# النهج الصحيح: استخدم الوزن لإعطاء الأولوية للأرصدة التي ستنتهي قريبًا
1 الرصيد:
  ID: "Data_Package_A"
  Value: 5GB
  Expiry: 2024-12-25
  Weight: 11  # وزن أعلى = يتم استهلاكه أولاً

2 الرصيد:
  ID: "Data_Package_B"
  Value: 10GB
  Expiry: 2025-01-15
  Weight: 10  # وزن أقل = يتم استهلاكه ثانيًا

# يتم استهلاك الرصيد 1 أولاً (الوزن 11 > الوزن 10)
```

أفضل ممارسة: إذا كنت ترغب في التأكد من استهلاك الأرصدة التي ستنتهي قريبًا أولاً، قم بتعيين أوزان أعلى لها عند إنشائها.

أمثلة عملية

المثال 1: إضافة بيانات بسيطة (مستقلة)

السيناريو: يمكن للعميل شراء إضافة 5GB 5 أيام" عدة مرات، كل منها ينشئ رصيدًا منفصلًا.

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 5 GB تنتهي في 29 ديسمبر
- يشتري الإضافة في 25 ديسمبر GB 5 ← يحصل على 5 GB تنتهي في 30 ديسمبر
- كلا الرصيدين يتواجدان؛ يعتمد ترتيب الاستهلاك على أوزان الرصيد (كلاهما له وزن 10، لذا فإن الترتيب غير مضمون)

المثال 2: حزمة بيانات قابلة للترحيل

السيناريو: خطة شهرية 50 GB" حيث يتم ترحيل البيانات غير المستخدمة عندما يقوم العميل بشحن مبكرًا.

التنفيذ:

```
- name: إضافة رصيد بيانات قابلة للترحيل
  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 يوم
            "Weight": 10
          }
        }]
      }
```

(الترحيل مفعل) `*topup` **نوع الإجراء:** يستخدم السلوك الافتراضي

تجربة العميل:

- اليوم 1: يحصل على 50GB تنتهي في 30 يوم
- اليوم 20: استخدم 30GB، لديه 20GB المتبقية
- اليوم 20: يشحن مرة أخرى → يحصل على 70GB إجمالي (20GB + 50GB)، يتم تمديد الانتهاء إلى +30 يوم من اليوم 20

المثال 3: خطة شهرية ثابتة (بدون ترحيل)

السيناريو: خطة "100GB شهرية غير محدودة" تعيد تعيينها إلى 100GB بالضبط كل شهر، بدون ترحيل.

التنفيذ:

```
- name: إنشاء إجراء إعادة تعيين شهري
  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 العمل 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 العمل 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:

- الشهر 1: يحصل على 100 GB، يستخدم 95 GB، لديه 5 GB المتبقية
- الشهر 2: يتم إعادة تعيين الرصيد إلى 100 GB (تم فقدان 5 GB غير المستخدمة)
- الشهر 2: يستخدم 20 GB، لديه 80 GB المتبقية
- الشهر 3: يتم إعادة تعيين الرصيد إلى 100 GB (تم فقدان 80 GB غير المستخدمة)

المثال 4: أرصدة متعددة الطبقات مع أولوية الوزن

السيناريو: لدى العميل "بيان��ت مكافأة" (أولوية عالية) و"بيانات عادية" (أولوية منخفضة). يتم استهلاك بيانات المكافأة أولاً.

التنفيذ:

```yaml
# إضافة بيانات المكافأة بوزن عالٍ
- name: إضافة رصيد بيانات مكافأة
  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": "Standard_Data",
            "Value": 53687091200,  # 50GB
            "ExpiryTime": "+720h",  # 30 يوم
            "Weight": 10  # أولوية عادية
          }
        }]
      }
```

تجربة العميل:

- بيانات عادية (وزن 10) GBبيانات مكافأة (وزن 20) + 50 GBلديه 5
- يتم استهلاك جميع بيانات المكافأة أولاً
- ثم يتم الاستهلاك من مجموعة بيانات 50 GB
- يتم الحفاظ على البيانات العادية لفترة أطول

الإجراءات الشائعة

يدعم CGRateS عدة معرفات إجراءات لمختلف العمليات:

التلاعب بالرصيد

***topup** - إضافة إلى الرصيد الموجود (الترحيل) - ***topup_reset** إعادة تعيين الرصيد إلى قيمة جديدة (بدون ترحيل) - ***debit** خصم من الرصيد - ***debit_reset** تعيين الرصيد إلى قيمة سالبة - ***reset_account** إزالة جميع الأرصدة

تصميم المنتجات الإضافية

عند تصميم المنتجات الإضافية في OmniCRM، ضع في اعتبارك هذه الأسئلة:

السؤال 1: هل يجب أن تتراكم الأرصدة؟

لا (استبدال) → استخدم UUID مع **نعم (مستقلة)** → ستخدم معرفات رصيد فريدة `*topup` **نعم (ترحيل)** → استخدم معرف رصيد ثابت مع `*topup_reset` معرف رصيد ثابت مع

السؤال 2: ماذا يحدث للرصيد غير المستخدم؟

مجموعات `*topup_reset` **فقد** → استخدم إجراء `*topup` **ترحيل** → استخدم إجراء
منفصلة → استخدم معرفات رصيد فريدة

السؤال 3: كيف يجب استهلاك الأرصدة؟

الأقدم أولاً → استخدم أوزانًا مختلفة (قم بتعيين وزن أعلى للأرصدة القديمة/التي ستنتهي قريبًا)
الأولوية أولاً → أوزان مختلفة (وزن أعلى = أولوية أعلى) **ترتيب محدد** → استخدم الأوزان: 30

(الأولوية)، 20 (المكافأة)، 10 (العادية) **عشوائي/بدون تفضيل** ← استخدم نفس الوزن (ترتيب الاستهلاك غير مضمون)

السؤال 4: ما هي استراتيجية الانتهاء؟

`*monthly` **نهاية الشهر** ← استخدم (لمدة 30 يوم `+720h`) **مدة ثابتة** ← استخدم انتهاء نسبي أو مدة طويلة جدًا `*unlimited` في خطة العمل **لا تنته�� أبدًا** ← استخدم

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 موجودًا
          "Actions": [
            {
              "Identifier": "*topup",  # أو *topup_reset
              "BalanceType": "*data",  # *data، *voice، *sms، *monetary
              "Units": 5368709120,  # المبلغ المراد إضافته
              "ExpiryTime": "+120h",  # +Xh، *unlimited، *monthly
              "DestinationIds": "*any",  # عادةً *any
              "BalanceId": "Balance_Name",  # فريد أو مشترك
              "Weight": 10,  # الأولوية (الأعلى = يتم استهلاكه أولاً)
              "Blocker": false,  # إذا كان صحيحًا، يمنع الرصيد السالب
              "Disabled": false,  # إذا كان صحيحًا، يتم تعطيل الرصيد
              "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 SMS (عدد)
- `*monetary` - أرصدة نقدية (وحدات العملة)
- `*generic` - أرصدة عامة

Units - المبلغ المراد إضافته/خصمه (بالوحدات الأساسية: بايت للبيانات، ثواني للصوت)

ExpiryTime - متى ينتهي الرصيد:

- `+Xh` - نسبي (`720+` مثل، `h` = 30 يوم)
- `*unlimited` - لا ينتهي أبدًا
- `*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 كان صحيحًا، يمنع الحساب من أن يصبح سالبًا

Disabled - إذا كان صحيحًا، يوجد الرصيد ولكن لا يمكن استخدامه

تعريف الإجراءات عبر بايثون (الإعداد الأولي)

عادةً ما يتم تعريف الإجراءات خلال التكوين الأولي للنظام باستخدام نصوص بايثون مع مكتبة `cgrateshttpapi`. تُظهر 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**/SMS - بادئات الأرقام للمكالمات الصوتية (مثل، إلى الأماكن `Dest_International_UK`)
- **وجهات PLMN** - رموز الشبكة للبيانات من الأماكن (مثل، `Dest_PLMN_OnNet`، `Dest_PLMN_US_Verizon`)

قاعدة حاسمة:

- **أرصدة الصوت/SMS** → استخدم الوجهات الجغرافية (الرقم الذي يتم الاتصال به)
- **أرصدة البيانات** → استخدم وجهات PLMN (الشبكة التي يتصل بها العميل)

للحصول على تكوين الوجهات الكامل بما في ذلك:

- الوجهات الجغرافية (محلية، دولية، مجانية، مميزة)
- وجهات PLMN (الشبكات المحلية، الشبكات المتجولة، المناطق)
- قواعد تنسيق PLMN وأفضل الممارسات
- استكشاف مشكلات الوجهة

انظر: **تكوين وجهات CGRateS**

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 (بالنانوسكند)
1_minute = 60 * 1000000000  # 60 مليار نانوثانية
3000_minutes = 3000 * 60 * 1000000000

# أرصدة SMS (بالعدد)
3000_sms = 3000
```

المثال 1: خطة شهرية متعددة الأرصدة (بايثون)

خطة شاملة شهرية تحتوي على بيانات وصوت وSMS وأرصدة تجوال 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: تستخدم أرصدة البيانات وجهات PLMN (الشبكة التي يتصل بها العميل)
            # الخاص بك PLMN ليست الوجهات الجغرافية. استخدم.
            {
                "Identifier": "*topup_reset",
                "BalanceId": "AU_Data_Domestic__" + str(100 * 1024 * 1024 * 1024),
                "BalanceType": "*data",
                "DestinationIDs": "Dest_PLMN_OnNet",  # PLMN المحلي 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 SMS 3000
```

```
            {
                "Identifier": "*topup_reset",
                "BalanceId": "AU_SMS_Domestic__" + str(3000),
                "BalanceType": "*sms",
                "DestinationIDs": "Dest_AU_Mobile;",
                "Units": 3000,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 88
            },
            # 6 إضافة رصيد بيانات تجوال GB
            {
                "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
            },
            # سجل 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")
```

النقاط الرئيسية:

- يستخدم `*reset_account` لمسح الأرصدة القديمة أولاً
- يستخدم `*topup_reset` لمخصصات شهرية ثابتة (بدون ترحيل)
- BalanceWeight يحدد ترتيب الاستهلاك (المحلي 1200 < التجوال 1100)
- Weight يحدد ترتيب التنفيذ داخل الإجراء
- يتضمن `*cdrlog` لتتبع التنشيطات

المثال 2: إضافة بيانات بسيطة (بايثون)

إضافة 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 إضافةGB بيانات
            # تستخدم أرصدة البيانات وجهات PLMN (الشبكة التي يتصل بها العميل)
            {
                "Identifier": "*topup_reset",
                "BalanceId": "AU_Data_Domestic__" + str(20 * 1024
* 1024 * 1024),
                "BalanceType": "*data",
                "DestinationIDs": "Dest_PLMN_OnNet",  # PLMN المحلي
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 أرصدة الصوت الوجهات الجغرافية (الرقم الذي يتم الاتصال به)، بينما SMS/ تستخدم البيانات وجهات PLMN (الشبكة التي يتصل بها العميل). انظر تعريف المنتجات لتكوين الوجهات.

مرجع حقول إجراء بايثون

حقول تعريف الإجراء:

- **ActionsId** (مطلوب) - معرف فريد لمجموعة الإجراءات هذه (يجب أن يتطابق مع اتفاقية product_slug في CRM)
- **Overwrite** - إذا كان صحيحًا، استبدل الإجراء الموجود بنفس المعرف
- **Tenant** - اسم مستأجر CGRateS
- **Actions** - مصفوفة من الإجراءات الفردية التي سيتم تنفيذها

حقول الإجراء الفردية:

- **Identifier** - نوع الإجراء (`*topup`، `*topup_reset`، `*cdrlog`، إلخ)
- **BalanceId** - معرف فريد لهذا الرصيد (يجب أن يتطابق عبر الشحنات لتعمل الترحيل)
- **BalanceType** - نوع الرصيد (`*data`، `*voice`، `*sms`، `*monetary`)
- **DestinationIDs** - يتحكم في مكان استخدام الرصيد:
  - **الصوت**/**SMS**: استخدم الوجهات الجغرافية (مثل، `"Dest_AU_Mobile"`، `"Dest_International_UK"`)
  - **البيانات**: استخدم وجهات PLMN (مثل، `"Dest_PLMN_OnNet"`، `"Dest_PLMN_US_Verizon"`)
- **Units** - المبلغ المراد إضافته (بالوحدات الأساسية: بايت للبيانات، نانوثانية للصوت)
- **ExpiryTime** - متى ينتهي الرصيد:
  - `+Xh` - نسبي (`h`مثل، `720+` = 30 يوم)
  - `*unlimited` - لا ينتهي أبدًا
  - `*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** (اختياري) - علامة منطقية؛ إذا كانت `True`، تمنع الحساب من أن يصبح سالبًا
- **Disabled** (اختياري) - علامة منطقية؛ إذا كانت `True`، يوجد الرصيد ولكن لا يمكن استخدا��ه

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 المعرفة في الإجراء على الحساب المحدد.

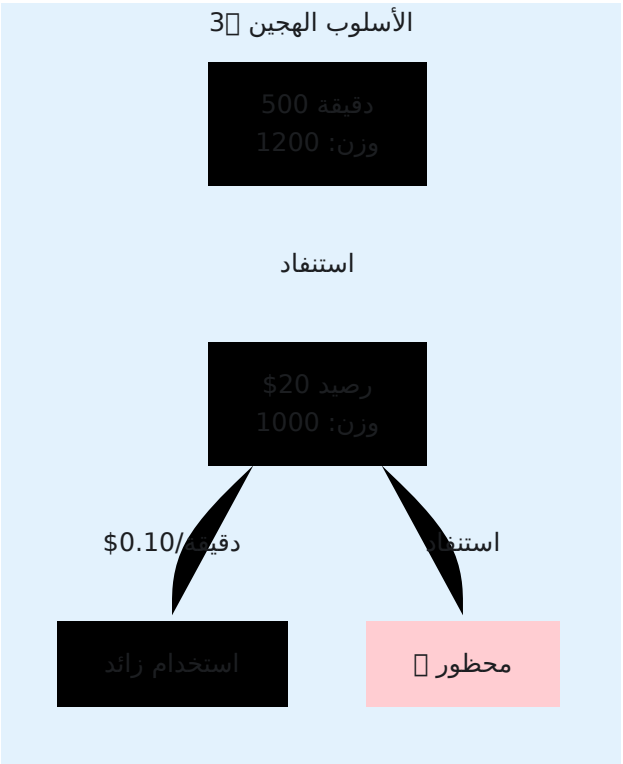
أساليب إدارة الرصيد

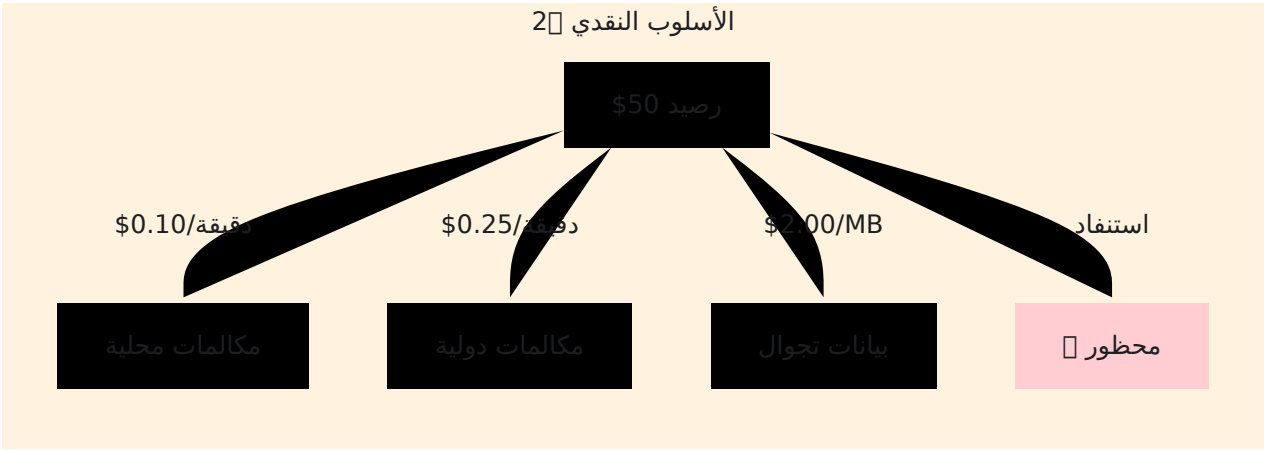
هناك ثلاث طرق رئيسية لإدارة الأرصدة في CGRateS، كل منها له مزايا وعيوب مختلفة من حيث تجربة العميل، وتوقعات الفواتير، وتعقيد العمليات.

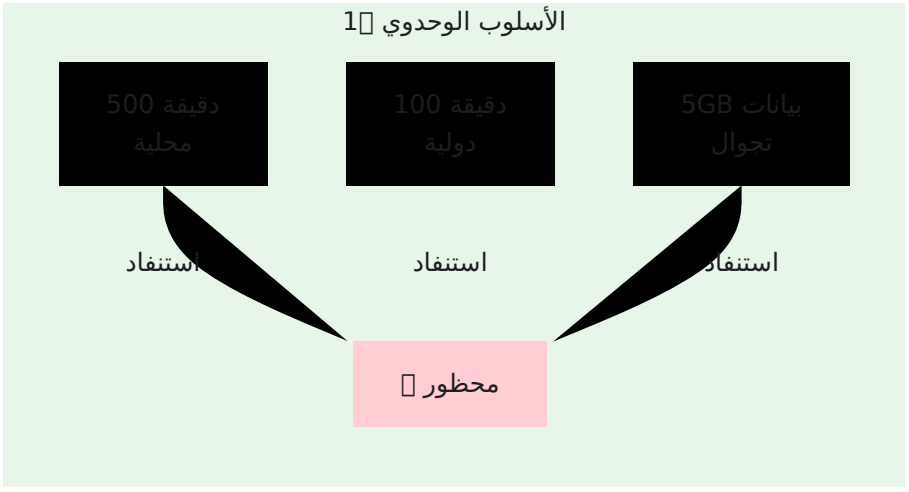
مقارنة: وحدوي مقابل نقدي مقابل هجينة

تلخص هذه الجدول المزايا والعيوب بين الثلاثة أساليب للرصيد:

الميزة	وحدوي	نقدي (PAYG)	هجينة
توقعات العميل	تكلفة ثابتة شهرية ✅	تكاليف متغيرة ❌	معظمها متوقعة ⚠️
مرونة الوجهة	❌ محدودة للوجهات المضمنة	✅ اتصل/استخدم في أي مكان	✅ مضمنة + في أي مكان
معالجة النفقات الزائدة	❌ قطع صارم	✅ استخدام تلقائي	✅ تدفق تلقائي
خطر صدمة الفاتورة	✅ منخفض (حدود صارمة)	❌ مرتفع (فواتير غير محدودة)	⚠️ معتدل (تدفق محدود)
تعقيد التكوين	⚠️ معتدل (العديد من الأرصدة)	✅ بسيط (رصيد واحد)	❌ معقد (كلاهما)
تحسين الإيرادات	⚠️ أقل ARPU	✅ أعلى من ARPU المستخدمين الثقيلين	✅ متوازن ARPU
رضا العميل	❌ مرتفع (لا مفاجآت)	✅ منخفض (صدمة الفاتورة)	✅ مرتفع (أفضل ما في كلا العالمين)
أفضل من أجل	المستخدمون المتوقعة	المستخدمون العرضيون	معظم العملاء


3️⃣ الأسلوب الهجين
500 دقيقة
وزن: 1200
استنفاد
$20 رصيد
وزن: 1000
$0.10/دقيقة
استنفاد
استخدام زائد
محظور ❌


2️⃣ الأسلوب النقدي
$50 رصيد
$0.10/دقيقة
$0.25/دقيقة
$2.00/MB
استنفاد
مكالمات محلية
مكالمات دولية
بيانات تجوال
محظور ❌


1️⃣ الأسلوب الوحدوي
500 دقيقة
محلية
100 دقيقة
دولية
5GB بيانات
تجوال
استنفاد
استنفاد
استنفاد
محظور ❌

الأسلوب 1: الأرصدة الوحدوية

المفهوم: توفير كميات محددة لوجهات محددة. كل رصيد له مبلغ ثابت (دقائق، GB، عدد SMS) مرتبط بوجهات محددة. عند الاستنفاد، يتم حظر الاستخدام ما لم يكن هناك احتياطي نقدي.

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",  # فقط محلية
                "Units": 500 * 60 * 1000000000,  # 500 دقيقة
بالنانوسكند
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 90
            },
            # 1000 SMS محلية
            {
                "Identifier": "*topup_reset",
                "BalanceId": "Domestic_SMS__1000",
                "BalanceType": "*sms",
                "DestinationIDs": "Dest_Domestic_All",
                "Units": 1000,
                "ExpiryTime": "*month
```

CGRateS تكوين وجهات

تشرح هذه الدليل كيفية تكوين الوجهات في CGRateS لـ OmniCRM. تحدد الوجهات أين يمكن استخدام الأرصدة - سواء كان ذلك بالاتصال بأرقام محددة (جغرافية) أو استخدام البيانات/الصوت على شبكات محددة (PLMN).

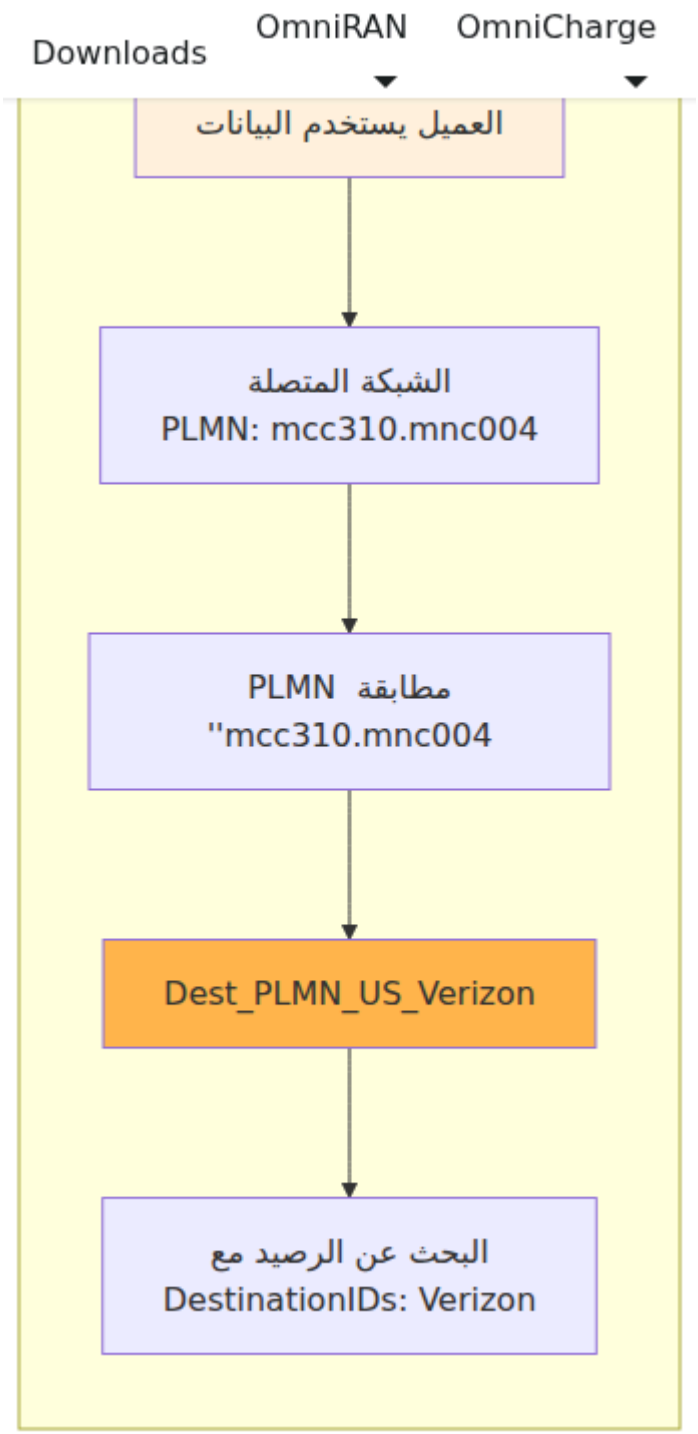
نظرة عامة

تستخدم CGRateS للتحكم في المكان الذي يمكن للعملاء استخدام أرصدتهم فيه **الوجهات**:

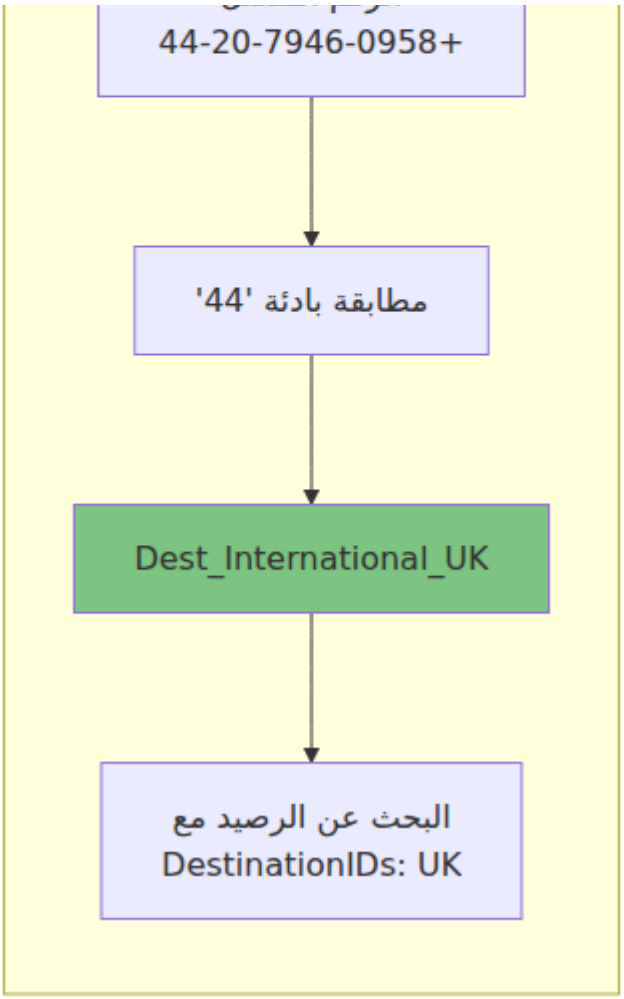
- **وجهات جغرافية**/SMS (مثل أرقام بادئات الأرقام للمكالمات - إلى مواقع محددة المملكة المتحدة، أرقام الولايات المتحدة المجانية)
- **وجهات PLMN** - رموز الشبكة لاستخدام البيانات والتجوال من شبكات محددة (مثل Verizon، Vodafone UK)

عندما يستخدم العملاء الخدمات، تطابق CGRateS:

- **الصوت/SMS**: بادئة الرقم المتصل ← وجهة جغرافية
- **البيانات**: رمز شبكة PLMN ← وجهة PLMN


Downloads
OmniRAN
OmniCharge
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 يقوم بإجراء مكالمة



القاعدة الحرجة

- استخدم الوجهات الجغرافية (الرقم الذي يتم الاتصال به) ← **أرصدة الصوت/SMS**
- استخدم وجهات PLMN (الشبكة التي يتصل بها العميل) ← **أرصدة البيانات**

ا��متطلبات 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 المنطقة)
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_Domestic_All",
        "Prefixes": ["1"]  # NANP (الولايات المتحدة/كندا)
    }]
})

# الولايات المتحدة مجانية
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_Domestic_TollFree",
        "Prefixes": ["1800", "1888", "1877", "1866", "1855",
"1844", "1833"]
    }]
})

# الولايات المتحدة - سعر مرتفع
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_Domestic_Premium",
        "Prefixes": ["1900"]
    }]
})
```

الوجهات الدولية

```
#
# ============================================================================
# الوجهات الدولية (حسب البلد)
#
# ============================================================================

# 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 - مجانية
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_AU_TollFree",
        "Prefixes": [
            "611800",  # مجانية
            "611300"   # سعر محلي
        ]
    }]
})

# أستراليا - سعر مرتفع
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_AU_Premium",
        "Prefixes": ["6119"]  # خدمات سعر مرتفع
    }]
})

# أوروبا (مجموعة)
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_International_Europe",
        "Prefixes": [
            "33",  # فرنسا
            "49",  # ألمانيا
            "39",  # إيطاليا
```

```
            "34",  # إسبا��يا
            "31",  # هولندا
            "32",  # بلجيكا
            "41"   # سويسرا
        ]
    }]
})

# الدولية - الكل (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"]  # "الكل باستثناء 1" (NANP)
    }]
})
```

حالات الاستخدام:

- "**خطط محلية**: مكالمات غير محدودة إلى أرقام الولايات المتحدة/كندا" → `Dest_Domestic_All`
- "**إضافة مكالمات إلى المملكة المتحدة**: 100 دقيقة إلى المملكة المتحدة" → `Dest_International_UK`
- "**خطة أسترالية**: 3000 دقيقة إلى الهواتف المحمولة والخطوط الثابتة الأسترالية" → `Dest_AU_Mobile`، `Dest_AU_Fixed`، `Dest_AU_TollFree`
- "**حزمة دولية**: 50 دقيقة إلى أي رقم دولي" → `Dest_International_All`

الجزء 2: وجهات PLMN (من الاستخدام / التجوال - استخدام البيانات الأماكن)

رموز الشبكة لاستخدام البيانات وسي��اريوهات 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"` → شبكة Verizon 004 فقط
- **جميع الشبكات في البلد**: `"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
    }]
})
# لمشغل الشبكة الفعل🔴 الخاص بك PLMN برمز استبدل mcc505.mnc057
# يجب أن تستخدم جميع أرصدة البيانات المحلية على الشبكة البادئات المحددة هنا
# وإلا 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) في الولايات المتحدة
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 الرموز MCC (310، 311، إلخ.)
# العميل على mcc310.mnc004 (Verizon)، mcc310.mnc410 (AT&T)،
mcc311.mnc580 (US Cellular)
# سيتطابق جميعهم مع هذه الوجهة
```

وجهات PLMN للتجوال في المملكة المتحدة

```
#
=========================================================================
# وجهات PLMN للتجوال في المملكة المتحدة
#
=========================================================================

# 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" أو "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 المشغلين

# 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 جدًا)
    }]
})
```

حالات الاستخدام:

- **التجوال المحدد بالشبكة**: "5 جيجابايت على Verizon فقط" → `Dest_PLMN_US_Verizon`
- **التجوال على مستوى البلد**: "10 جيجابايت في الولايات المتحدة (أي شبكة)" → `Dest_PLMN_US_All`
- **التجوال الإقليمي**: "15 جيجابايت للتجوال في أوروبا" → `Dest_PLMN_Zone_Europe`
- **التجوال العالمي**: "1 جيجابايت في أي مكان" → `Dest_Roaming_All`

متى تستخدم كل نهج

نوع الوجهة	حالة الاستخدام	مثال	أنواع الأرصدة
جغرافية	الصوت/SMS إلى دول محددة	"100 دقيقة إلى أرقام المملكة المتحدة"	صوت، SMS
PLMN على الشبكة	بيانات محلية على شبكتك	"100 جيجابايت من البيانات على الشبكة"	بيانات
PLMN محدد	تجوال متميز على الشبكات المفضلة	"5 جيجابايت على Verizon فقط"	بيانات (وصوت/SMS أثناء التجوال)
PLMN على مستوى البلد	التجوال في أي شبكة في البلد	"5 جيجابايت في الولايات المتحدة (أي شبكة)"	بيانات (وصوت/SMS أثناء التجوال)
PLMN المنطقة	حزم التجوال الإقليمية	"10 جيجابايت للتجوال في أوروبا"	بيانات (وصوت/SMS أثناء التجوال)

تحديثات الوجهات الديناميكية والوصول إلى الخدمة

كيفية عمل تحديثات الوجهات

مفهوم حرج: يتم **الإشارة إلى الوجهات في CGRateS بواسطة ID**، وليس نسخها. عندما تقوم PLMN بتحديث بادئات الوجهة أو رموز **وتحميل خطة التعريفة في وقت 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 تعديل - **تدخل التغييرات حيز التنفيذ عند تحميل خطة التعريفة** 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` تشير إلى **العملاء الذين لديهم أرصدة** 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 **حظر** العملاء الذين يحاولون التجوال في ألمانيا، حتى لو كان لديهم رصيد متبقي لـ `Dest_PLMN_Zone_Europe`. الرصيد موجود لكنه 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",  # (وكندا سابقًا) تشمل بالفعل الولايات المتحدة
            # لا حاجة لتغيير - بادئة "1" تغطي كلاهما
        ]
    }]
})

# أو إنشاء وجهة شمال أمريكا منفصلة
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_North_America",
        "Prefixes": [
            "1"  # كندا + الولايات المتحدة
        ]
    }]
})
```

تعمل الآن لكل من `DestinationIDs: "Dest_Domestic_All"` **الأثر:** 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) **ولم تتطابق أي وجهة**، سيتم **رفض الخدمة**، حتى لو كان لديهم أرصدة.

السيناريو 1: عدم وجود وجهة مطابقة

```
# العميل لديه رصيد:
{
    "BalanceId": "Europe_Data_5GB",
    "BalanceType": "*data",
    "DestinationIDs": "Dest_PLMN_Zone_Europe",  # فقط فرنسا، ألمانيا، إيطاليا
    "Units": 5 * 1024 * 1024 * 1024
}

# العميل يتجول على الشبكة في إسبانيا (mcc214.mnc001)
# إسبانيا ليست في Dest_PLMN_Zone_Europe
```

النتيجة:

- ❌ جلسة البيانات **محظورة** (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 مع `DestinationIDs: "*any"` كخيار احتياطي.

السيناريو 2: عدم وجود خيار احتياطي عام

```
# العميل لديه 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"

# العميل يتصل بأستراليا (+61)
```

النتيجة:

- ❌ المكالمة **محظورة** (لا توجد وجهة تطابق بادئة "61")
- يبقى 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 العام

```
# العميل لديه:
# رصيد 1: دقائق المملكة المتحدة
{
    "BalanceId": "UK_Voice_100min",
    "BalanceType": "*voice",
    "DestinationIDs": "Dest_International_UK",
    "Units": 100 * 60 * 1000000000,
    "BalanceWeight": 1200
}

# رصيد 2: نقدي مع خيار عام
{
    "BalanceId": "PAYG_Credit",
    "BalanceType": "*monetary",
    "DestinationIDs": "*any",  # يتطابق مع أي شيء
    "Units": 5000,  # $50
    "BalanceWeight": 1000  # أولوية أقل
}

# العميل يتصل بأستراليا (+61)
```

النتيجة:

- ✅ **مسموح بها** المكالمة
- رصيد دقائق المملكة المتحدة غير مستخدم (لا يتطابق مع "61")
- يتم استخدام رصيد PAYG Credit (يتطابق مع "*any")
- يتم تحصيل الرسوم بمعدل 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. استخدم `LoadTariffPlanFromStorDb` لجعل **تحميل خطة التعريفة لتفعيل** تغييرات الوجهة نشطة في وقت التشغيل
3. **عدم تطابق الوجهة = عدم وجود خدمة** - سيتم حظر العملاء إذا لم يتطابق استخدامهم مع أي وجهة
4. **قوي الاحتياطي العام (`*any`)** - استخدمه كخيار احتياطي نقدي لمنع الحظر 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 الوجهات على جميع العملاء بمجرد تحميلها

ملفات التعريف السعرية للأرصدة النقدية/الدفع حسب الاستخدام

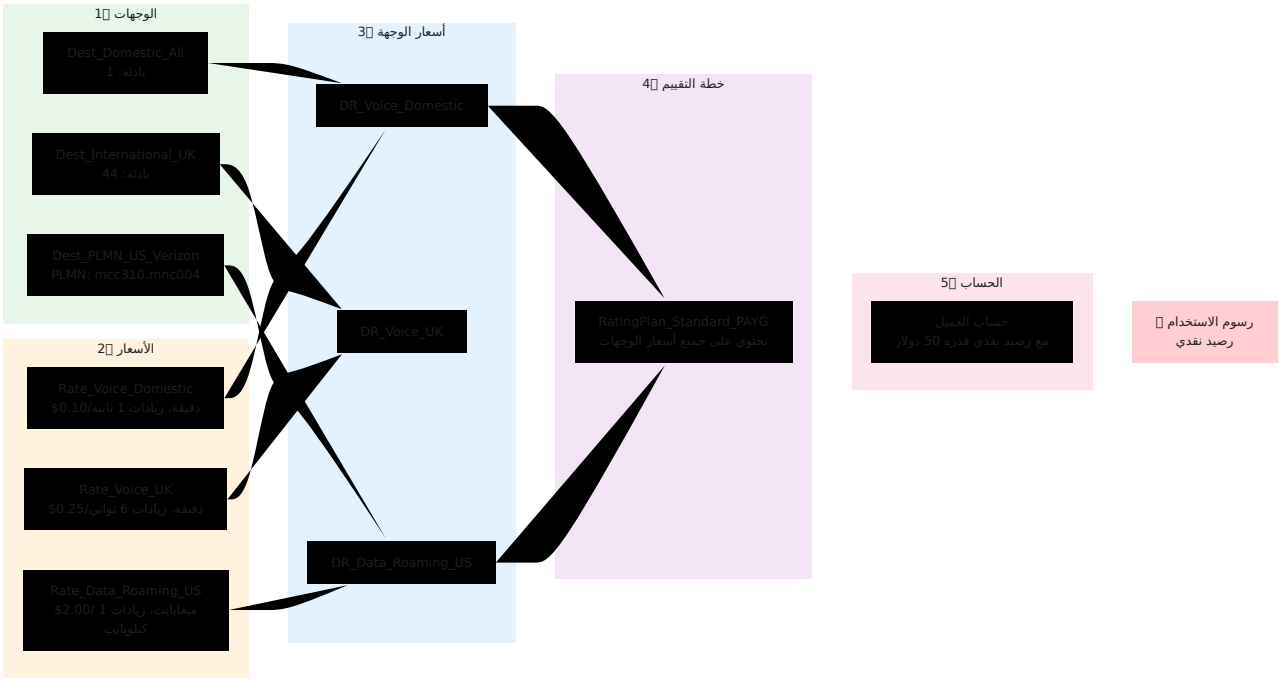
تحتاج إلى تحديد **ملفات التعريف السعرية** التي تحدد مقدار الرسوم لكل وحدة ل��واقع مختلفة عند استخدام الأرصدة النقدية (PAYG - الدفع حسب الاستخدام)،.

ملاحظة مهمة حول مرونة CGRateS: CGRateS مرن للغاية ويمكنه تطبيق أسعار وخطط مختلفة بناءً على العديد من العوامل بما في ذلك:

- من أين يتصل العميل (PLMN - تم تغطيته هنا)
- ما الذي يتصلون به (وجهة جغرافية - تم تغطيته هنا)
- وقت اليوم (تقييم ذروة/غير ذروة)
- مستوى العميل (VIP، إلخ، قياسي.)
- مستوى جودة الخدمة
- والعديد من سيناريوهات التقييم المعقدة الأخرى

والتقييم القائم على الوجهات PLMN لتجوال **CRM** يركز هذا الدليل على **تكوين مستوى** الجغرافية، والتي تغطي أكثر حالات الاستخدام شيوعًا. السيناريوهات المتقدمة للتقييم (وقت اليوم، ولكنها خارج نطاق هذه CGRateS مستويات العملاء، التوجيه القائم على الجودة) ممكنة في الوثائق التي تركز على CRM.

كيفية عمل ملفات التعريف السعرية



تعريف ملفات التعريف السعرية

تحدد ملفات ��لتعريف السعرية تكلفة الوحدة لأنواع الاستخدام المختلفة. إليك أمثلة للأسعار الخاصة بالصوت، SMS، والبيانات:

```
#
# ============================================================================
# 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",      # زيادات 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 كيلوبايت
            "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** - متى تبدأ هذه الفئة السعرية

ربط الوجهات بالأسعار

بعد تعريف الأسعار، اربطها بالوجهات باستخدام أسعار الوجهة:

```
# 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 البريطانية بسعر الصوت البريطاني
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 البيانات للتجوال
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 التقييم

اجمع جميع أسعار الوجهات في خطة تقييم:

```
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPRatingPlan',
    'params': [{
        "TPid": tpid,
        "ID": "RatingPlan_Standard_PAYG",
        "RatingPlanBindings": [
            # أسعار الصوت (وزن أعلى = 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 للتجوال
            {"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:

- تعريف المنتجات - سير العمل الكامل لإنشاء المنتجات
- إجراءات CGRateS وسلوكيات الشحن - كيفية تعريف الإجراءات واستراتيجيات 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 (جميع 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 بشكل منطقي

```
# 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 PLMN، عند تعريف وجهات MCC/MNC:

```
# 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: العميل لديه رصيد لكنه لا يمكنه استخدامه 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 بالنسبة لـ PLMN:
# تحقق من رسائل CCR Diameter لرسالة AVP Visited-PLMN-Id
```

المشكلة: استخدام البيانات محسوب كالتجوال

الأعراض: استخدام البيانات المحلية يستهلك رصيد التجوال بدلاً من رصي�ي الشبكة المحلية

السبب: لم يتم تعريف PLMN الخاص بالعميل 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 البادئات تشمل الشبكة الفعلية للعميل
# مثال: ["mcc505.mnc057"] يجب أن تتطابق مع شبكة SIM الخاصة بالعميل
```

المشكلة: المكالمات الدولية لا تعمل

الأعراض: العميل لديه "دقائق دولية" لكن المكالمات تفشل

تحقق:

1. **بادئة الوجهة**: هل يتطابق الرقم المتصل مع بادئة الوجهة؟
   - "يحتاج إلى وجهة مع بادئة "44 → الاتصال بـ +44-20-xxxx
2. **في الرصيد**: هل يتضمن الرصيد تلك الوجهة؟ **حقل DestinationIDs**
   - تحقق من أن الرصيد يحتوي على `"DestinationIDs": "Dest_International_UK"` أو `"Dest_International_All"`
3. **الرصيد لم ينتهِ**: تحقق من `ExpiryTime`
4. **الرصيد ي�ديه وحدات**: تحقق من `Units > 0`

الوثائق ذات الصلة

- وسلوكيات الشحن - كيفية استخدام الوجهات في الإجراءات، CGRateS إجراءات استراتيجيات الأرصدة، واستراتيجيات التقييم
- تعريف المنتجات - سير العمل الكامل لإنشاء المنتجات

أدلة تفصيلية لبرامج Ansible

تتم تهيئة منتجات OmniCRM باستخدام **Ansible**، مما يسمح بإدارة الخدمات بشكل آلي بناءً على المتطلبات المحددة لكل منتج ومخزونه المرتبط.

للحصول على `SIM Card Provisioning <concepts_sim_provisioning>` انظر أيضًا: مثال كامل عن التهيئة المعتمدة على Ansible للخدمات المحمولة، بما في ذلك بطاقات SIM الفعلية و eSIMs.

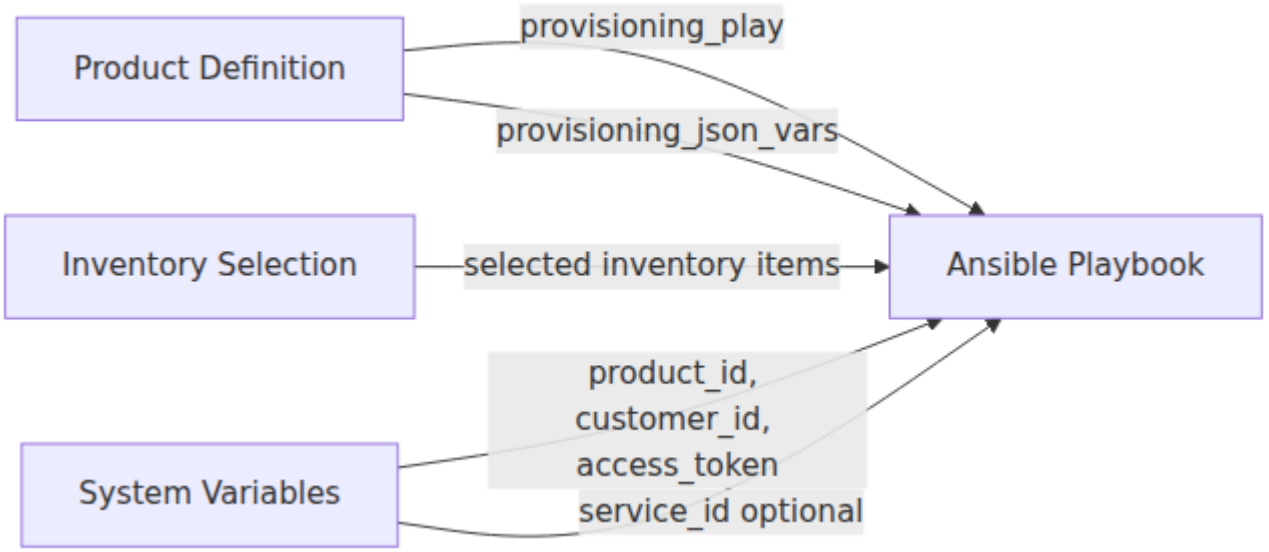
كيف تعمل البرامج مع المنتجات

مفهوم حاسم: البرامج هي ما ينشئ الخدمات فعليًا في OmniCRM. عندما تعين برنامجًا لمنتج، فإنك تحدد **ما يحدث** عندما يتم تهيئة ذلك المنتج - ولكن يمكن أن يعني ذلك أشياء مختلفة لمنتجات مختلفة.

المنتجات تحفز البرامج

عندما يتم تهيئة منتج في OmniCRM:

1. يحدد تعريف المنتج البرنامج الذي يجب تشغيله (عبر حقل `provisioning_play`)
2. يمرر المنتج المتغيرات إلى البرنامج (عبر `provisioning_json_vars` واختيارات المخزون)
3. ينفذ البرنامج ويفعل ما تم برمجته للقيام به
4. يحدد البرنامج ما الذي سيتم إنشاؤه (إن وجد)



ما يمكن أن تفعله البرامج

يمكن لبرنامج تهيئة واحد أن:

ينشئ خدمات متعددة
قد ينشئ برنامج منتج مجمع:

- سجل خدمة إنترنت رئيسي
- سجل خدمة إضافية IPTV
- سجل خدمة VoIP
- كل ذلك من خلال إجراء تهيئة منتج واحد

لا ينشئ أي خدمات
بعض البرامج لا تنشئ سجلات خدمات على الإطلاق:

- برنامج يقوم فقط بتكوين معدات CPE
- برنامج يرسل التكوين إلى معدات الشبكة
- برنامج يقوم بتحديث الأنظمة الخارجية

ينشئ خدمة واحدة
النمط الأكثر شيوعًا:

- إنشاء سجل خدمة واحد للعميل
- ربط المخزون بتلك الخدمة
- إعداد الفوترة لتلك الخدمة

تعديل الخدمات الموجودة
برامج التعبئة والإضافات:

- لا تنشئ خدمات جديدة
- تحديث سجلات الخدمة الموجودة (إضافة بيانات، تمديد انتهاء الصلاحية، إلخ)
- إضافة أرصدة إلى حسابات الفوترة الموجودة

تنفيذ إجراءات بدون سجلات خدمات
بعض البرامج هي عمليات تشغيلية بحتة:

- إعادة تعيين أرصدة الحسابات
- تبديل عناصر المخزون بين العملاء
- توليد تقارير أو تكوينات

مثال: سلوكيات برامج مختلفة

```
# المنتج SIM المحمولة (تنشئ خدمة واحدة) :1 خدمة
# provisioning_play: play_simple_service
- Creates service record in CRM
- Creates billing account in OCS
- Assigns SIM card and phone number inventory
- Sends welcome email

# المنتج 2: حزمة الإنترنت (تنشئ 3 خدمات)
# provisioning_play: play_bundle_internet_tv_voice
- Creates internet service record
- Creates IPTV service record
- Creates VoIP service record
- Links all to same customer
- Single billing account for the bundle

# المنتج 3: تعبئة البيانات (تنشئ 0 خدمات)
# provisioning_play: play_topup_no_charge
- Finds existing service by service_id
- Adds data balance to existing OCS account
- Updates service expiry date
- NO new service created

# المنتج CPE (تنشئ 0 خدمات) :4 تكوين
# provisioning_play: play_prov_cpe_mikrotik
- Generates router configuration
- Updates inventory record with config
- Emails config to support team
- NO service created (just equipment setup)
```

النقطة الرئيسية: **البرنامج يحدد السلوك، والمنتج هو مجرد محفز.**

الألعاب مقابل المهام

فهم التمييز بين الألعاب والمهام هو أمر أساسي للعمل مع برامج OmniCRM.

لعبة (برنامج)
سير عمل تهيئة كامل ينظم مهام متعددة لتحقيق هدف تجاري. الألعاب هي البرامج ذات المستوى

ويتم الإشارة إليها في تعريفات `OmniCRM-API/Provisioners/plays/` الأعلى المخزنة في المنتجات.

أمثلة:

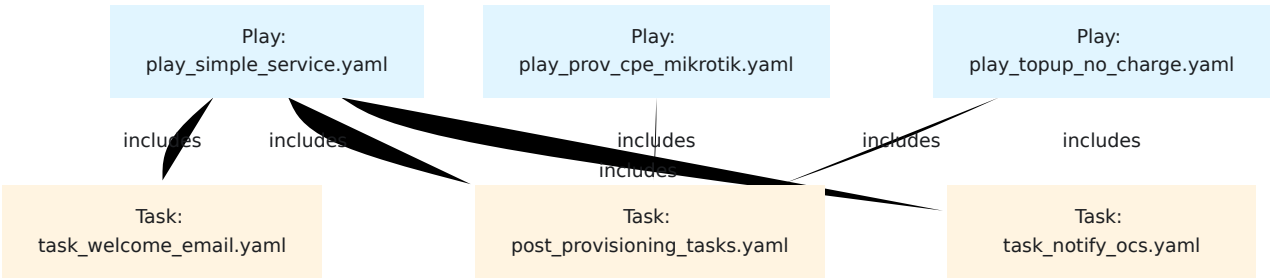
- `play_simple_service.yaml` - تهيئة خدمة أساسية
- `play_topup_no_charge.yaml` - تطبيق تعبئة مجانية على خدمة
- `play_prov_cpe_mikrotik.yaml` - تكوين معدات العملاء

مهمة (مكون قابل لإعادة الاستخدام)
مجموعة من العمليات ذاتية المحتوى، القابلة لإعادة الاستخدام والتي يمكن تضمينها بواسطة ألعاب متعددة. المهام مسبوقة بـ `task_` وتوجد 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 بينهما:



```yaml
# play_simple_service.yaml (لعبة)
- name: Simple Provisioning Play
  hosts: localhost
  tasks:
    - name: Main provisioning block
      block:
        - name: Create service
          uri: ...

        - name: Configure billing
          uri: ...

        # تضمين مهمة قابلة لإعادة الاستخدام
        - include_tasks: task_welcome_email.yaml

        # تضمين مهام ما بعد التهيئة
        - include_tasks: post_provisioning_tasks.yaml
```

هيكل البرنامج وتشريحه

تتبع جميع برامج OmniCRM هيكلًا متسقًا. فهم هذا الهيكل ضروري لإنشاء وصيانة البرامج.

الهيكل الأساسي

يبدأ كل برنامج بهذه الرؤوس القياسية:

```yaml
- name: Descriptive Name of the Playbook
  hosts: localhost           # OmniCRM لـ localhost دائمًا
  gather_facts: no           # معطل لأداء أفضل
  become: False              # لا تصعيد الامتيازات

  tasks:
    - name: Main block
      block:
        # مهام التهيئة تذهب هنا

      rescue:
        # مهام التراجع/التنظيف تذهب هنا
```

شرح الرأس

name
اسم وصفي يظهر في سجلات التهيئة وواجهة المستخدم. يظهر هذا كـ `playbook_description` في سجل التهيئة.

hosts: localhost
تعمل جميع برامج OmniCRM على localhost لأنها تتفاعل مع الأنظمة البعيدة عبر واجهات برمجة التطبيقات، وليس SSH.

gather_facts: no
تم تعطيل جمع الحقائق في Ansible لأن:

- لا نحتاج إلى معلومات النظام
- يضيف عبئًا 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 برنامج تحميل ملف التكوين المركزي:

```
tasks:
  - name: Include vars of 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:

```
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:

```
- name: Access product_id passed by OmniCRM
  debug:
    msg: "Provisioning product {{ product_id }}"
```

من اختيار المخزون:

```
- name: Get inventory ID for SIM Card
  set_fact:
    sim_card_id: "{{ hostvars[inventory_hostname]['SIM Card'] |
int }}"
  when: "'SIM Card' in hostvars[inventory_hostname]"
```

من استجابات API:

```
- 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: Use the product name
  debug:
    msg: "Product name is {{
api_response_product.json.product_name }}"
```

أنماط البرامج الشائعة

نمط تهيئة الخدمة

هذا هو النمط الأكثر شيوعًا لإنشاء خدمات جديدة.

```yaml
- name: Service Provisioning Playbook
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Main block
      block:

      # 1. تحميل التكوين
      - name: Include vars of crm_config
        ansible.builtin.include_vars:
          file: "../../crm_config.yaml"
          name: crm_config

      # 2. الحصول على معلومات المنتج
      - 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

      # 3. الحصول على معلومات العميل
      - name: Get Customer information from CRM API
        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 المستردة
      - name: Set package facts
        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: Generate UUID
        set_fact:
          uuid: "{{ 99999999 | random | to_uuid }}"

      - name: Generate Service UUID
        set_fact:
          service_uuid: "Service_{{ uuid[0:8] }}"

      # 6. إنشاء حساب في نظام الفوترة
      - name: Create account in 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: Verify OCS account creation
        assert:
```

```
      that:
        - ocs_response.status == 200
        - ocs_response.json.result == "OK"

    # 7. إضافة رصيد أولي
    - name: Add 0 Monetary 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": "*monetary",
                "Categories": "*any",
                "Balance": {
                    "ID": "Initial Balance",
                    "Value": 0,
                    "ExpiryTime": "+4320h",
                    "Weight": 1,
                    "Blocker": true
                }
            }]
          }
        status_code: 200
      register: balance_response

    # 8. إنشاء سجل خدمة في CRM
    - name: Get current date and time in ISO 8601 format
      command: date --utc +%Y-%m-%dT%H:%M:%S%z
      register: current_date_time

    - name: Add Service via 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: 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,
              "retail_cost": "{{ setup_cost | float }}"
            }
          return_content: yes
        register: transaction_response

      # 10. تضمين مهام ما بعد التهيئة
      - include_tasks: post_provisioning_tasks.yaml
```

```
    rescue:

    # قسم التراجع/التنظيف
    - name: Print all vars for debugging
      debug:
        var: hostvars[inventory_hostname]

    - name: Remove account in 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: Delete Service from CRM if it was created
      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: Fail if not intentional deprovision
      assert:
        that:
          - action == "deprovision"
```

نمط التعبئة/إعادة الشحن

يستخدم لإضافة أرصدة أو بيانات أو وقت إلى الخدمات الموجودة.

```yaml
- name: Service Topup Playbook
  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

    # 1. الحصول على معلومات 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 }}"
        return_content: yes
      register: api_response_service

    # 2. الحصول على معلومات المنتج (ما يجب تعبئته)
    - 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

    # 3. استخراج تفاصيل الخدمة
    - 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
}}"
        topup_value: "{{ api_response_product.json.retail_cost }}"
```

```
    # 4. تنفيذ الإجراء في نظام الفوترة (تعبئة مجانية)
    - name: Execute Action to add credits
      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: Verify action executed successfully
      assert:
        that:
          - action_response.status == 200
          - action_response.json.result == "OK"

    # 5. إعادة تعيين أي حدود تم تفعيلها
    - 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
            }]
          }
        status_code: 200

    # 6. تحديث تواريخ الخدمة
    - name: Calculate new expiry date
      command: "date --utc +%Y-%m-%dT%H:%M:%S%z -d '+30 days'"
```

```
      register: new_expiry_date

    - name: Update Service with new expiry
      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: Send Notification SMS
      uri:
        url: "http://sms-gateway/api/send"
        method: POST
        body_format: json
        body:
          {
            "source": "CompanyName",
            "destination": "{{ customer_phone }}",
            "message": "Your service has been topped up. New
expiry: {{ new_expiry_date.stdout }}"
          }
        status_code: 201
      ignore_errors: True
```

نمط تهيئة CPE

يستخدم لتكوين معدات العملاء (المودمات، الموجهات، ONTs).

```yaml
- name: CPE Provisioning Playbook
  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

    # 1. الحصول على عنصر المخزون لـ CPE
    - name: Set CPE inventory ID from 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: Get Inventory data for 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: Get Site info from 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 CPE inventory item with location
      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: Set CPE hostname
      set_fact:
        cpe_hostname: "CPE_{{ cpe_inventory_id }}"
        cpe_username: "admin_{{ cpe_inventory_id }}"

    - name: Generate random password
      set_fact:
        cpe_password: "{{ lookup('pipe', 'cat /dev/urandom | tr -
dc a-zA-Z0-9 | head -c 16') }}"

    # 6. توليد بيانات اعتما WiFi
    - name: Set WiFi SSID
      set_fact:
        wifi_ssid: "Network_{{ cpe_inventory_id }}"

    - name: Generate WiFi password
      set_fact:
        word_list:
          - apple
          - cloud
          - river
          - mountain
          - ocean

    - name: Create WiFi PSK
```

```
      set_fact:
        random_word: "{{ word_list | random }}"
        random_number: "{{ 99999 | random(start=10000) }}"

    - name: Combine WiFi PSK
      set_fact:
        wifi_psk: "{{ random_word }}{{ random_number }}"

    # 7. توليد ملف التكوين
    - name: Set config filename
      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: Create config from template
      template:
        src: "templates/cpe_router_config.j2"
        dest: "{{ config_dest }}"

    # 8. قراءة التكوين الناتج
    - name: Read config file
      ansible.builtin.slurp:
        src: "{{ config_dest }}"
      register: config_content

    # 9. تحديث المخزون بمعلومات التهيئة
    - name: Patch CPE inventory with config
      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": "Provisioned: {{ lookup('pipe',
'date +%Y-%m-%d') }}"
          }
        status_code: 200

    # 10. إرسال التكوين إلى فريق الدعم
    - name: Email configuration to support
      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": "Provisioning System"
              },
              "To": [{
                "Email": "support@example.com",
                "Name": "Support Team"
              }],
              "Subject": "CPE Config - {{ 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 الرسوم المتكررة التلقائية أو التجديدات باستخدام خطط العمل من CGRateS.

```
# جزء من برنامج تعبئة يقوم بإعداد التجديد التلقائي

# 1. تطبيع معلمة auto_renew
- name: Normalize auto_renew to boolean
  set_fact:
    auto_renew_bool: "{{ (auto_renew | string | lower) in ['true',
'1', 'yes'] }}"

# 2. إنشاء إجراء للتجديد التلقائي
- name: Create Action for AutoRenew
  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: Create ActionPlan for Monthly Renewal
```

```
  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: Assign ActionPlan to account
  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: Remove ActionPlan from account
  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 الاستخد��م

المهام القابلة لإعادة الاستخدام هي برامج صغيرة ذات محتوى ذاتي يمكن تضمينها بواسطة ألعاب متعددة. إنها تعزز إعادة استخدام الكود والاتساق.

مهمة البريد الإلكتروني الترحيبي

`task_welcome_email.yaml` - ترسل بريدًا إلكترونيًا ترحيبيًا للعملاء الجدد.

```
# تتوقع هذه المهمة أن يتم تعيين هذه المتغيرات بواسطة اللعبة الأم:
# - api_response_customer (تفاصيل العميل)
# - package_name (اسم المنتج)
# - monthly_cost (التكلفة المتكررة)
# - setup_cost (التكلفة لمرة واحدة)

- name: Set email configuration
  set_fact:
    mailjet_api_key: "{{ lookup('env', 'MAILJET_API_KEY') }}"
    mailjet_api_secret: "{{ lookup('env', 'MAILJET_SECRET') }}"
    email_from: "noreply@example.com"
    recipients: []

- name: Set email subject and sender name
  set_fact:
    email_subject: "Welcome to our service!"
    email_from_name: "Customer Service Team"

- name: Prepare list of recipients from customer contacts
  loop: "{{ api_response_customer.json.contacts }}"
  set_fact:
    recipients: "{{ recipients + [{'Email': item.contact_email,
'Name': item.contact_firstname ~ ' ' ~ item.contact_lastname}] }}"

- name: Get first contact name
  set_fact:
    first_contact: "{{
api_response_customer.json.contacts[0].contact_firstname }}"

- name: Send welcome email
  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": "Dear {{ first_contact }}, welcome! Your
service is ready.",
            "HTMLPart": "Dear {{ first_contact }},<br/><h3>Welcome!
</h3><br/>Your {{ package_name }} service is now active.
<br/>Monthly cost: ${{ monthly_cost }}<br/>Setup fee: ${{
setup_cost }}<br/>If you have any questions, contact
support@example.com"
          }]
        }
      user: "{{ mailjet_api_key }}"
      password: "{{ mailjet_api_secret }}"
      force_basic_auth: true
      status_code: 200
    register: email_response
```

مهام ما بعد التهيئة

`post_provisioning_tasks.yaml` - تنظيف قياسي وإشعارات يتم تشغيلها بعد كل تهيئة.

```
# يتم تضمين هذا الملف في نهاية معظم برامج التهيئة
# يتعامل مع العمليات الشائعة بعد التهيئة

- include_tasks: task_notify_ocs.yaml
```

قد تحتوي `task_notify_ocs.yaml` على:

```
- name: Notify OCS of provisioning completion
  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

استرداد تفاصيل المخزون:

```
- name: Get SIM Card inventory ID
  set_fact:
    sim_inventory_id: "{{ hostvars[inventory_hostname]['SIM Card']
| int }}"
  when: "'SIM Card' in hostvars[inventory_hostname]"

- name: Get SIM Card details
  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: Extract SIM details
  set_fact:
    iccid: "{{ sim_response.json.iccid }}"
    imsi: "{{ sim_response.json.imsi }}"
    ki: "{{ sim_response.json.ki }}"
```

تعيين المخزون للعميل:

```
- name: Assign SIM to customer
  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: Get current date and time in ISO 8601 format
  command: date --utc +%Y-%m-%dT%H:%M:%S%z
  register: current_date_time

- name: Get today's date only
  set_fact:
    today: "{{ lookup('pipe', 'date +%Y-%m-%d') }}"
```

حساب التواريخ المستقبلية:

```
- name: Calculate expiry date 30 days from now
  command: "date --utc +%Y-%m-%dT%H:%M:%S%z -d '+30 days'"
  register: expiry_date

- name: Calculate date 90 days in future
  command: "date --utc +%Y-%m-%d -d '+{{ days }} days'"
  register: future_date
  vars:
    days: 90
```

توليد قيم عشوائية

UUIDs والمعرفات:

```
- name: Generate UUID
  set_fact:
    uuid: "{{ 99999999 | random | to_uuid }}"

- name: Generate service identifier
  set_fact:
    service_uuid: "SVC_{{ uuid[0:8] }}"
```

كلمات مرور عشوائية:

```
- name: Generate secure password
  set_fact:
    password: "{{ lookup('pipe', 'cat /dev/urandom | tr -dc a-zA-
Z0-9 | head -c 16') }}"
```

عبارات مرور يمكن تذكرها:

```
- name: Set word list
  set_fact:
    words:
      - alpha
      - bravo
      - charlie
      - delta
      - echo

- name: Generate passphrase
  set_fact:
    word: "{{ words | random }}"
    number: "{{ 99999 | random(start=10000) }}"

- name: Combine into passphrase
  set_fact:
    passphrase: "{{ word }}{{ number }}"
```

العمل مع CGRateS/OCS

إنشاء الحسابات:

```
- name: Create billing account
  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: Add 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",
          "Categories": "*any",
          "Balance": {
            "ID": "Data Package",
            "Value": 10737418240,
            "ExpiryTime": "+720h",
            "Weight": 10
          }
        }]
      }
    status_code: 200
```

تنفيذ الإجراءات:

```
- name: Execute charging action
  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: Get account details
  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: Get AttributeProfile
  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: Extract attribute value
  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: Use custom value or default
  set_fact:
    monthly_cost: "{{ custom_cost | default(50.00) }}"

- name: Only run if variable is defined
  debug:
    msg: "Service UUID is {{ service_uuid }}"
  when: service_uuid is defined
```

الشروط المنطقية:

```
- name: Provision equipment
  include_tasks: configure_cpe.yaml
  when: provision_cpe | default(false) | bool

- name: Skip if deprovision
  assert:
    that:
      - action != "deprovision"
  when: action is defined
```

شروط متعددة:

```
- name: Complex conditional task
  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: Create multiple balances
  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 عبر استجابات API:

```
- name: Get all customer sites
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/site/customer_id/{{
customer_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: sites_response

- name: Configure equipment at each site
  debug:
    msg: "Configuring site at {{ item.address_line_1 }}"
  loop: "{{ sites_response.json }}"
```

معالجة الأخطاء

استخدام ignore_errors:

```yaml
- name: Optional SMS notification
  uri:
    url: "http://sms-gateway/send"
    method: POST
    body: {...}
  ignore_errors: true
```

التحقق من صحة الاستجابة:

```yaml
- name: Verify API response
  assert:
    that:
      - response.status == 200
      - response.json.result == "OK"
    fail_msg: "API call failed: {{ response.json }}"
```

معالجة الأخطاء الشرطية:

```yaml
- name: Try to get existing service
  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: Create service if it doesn't exist
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/service/"
    method: PUT
    body: {...}
  when: service_lookup.status == 404
```

أفضل الممارس��ت

تسمية المتغيرات

استخدم أسماء وصفية ومتسقة:

```
# جيد
service_uuid: "SVC_12345"
customer_name: "John Smith"
monthly_cost: 49.99

# سيء
svc: "SVC_12345"
name: "John Smith"
cost: 49.99
```

قم ببادئة المتغيرات حسب المصدر:

```
api_response_customer: {...}
api_response_product: {...}
cgr_account_info: {...}
```

تصحيح الأخطاء

طباعة المتغيرات لأغراض استكشاف الأخطاء:

```
- name: Print all variables
  debug:
    var: hostvars[inventory_hostname]

- name: Print specific variable
  debug:
    msg: "Service UUID: {{ service_uuid }}"

- name: Print API response
  debug:
    var: api_response_product.json
```

التحقق من الصحة

تحقق دائمًا من استجابات API الحرجة:

```
- name: Create account
  uri:
    url: "{{ billing_endpoint }}"
    method: POST
    body: {...}
  register: response

- name: Verify account creation
  assert:
    that:
      - response.status == 200
      - response.json.result == "OK"
    fail_msg: "Failed to create account: {{ response.json }}"
```

التكرار

صمم المهام لتكون قابلة للتشغيل بأمان:

```
# تحقق مما إ كان المورد موجودًا أولاً
- name: Check if account exists
  uri:
    url: "{{ ocs_endpoint }}/get_account"
    method: POST
    body: {"Account": "{{ service_uuid }}"}
  register: account_check
  failed_when: false

# أنشئ فقط إذا لم يكن موجودًا
- name: Create account
  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 متغيرات البيئة
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: Call external API
  uri:
    url: "https://api.example.com/endpoint"
    method: POST
    headers:
      Authorization: "Bearer {{ access_token }}"
    validate_certs: yes
```

الوثائق

وثق المنطق المعقد:

```
# حساب الرسوم النسبية للشهر الجزئي
# إذا قام العميل بالتسجيل في اليوم الخامس عشر وكانت الفوترة في اليوم الأول،
# فقم بفرض 50% من التكلفة الشهرية للأيام المتبقية
- name: Calculate days until end of month
  command: "date -d 'last day of this month' +%d"
  register: days_in_month

- name: Get current day
  command: "date +%d"
  register: current_day

- name: Calculate pro-rata amount
  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 البرامج

نهج الاختبار

1. **جرب أولاً**: اختبر مع الأنظمة غير الإنتاجية
2. **تحقق من المتغيرات**: استخدم مهام التصحيح لتأكيد 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 التهيئة**: اختبر باستخدام `action: "deprovision"` للتحقق من التنظيف

مثال على برنامج اختبار:

```yaml
- name: Test Service Provisioning
  hosts: localhost
  gather_facts: no

  tasks:
    - name: Verify required variables
      assert:
        that:
          - product_id is defined
          - customer_id is defined
          - access_token is defined
        fail_msg: "Missing required variables"

    - name: Test API connectivity
      uri:
        url: "http://localhost:5000/crm/health"
        method: GET
      register: health_check

    - name: Verify health check
      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 افتراضيًا
service_uuid: "{{ service_uuid | default('') }}"
```

نسيان التحقق من الصحة:

```
# خطأ - لا يتحقق من الاستجابة
- name: Create account
  uri: ...
  register: response

# صحيح - تحقق من الاستجابة
- name: Create account
  uri: ...
  register: response

- name: Verify creation
  assert:
    that:
      - response.json.result == "OK"
```

سير عمل التهيئة

بشكل عام، سيعمل موظفو Omnitouch مع العميل على:

1. تحديد متطلبات المنتج
2. تطوير برامج Ansible اللازمة لأتمتة عملية التهيئة
3. اختبار البرامج في بيئة staging
4. نشرها في الإنتاج

هذا يضمن أن كل خدمة يتم نشرها بشكل متسق وموثوق، مما يقلل من خطر الأخطاء ويضمن إكمال جميع الخطوات اللازمة بالترتيب الصحيح.

متغيرات Ansible

تشمل المتغيرات المرسلة إلى برامج Ansible:

متغيرات المنتج
مشتقة من تكوينات منتج OmniCRM وتحدد كيفية إعداد الخدمة.

متغيرات المخزون
محددة من المخزون، تشمل عناصر مثل المودمات، بطاقات SIM، كتل عناوين IP، أو أرقام الهواتف المطلوبة للتهيئة.

متغيرات النظام
تضاف تلقائيًا بواسطة OmniCRM:

- `product_id` - المنتج الذي يتم تهيئته
- `customer_id` - العميل الذي يتلقى الخدمة
- `service_id` - الخدمة التي يتم تعديلها (للتعبئة/التغييرات)
- `access_token` - JWT لمصادقة API

إلغاء التهيئة

لإلغاء تهيئة الخدمة باستخدام **Ansible** عندما لم تعد الخدمة مطلوبة، يتم أيضًا استخدام **برامج** نمط كتلة `rescue`. هذا:

- يزيل أي تكوينات
- يحرر المخزون مرة أخرى إلى المجموعة
- يحذف حسابات الفوترة
- يضمن الحفاظ على نظافة النظام

طرق إلغاء التهيئة

هناك طريقتان رئيسيتان يتم تحفيز إلغاء التهيئة:

1. التراجع التلقائي (فشل التهيئة)
عندما تفشل أي مهمة في كتلة التهيئة الرئيسية، يتم تنفيذ قسم الإنقاذ تلقائيًا لتنظيف التغييرات الجزئية.

2. إلغاء التهيئة اليدوي
تعيين `action: "deprovision"` عند تشغيل برنامج يثير كتلة الإنقاذ عمدًا لإزالة خدمة.

نمط إلغاء التهيئة

هذا الهيكل للتنظيف الآمن OmniCRM تتبع جميع برامج:

```yaml
- name: Service Provisioning Playbook
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Main block
      block:
        # جميع مهام التهيئة تذهب هنا
        - name: Create OCS account
          uri: ...

        - name: Create service record
          uri: ...
          register: service_creation_response

        - name: Add transaction
          uri: ...

      rescue:
        # مهام إلغاء التهيئة/التراجع تذهب هنا
        - name: Print all vars for debugging
          debug:
            var: hostvars[inventory_hostname]

        # تنفيذ مهام التنظيف بترتيب عكسي
        - name: Remove account in OCS
          uri: ...

        - name: Delete Service from CRM
          uri: ...

        - name: Fail if not intentional deprovision
          assert:
            that:
              - action == "deprovision"
```

مثال كامل لإلغاء التهيئة

إليك مثال كامل من `play_simple_service.yaml`:

```
rescue:
  # 1. معلومات التصحيح
  - name: Print all vars for Deprovision/Rollback
    debug:
      var: hostvars[inventory_hostname]

  - name: Get today's date
    set_fact:
      today: "{{ lookup('pipe', 'date +%Y-%m-%d') }}"

  # 2. الحصول على معلومات الخدمة إذا كانت متاحة
  - name: Try to get Service information from CRM API to
Deprovision
    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: Print api_response_service
    debug:
      var: api_response_service
    when: api_response_service is defined

  # 3. تعيين service_uuid للتنظيف
  - name: Set service_uuid facts for Deprovision
    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: Remove account in 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: Print response from OCS account removal
    debug:
      var: response
    when: response is defined

  # 5. حذف سجل الخدمة من CRM
  - name: Delete Service from CRM if it was created
    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: Print delete service response
    debug:
      var: delete_service_response
    when: delete_service_response is defined
```

```
  # 6. تحديد ما إذا كان هذا إلغاء تهيئة مقصودًا أو فشلًا
  - name: Set status to "Success" if Manual deprovision / Fail if
failed provision
    assert:
      that:
        - action == "deprovision"
      fail_msg: "Provisioning failed and was rolled back"
      success_msg: "Service deprovisioned successfully"
```

مفاهيم رئيسية لإلغاء التهيئة

ignore_errors: True

تستخدم معظم مهام التنظيف `ignore_errors: True` (على سبيل المثال، إذا فشلت التهيئة قبل إنشائها) لأن الموارد قد لا توجد.

التنفيذ الشرطي

استخدم عبارات `when` لتنظيف الموارد فقط التي تم إنشاؤها:

```
when: service_creation_response is defined
```

ترتيب عكسي

يحدث التنظيف بترتيب عكسي من الإنشاء:

1. حذف الموارد التابعة أولاً (المعاملات، تعيينات المخزون)
2. حذف سجل الخدمة
3. حذف حساب OCS/الفوترة
4. تحرير أي موارد محجوزة

التحقق النهائي

التحقق النهائي يميز بين:

- **إلغاء تهيئة مقصود** (`action == "deprovision"`) → نجاح
- **فشل التهيئة** (لم يتم تعيين الإجراء) → فشل

إلغاء تهيئة أنواع الموارد المختلفة

حسابات OCS/CGRateS:

```
- name: Remove account in 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 خدمات CRM:

```
- name: Delete Service from 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: Release SIM card back to inventory pool
  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: Remove ActionPlan from account
  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: Release payment authorization
  uri:
    url: "http://localhost:5000/crm/payments/release/{{
authorization_id }}"
    method: POST
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "metadata": {
          "release_reason": "provisioning_failed"
        }
      }
    return_content: yes
  ignore_errors: True
  when: authorization_id is defined
```

سير عمل إلغاء التهيئة اليدوي

لإلغاء تهيئة خدمة يدويًا:

1. تحديد الخدمة التي سيتم إلغاء تهيئتها
2. تشغيل برنامج التهيئة الأصلي مع `action: "deprovision"`
3. تدخل البرنامج كتلة الإنقاذ على الفور
4. يتم تنفيذ جميع مهام التنظيف
5. تتم إزالة الخدمة بشكل نظيف

مثال على مكالمة 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، فشلت إنشاء الخدمة

- حساب OCS موجود
- سجل الخدمة لا يوجد
- يقوم قسم الإنقاذ بإزالة حساب OCS
- لا توجد خدمة للحذف (تم تخطيها بأمان مع شرط `when`)

السيناريو 2: تم إنشاء الخدمة، فشلت المعاملة

- حساب OCS موجود
- سجل الخدمة موجود
- المعاملة لا توجد
- يقوم قسم الإنقاذ بإزالة كل من حساب OCS والخدمة
- لا توجد معاملة لإبطالها (لم يتم إنشاؤها أبدًا)

السيناريو 3: التهيئة الكاملة، فشل التقاط الدفع

- حساب OCS موجود
- سجل الخدمة موجود
- تم تفويض الدفع ولكن لم يتم التقاطه
- يقوم قسم الإنقاذ:
  - بإصدار تفويض الدفع (لم يتم تحصيل رسوم من العميل)
  - إزالة سجل الخدمة
  - إزالة حساب OCS

أفضل الممارسات لإلغاء التهيئة

1. استخدم ignore_errors بحرية
يجب أن يكون التنظيف متسامحًا - لا تفشل إذا لم يكن المورد موجودًا.

2. تحقق من وجود المورد باستخدام عبارات when
حاول التنظيف فقط إذا تم إنشاء المورد:

```
when: service_creation_response is defined
```

3. طباعة معلومات التصحيح
تأكد دائمًا من تضمين مهام التصحيح للمساعدة في استكشاف الأخطاء في التهيئات الفاشلة:

```
- name: Print all vars for debugging
  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 بإصدار تفويضات الدفع في كتل الإنقاذ لتجنب فرض رسوم على العملاء بسبب التهيئات الفاشلة.

6. إعادة تحميل المجدولين بعد التنظيف
عند إزالة الموارد من OCS، قم بتضمين `ReloadScheduler: true` لضمان تحديث CGRateS على الفور.

إلغاء التهيئة مقابل الحذف

إلغاء التهيئة (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 التهيئة وإلغاء التهيئة للتعامل مع الأخطاء في Ansible تستخدم ميزة **block/rescue** بشكل سلس. إذا فشلت أي مهمة في أي نقطة أثناء التهيئة، يتم تنفيذ قسم الإنقاذ تلقائيًا للتراجع عن التغييرات للعودة إلى حالة متسقة. يضمن ذلك الموثوقية ويقلل من خطر النشر الجزئي أو الفاشل.

مثال على معالجة الأخطاء

```
- name: Main block
  block:
    - name: Step 1: Create account
      uri: ...
      register: response

    - name: Verify step 1
      assert:
        that:
          - response.status == 200

    - name: Step 2: Create service
      uri: ...

  rescue:
    # إذا فشلت أي تأكيد أو حدث خطأ في أي مهمة:
    # 1. ينتقل التنفيذ إلى كتلة الإنقاذ
    # 2. يتم تنفيذ مهام التنظيف
    # 3. يفشل البرنامج مع رسالة خطأ
    - name: Cleanup partial changes
      uri: ...
```

للحصول على تفاصيل كاملة حول نظام التهيئة، وسير العمل، والمصادقة، راجع `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 مرتبط بأدوار وأذونات محددة و سلسلة عشوائية آمنة (حد أدنى 32 حرفًا) تقوم بمصادقة الطلبات عند تمريرها في رأس `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 محددة 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

CRM ملاحظات خدمة / منتج

ملاحظة

للحصول على دليل شامل من البداية إلى النهاية يغطي تعريف المنتج، التوفير، الإضافات، وإلغاء التوفير مع أمثلة مفصلة باستخدام Ansible واستراتيجية التسعير، راجع `دليل دورة حياة المنتج الكامل <guide_product_lifecycle>`.

للحصول على معلومات مفصلة حول توفير خدمات الهاتف المحمول باستخدام بطاقات SIM، راجع `توفير بطاقة SIM <concepts_sim_provisioning>`.

نظرة عامة على المنتجات والخدمات

المنتج (عنصر القائمة):

المنتج يشبه طبقًا محددًا 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 مشمول — الميزات، المواصفات، في OmniCRM، والتسعير.

غالبًا ما يرغب العملاء في تعديلات، مثل "استبعاد البصل" أو "إضافة جبن إضافي" إلى وجبتهم. يتوافق هذا مع تخصيص خدمة قبل إنشائها. مستوى التخصيصات أو التعديلات داخل OmniCRM، على الخدمة متروك لك (المشغل) لتعريفه.

قد يقوم العملاء أو الموظفون بتعديل منتج ليتناسب بشكل أفضل مع احتياجات في OmniCRM، 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 يشبه القائمة الكاملة في مطعم، التي 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 أفضل.

الخدمة (الطبق المعد):

عندما يطلب عميل عنصرًا من القائمة، يتم إعداد الطبق في المطبخ. هذا يشبه إنشاء خدمة من منتج.

في OmniCRM، عندما يختار عميل منتجًا، يتم إنشاء نسخة من ذلك المنتج ويتم تسليمه 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 — (مودمات ومنافذ 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 واحدة مريحة — مثل "حزمة المستلزمات المنزلية" التي تشمل خدمات الإنترنت، الكابل، والهاتف بسعر مخفض.

بمجرد تحديدها، يتم تحويل هذه الحزمة إلى خدمات متعددة مصممة خصيصًا للعميل.

تعريفات المنتجات

المنتج هو قالب يُستخدم لإنشاء خدمة / إضافة / خصم / إضافة إضافية، إلخ.

داخل التعريف ندرج:

- معلومات حول المنتج (الميزات، المكونات، الشروط والأحكام، مدة العقد، الرمز، 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 تم توفيرها (م��ل الإضافات الهاتفية المتاحة فقط للعملاء الذين لديهم خدمة هاتف محمول)، إذا كان يمكن طلبه مباشرة من قبل عميل عبر الرعاية الذاتية أو فقط من قبل وكيل خدمة العملاء، ومتى يمكن شراء المنتج (السماح لمنتج أن يكون متاحًا 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 تعيينه:

  `عناصر المخزون` هذه تتوافق مع `['SIM Card', 'Phone Number']` المحددة في CRM `<administration_inventory>`.

- لتوفير الخدمة `لعب التوفير` الإشارة إلى كتاب لعب Ansible بالإضافة إلى المتغيرات التي يجب تمريرها إلى `<concepts_ansible>` Ansible. هذه المتغيرات التي يجب تمريرها هي سحرية، حيث قد تكون متغيرات مثل التي يتم تعريفها بواسطة المنتج الذي نضيفه إليه، أو قد تكون مثل `service_id` حيث ا��ترنا عناصر المخزون التي يتم تمريرها عند تعيين `MSISDN` و `ICCID` المخزون. يتم التعامل مع التجميع في لعبة التوفير لتحتوي على خدمات متعددة، على سبيل المثال قد توفر منتج إنترنت منزلي مجمع، تلفاز وصوت خدمة لكل منها.

فئات المنتجات وأنواع الخدمات

تستخدم المنتجات حقلين تصنيفيين للمساعدة في تنظيم وتصنيف العروض:

فئات المنتجات

حقل `category` يتحكم 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 كإضافة للخدمات الموجودة)
- **عرض ترويجي** - عروض ترويجية خاصة

هذه الفئات تنظيمية بحتة ولا تحدد ما يتم توفيره. سلوك التوفير الفعلي يتم تحديده بالكامل من خلال كتاب اللعب Ansible المش`provisioning_play`ار إليه في.

على سبيل المثال: - منتج `مستقل` عادةً ما ينشئ كائن خدمة جديدة - منتج `إضافة` أو `حزمة` عادةً ما يتم إضافته إلى خدمة موجودة - لكن هذا متروك للمطور الذي يكتب كتاب اللعب - يمكنك إنشاء عدة كائنات خدمة من إضافة، أو تعديل الخدمات الموجودة من منتج مستقل إذا لزم الأمر

الفئة تتحكم ببساطة في تدفق واجهة المستخدم وأين يرى العملاء/الموظفون خيار المنتج.

أنواع الخدمات

حقل `service_type` يصنف 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 اتصال** - أجهزة نقطة اتصال محمولة أو إيجار
- **دونجل** - خدمات المودم USB أو الدونجل
- **صوت** - خدمات الصوت فقط
- **بيانات** - خدمات البيانات فقط

مثل الفئات، أنواع الخدمات قابلة للتخصيص بناءً على عروضك. إنها تساعد في:

- تصفية الإضافات التي تنطبق على أي خدمات أساسية
- تنظيم المنتجات في بوابة العملاء
- مطابقة متطلبات المخزون
- تحديد سير العمل للتوفير

مثال: يمكن لعميل لديه خدمة `محمول` رؤية الإضافات المحمولة، بينما يمكن لعميل لديه خدمة `ثابت` رؤية الإضافات الثابتة.

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 تبويب محسنة تنظم جميع إعدادات المنتج في مجموعات منطقية لتسهيل التنقل والتحرير.

تتميز نافذة إدارة المنتج بخمس علامات منظمة:

1. **المعلومات الأساسية** - معلومات المنتج الأساسية (الاسم، الشعار، الفئة، الرمز، 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 لعب Ansible ومتطلبات المخزون
5. **التوافر** - نطاقات التاريخ والطوابع الزمنية للنظام

تنظيم علامة التسعير:

تجمع علامة التسعير حقول التكلفة في أقسام منطقية:

- **التكاليف المتكررة** - تكاليف التجزئة والجملة الشهرية جنبًا إلى جنب
- **تكاليف الإعداد** - رسوم التفعيل لمرة واحدة للتجزئة والجملة
- **الضريبة** - تكوين نسبة الضريبة مع حساب تلقائي

ميزات وضع التحرير:

- **اختيار الرمز** - البحث واختيار رموز FontAwesome بصريًا
- **اختيار عناصر المخزون** - اختيار من أنواع عناصر المخزون المتاحة
- **اختيار التاريخ/الوقت** - اختيار سهل لنوافذ التوافر
- **تنسيق العملة** - بادئة $ تلقائية لحقول التكلفة
- **اختيارات منسدلة** - خيارات محددة مسبقًا للفئات والحقول المنطقية

اختيار الرمز:

عند تحرير حقل الرمز، تظهر واجهة اختيار رمز قابلة للبحث تتيح لك تصفح واختيار بصري من آلاف رموز FontAwesome.

ميزات: * البحث عن الرموز حسب الكلمة الرئيسية (مثل "مفتاح"، "محمول"، "واي فاي") * عرض معاينة للرمز في الوقت الفعلي * عرض اسم فئة الرمز للرجوع إليه * اختيار منسدل للوصول السريع

علامة التكوين:

تنظم علامة 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 الاستهلاكيين
  - تجاري - متاح للعملاء التجاريين
- **الاعتمادات:**
  - يعتمد على القائمة - معرفات المنتجات أو أنواع الخدمات المطلوبة قبل إضافة هذا المنتج
  - يستخدم لاعتمادات الإضافات (مثل، تتطلب الإضافات الهاتفية خدمة هاتف محمول نشطة)

علامة التوفير:

تتعامل علامة التوفير مع أتمتة Ansible ومتطلبات المخزون.

حقول التوفير:

- **لعب التوفير:**
  - اسم كتاب لعب Ansible (بدون امتداد .yaml)
  - يجب أن يكون موجودًا في دليل OmniCRM-API/Provisioners/plays/
  - يتم استدعاؤه عند إنشاء الخدمة أو تحديثها أو إلغاء توفيرها
- **متغيرات JSON للتوفير:**
  - المتغيرات الافتراضية المرسلة إلى كتاب لعب Ansible كـ JSON
  - يمكن تجاوزها أثناء التوفير
  - يتلقى كتاب اللعب هذه بالإضافة إلى customer_id، product_id، service_id، access_token
- **قائمة عناصر المخزون:**
  - اختيار متعدد يظهر أنواع عناصر المخزون المتاحة
  - أمثلة: بطاقة SIM، عنوان IPv4، رقم الهاتف، مودم راوتر
  - يختار العميل/الموظف عناصر معينة من المخزون المتاح أثناء الطلب
  - يتم تمرير معرفات المخزون المحددة إلى كتاب اللعب مع نوع المخزون كاسم المتغير

علامة التوافر:

تتحكم علامة التوافر في متى يمكن شراء المنتج وتعرض بيانات التعريف النظامية.

إعدادات التوافر:

- **متاح من:**
  - تاريخ/وقت متى يصبح المنتج متاحًا للشراء
  - اتركه فارغًا للتوافر الفوري
  - مفيد للإعلان المسبق عن المنتجات الجديدة
- **متاح حتى:**
  - تاريخ/وقت متى لم يعد المنتج متاحًا للشراء
  - اتركه فارغًا للتوافر غير المحدود
  - مثالي للعروض المحدودة 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
  - استنساخ المنتج - إنشاء نسخة مع لاحقة "_clone"
  - تحرير المنتج - الانتقال إلى وضع التحرير
- **وضع التحرير/الإنشاء:**
  - إلغاء - discard changes and close
  - حفظ التغييرات - إنشاء أو تحديث المنتج (زر كبير للتأكيد)

حقول المنتج

نموذج المنتج يحتوي 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 أساسي عند إنشاء خدمة جديدة
  - `إضافة` - يظهر عند الإضافة إلى خدمة موجودة
  - `حزمة` - يظهر كخيار خدمة مجمعة
  - `عرض ترويجي` - عروض ترويجية خاصة
- **service_type** - نوع الخدمة المقدمة (مثل، محمول، ثابت، صوت ثابت، نقطة اتصال، دونجل، صوت، بيانات). يستخدم لتصفية الإضافات التي تنطبق على أي خدمات.
- **comment** - ملاحظات داخلية حول المنتج للرجوع إليها من قبل الموظفين فقط (لا تظهر للعملاء)
- **icon** - فئة رمز 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 (تستخدم لحسابات الهامش)
- **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 للـ 10%، 12.5 للـ 12.5%). تعيين إلى 0 للمنتجات المعفاة من الضرائب. 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 10% ضريبة، $50.00 تكلفة تجزئة → المعاملة بها $5.00 ضريبة
- منتج مع 0% ضريبة (معفى من الضريبة) → المعاملة بها $0.00 ضريبة
- تجاوز المعاملة اليدوية → يمكن للموظفين تغيير نسبة الضريبة لكل معاملة

رؤية العميل والوصول

- **enabled** - ما إذا كان هذا المنتج نشطًا ومتاحًا للشراء (تعيين إلى false لإخفائه دون حذفه)
- **residential** - ما إذا كان يمكن للعملاء السكنيين (المستهلكين) شراء هذا المنتج
- **business** - ما إذا كان يمكن للعملاء 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** - تاريخ/وقت متى يصبح هذا ��لمنتج متاحًا للشراء (اختياري)
- **available_until** - تاريخ/وقت متى لم يعد هذا المنتج متاحًا للشراء (اختياري، مفيد للعروض المحدودة)

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` - يجدد تلقائيًا دون سؤال
  - `false` - يتطلب تجديد يدوي
- **allow_auto_renew** - تعيين إلى) ما إذا كان يمكن للعملاء تمكين التجديد التلقائي false للشراء لمرة واحدة)

المحتوى الموجه للعميل

- **terms** - الشروط والأحكام المعروضة للعملاء قبل الشراء (تتضمن القيود، قواعد انتهاء الصلاحية، شروط الاستخدام)
- **features_list** - قائمة الميزات والمكونات المعروضة للعملاء (تنسيق قائمة بايثون: `['Feature 1', 'Feature 2']`)

تكوين التوفير

- **provisioning_play** - اسم كتاب لعب Ansible الذي يوفر هذه الخدمة (بدون امتداد .yaml). يجب أن يكون موجودًا في `OmniCRM-API/Provisioners/plays/`.
- **provisioning_json_vars** - المتغيرات الافتراضية المرسلة إلى كتاب لعب Ansible كـ JSON. يمكن تجاوزها عند التوفير. يتلقى كتاب اللعب 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 اللعب مع نوع المخزون كاسم المتغير.
- **relies_on_list** - قائمة بمعرفات المنتجات أو أنواع الخدمات التي يجب أن تكون موجودة قبل إضافة هذا المنتج. يستخدم لاعتمادات الإضافات (مثل، تتطلب الإضافات الهاتفية خدمة هاتف محمول نشطة).

بيانات التعريف النظامية

- **created** - الطابع الزمني عندما تم إنشاء المنتج (يتم تعيينه ت��قائيًا)
- **last_modified** - الطابع الزمني عندما تم تحديث المنتج آخر مرة (يتم تحديثه تلقائيًا)

تعريفات المنتجات النموذجية

منتج مستقل (محمولة SIM بطاقة)

```
{
  "product_id": 1,
  "product_slug": "Mobile-SIM",
  "product_name": "Mobile SIM Only",
  "category": "standalone",
  "service_type": "mobile",
  "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": "['Australian Phone Number (04xxx)', 'Fastest speeds', 'Best coverage', 'Roaming on the Mainland']",
  "terms": "Must be activated within 6 months. All credit lost if service is not used for 12 months.",
  "comment": "Physical SIM card for use with Mobile Phones"
}
```


وينشئ خدمة جديدة (ورقم الهاتف SIM بطاقة) هذا المنتج المستقل يتطلب عنصرين من المخزون عند توفيره.

منتج إضافة (خطة بيانات شهرية)


```
{
  "product_slug": "norfone-mobile-prepaid-mini",
  "product_name": "Norfone Mini Plan",
  "category": "addon",
  "service_type": "mobile",
  "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 of Ultra fast data', 'Unlimited Calls & Texts to Norfone users', '100 Minutes of Talk to Australia', '100 SMS to Australia', '30 Day Expiry']",
  "terms": "Credit expires after 30 days. Once data, voice or sms is used up, you will need to top up to continue using the service.",
  "comment": "Our smallest plan for light users"
}
```


هذا المنتج الإضافي لا يتطلب مخزون ويتم تطبيقه على خدمة موجودة. يقوم بشحن العميل ويضيف أرصدة/رصيد إلى خدمتهم.

منتج ��زمة (حزمة كبار السن)


```
{
  "product_slug": "Bundle-Seniors",
  "product_name": "Seniors Bundle",
  "category": "bundle",
  "service_type": "fixed",
  "provisioning_play": "play_seniors_package",
  "provisioning_json_vars": "{\"IPTV_Service_ID\":
\"SeniorBundle\"}",
  "inventory_items_list": "['Modem Router']",
  "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": "['20Mbps Download', '5Mbps Upload', 'Unlimited
Data', 'Home Voice', 'TV: Extra +£5 per month', '£60 Installation
Fee']",
  "terms": "6 Month Contract, must show senior citizen's card to
qualify",
  "comment": "20Mbps/2Mbps GPON Service + IPTV + Phone"
}
```


هذا المنتج المجمّع يوفر خدمات متعددة (إنترنت + IPTV + هاتف) باستخدام كتاب لعب واحد. يتطلب عنصرًا واحدًا من المخزون (مودم راوتر).

منتج إضافة (إعادة شحن بسيطة)

```
{
  "product_slug": "Mobile-Topup-5",
  "product_name": "PAYG £5 Topup",
  "category": "addon",
  "service_type": "mobile",
  "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": "['£5 credit', 'Valid for 180 days']",
  "terms": "Valid for 180 days or until all credit is used. See
our website for full rates",
  "comment": "£5 to use for Calls, SMS & Data"
}
```

هذا الإضافة ببساطة تضيف رصيد مالي إلى خدمة موجودة. لا يتطلب مخزون، ويستخدم `service_id` لتحديد الخدمة التي سيتم إعادة شحنها.

كيفية تمرير المتغيرات إلى Ansible

أمر حاسم لكتابة Ansible إلى كتاب لعب API فهم كيفية تدفق المتغيرات من تعريف المنتج عبر كتب لعب توفير فعالة.

مصادر المتغيرات والدمج

عند إنشاء وظيفة توفير، تأتي المتغيرات من مصادر متعددة وتدمج معًا بهذا الترتيب (المصادر اللاحقة تتجاوز السابقة):

1. **متغيرات provisioning_json_vars للمنتج** - المتغيرات الافتراضية من تعريف المنتج
2. **جسم الطلب** - المتغيرات المرسلة في استدعاء API (يمكن أن تتجاوز الافتراضات الخاصة بالمنتج)
3. **المتغيرات المضافة بواسطة النظام** - تُضاف تلقائيًا بواسطة نظام التوفير
4. **اختيارات المخزون** - معرفات عناصر المخزون المختارة (إذا لم يكن `inventory_items_list` فارغًا)

عملية دمج المتغيرات

يقوم النظام بدمج المتغيرات من جميع المصادر، مع تجاوز المصادر اللاحقة للمصادر السابقة. هذا 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 Ansible:

```
{
    "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 المضافة بواسطة النظام

يضيف نظام التوفير تلقائيًا:

- `access_token` - رمز JWT لمصادقة استدعاءات API مرة أخرى إلى CRM (مقدمة لمصادقة IP/API key، أو تم إنشاؤها من `refresh_token` لمصادقة 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 المخزون (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. **معرفات المخزون تُضاف إلى المتغيرات** - تُضاف معرفات عناصر المخزون المحددة مع نوع المخزون كاسم المتغير
3. **كتاب اللعب يصل إلى معرفات المخزون** - يمكن لكتاب لعب 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

تشمل المتغيرات المرسلة إلى كتاب اللعب: - `SIM Card`: 789 - `Mobile Number`: 101

يمكن لكتاب اللعب بعد ذلك استخدام هذه المعرفات لاسترداد سجلات المخزون الكاملة (ICCID، IMSI، MSISDN، إلخ.) من API CRM واستخدام تلك المعلومات لتوفير الخدمة على المعدات الشبكية.

كيفية استلام Ansible للمتغيرات

يمرر نظام التوفير جميع المتغيرات المدمجة إلى كتاب لعب Ansible كـ `extravars`. داخل كتاب اللعب، تكون هذه المتغيرات متاحة من خلال نظام المتغيرات القياسي لـ Ansible ويمكن استخدامها في المهام.

يمكن الإشارة إلى المت��يرات مباشرة في مهام كتاب اللعب باستخدام الصيغة `{{ variable_name }}`. على سبيل المثال، `{{ product_id }}`, `{{ customer_id }}`, `{{ monthly_cost }}`، إلخ.

المتغيرات المرسلة إلى منتجات الإضافة

عند توفير منتج إضافة، يمرر النظام تلقائيًا:

- `product_id` - معرف منتج الإضافة الذي يتم توفيره
- `customer_id` - العميل الذي يمتلك الخدمة
- `service_id` - معرف الخدمة التي تتم إضافة هذه 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 "إعادة شحن 5£" كإضافة إلى خدمة الهاتف المحمول الخاصة به (`service_id: 123`)، يتلقى كتاب اللعب المتغيرات بما في ذلك:

- `product_id`: 45 (منتج إعادة الشحن)
- `customer_id`: 456 (العميل)
- `service_id`: 123 (الخدمة لإضافة الرصيد إليها)
- `access_token`: رمز المصادقة
- `provisioning_json_vars` للمنتج بالإضافة إلى أي متغيرات من

يستخدم كتاب اللعب 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 وإضافات المتغيرات

تتلقى **المنتجات المستقلة**:

- `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 **منتجات الإضافة**:

- `product_id` - منتج الإضافة الذي يتم توفيره
- `customer_id` - العميل الذي يمتلك الخدمة
- `service_id` - هو الفرق (هذ**معرف الخدمة الموجودة التي يجب تعديلها** الرئيسي)
- `access_token` - لمصادقة API

هذا يخبر كتاب اللعب أي خدمة موجودة يجب تعديلها أو - `service_id` الفرق الرئيسي هو الإضافة إليها.

المنتجات المجمعة

تُوفر المنتجات المجمعة مثل الإضافات ولكن قد ينشئ كتاب اللعب الخاص بها 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، `Modem Router`) إذا كانت مطلوبة
- `access_token` - لمصادقة API

بإنشاء خدمات متعددة مرتبطة (مثل، `play_seniors_package`) ثم يقوم كتاب اللعب المجمّع ويربطها معًا (هاتف، IPTV، إنترنت).

الخدمات

الخدمة هي نسخة من منتج تنتمي إلى عميل، يتم الفوترة لها.

الذي يتعامل مع توليد الرسوم (نظام الفوترة عبر الإنترنت) `OCS` إنها في الأساس رابط إلى حساب ويدير CGRateS بواسطة OCS والأرصدة والاستخدامات الفعلي�? للحساب. يتم تشغيل الأرصدة النقدية، الأرصدة الوحدوية (البيانات، الصوت، الرسائل القصيرة)، خطط 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)، تعرض 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:
   - المنتجات حيث `residential = true` أو `business = false` **فردية (سكنية)**:
   - المنتجات حيث `business = true` **تجارية**:
2. **الفئة** - يتم فصل المنتجات إلى:
   - المنتجات مع `category` = `standalone` أو `bundle` **خطط الخدمة**:
   - المنتجات مع `category` = `addon` (تظهر في دوارة منفصلة) **الإضافات**:
3. **التوافر** - يتم عرض المنتجات فقط إذا:
   - `enabled = true` - المنتج نشط وليس معطلاً
   - `available_until` و `available_from` - التاريخ الحالي بين المنتج ضمن نافذة توافره
   - `customer_can_purchase = true` - (إذا كان العميل يشتري بنفسه) المنتج يسمح بالشراء المباشر من العميل

ملاحظة

تصفية مستوى API: يقوم API تلقائيًا بتصفية المنتجات حسب حالة التمكين وتواريخ التوافر على مستويين:

- **نقاط نهاية الشراء/الاختيار** (`/crm/product/`) - تستخدم بواسطة نموذج الإضافات وقائمة الخطط لاختيار المنتجات. تقوم بتصفية تلقائيًا لعرض المنتجات المفعلة فقط ضمن نطاق تواريخ توافرها. هذا يضمن أن العملاء والموظفين يمكنهم فقط اختيار المنتجات المتاحة حاليًا للشراء.
- **نقاط نهاية الإدارة** (`/crm/product/paginated`) - تستخدم بواسطة صفحة إدارة المنتجات. تعرض جميع المنتجات بما في ذلك المعطلة وخارج تواريخ التوافر، مما يسمح للمسؤولين بإدارة الكتالوج الكامل للمنتجات بما في ذلك المنتجات غير النشطة.

مرر `include_disabled=true` (للاستخدام الإداري فقط) إلى نقطة نهاية المنتج الأساسية لتجاوز التصفية.
تعرض واجهة المستخدم دوارات منفصلة لـ:

- **خطط الخدمة الفردية** - المنتجات السكنية للعملاء الاستهلاكيين
- **خطط الخدمة التجارية** - المنتجات التجارية للعملاء التجاريين
- **الإضافات الفردية** - حزم الإضافات السكنية
- **الإضافات التجارية** - حزم الإضافات التجارية

تصفية المنتجات لخدمات الإضافة

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 لديها `service_type = "mobile"`، يتم عرض الإضافات فقط التي لديها `service_type = "mobile"`
   - ذا يضمن أن العملاء المحمولين يرون فقط الإضافات المحمولة، وعملاء الإنترنت يرون فقط الإضافات 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 فقط التي تم الوفاء باعتماداتها
3. **تصفية نفس نوع العميل** - لا تزال الإضافات مصفاة حسب `residential` مقابل `business` لمطابقة نوع العميل

سيناريو تصفية مثال

بالنسبة لعميل تجاري لديه خدمة هاتف محمول موجودة (`service_type = "mobile"`):

- **المنتجات المستقلة المعروضة**: جميع المنتجات المستقلة/المجمعة التجارية (`business = true`، `category != "addon"`)
- **المنتجات الإضافية المعروضة**: فقط الإضافات المحمولة التجارية (`business = true`، `category = "addon"`، `service_type = "mobile"`)
- **المنتجات المخفية**: المنتجات السكنية، الإضافات لأنواع خدمات أخرى (الإنترنت، الصوت، إلخ)، المنتجات المعطلة

حقول الخدمة

نموذج الخدمة يحتوي على حقول تتبع نسخة الخدمة المقدمة وعلاقتها بالعميل، المنتج، ونظام الفوترة.

معلومات الخدمة الأساسية

- **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 رمز FontAwesome (قابل للتحرير) للعرض في بوابة الرعاية الذاتية

حالة الخدمة والتواريخ

- **service_status** - الحالة الحالية: نشطة، غ��ر نشطة، معلقة، إلخ. (قابل للتحرير)
- **service_provisioned_date** - عندما تم توفير الخدمة لأول مرة (يتم 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 للعميل (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 (يتم 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 لعب Ansible (موروث المستخدم لتوفير هذه الخدمة من المنتج، للقراءة فقط)
- **provisioning_json_vars** - المتغيرات المرسلة إلى كتاب لعب التوفير (موروثة من المنتج، للقراءة فقط)
- **deprovisioning_play** - كتاب لعب Ansible الذي يتم تشغيله عند إلغاء توفير الخدمة (للقراءة فقط)
- **deprovisioning_json_vars** - المتغيرات لكتاب لعب إلغاء التوفير (للقراءة فقط)

علاقات الخدمة

- **bundled_parent** - إذا كانت هذه الخدمة جزءًا من حزمة، فإن service_id للخدمة الأم (للقراءة فقط)
- **site_id** - رابط إلى الموقع/المكان الفعلي حيث يتم تسليم الخدمة (قابل للتحرير)

الملاحظات والبيانات التعريفية

- **service_notes** - ملاحظات داخلية حول الخدمة للرجوع إليها من قبل الموظفين (قابل للتحرير)
- **created** - الطابع الزمني عندما تم إنشاء الخدمة (يتم 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 (يتم تعيينها تلقائيًا):

لا يمكن تعديل هذه الحقول مباشرة بعد الإنشاء:

- service_id، customer_id، product_id
- service_uuid (يتم إنشاؤه أثناء التوفير)
- service_provisioned_date
- provisioning_play، provisioning_json_vars
- deprovisioning_play، deprovisioning_json_vars
- bundled_parent
- invoiced (يديرها نظام الفوترة)
- created، last_modified (يديرها النظام تلقائيًا)

توفير المنتجات كخدمات

تقوم عملية التوفير بتحويل منتج (قالب) إلى خدمة (نسخة محددة للعميل) من خلال سلسلة من الخطوات المنسقة التي تشمل واجهة الويب، API، وكتب لعب Ansible.

تدفق التوفير على مستوى عالٍ

1. **إعداد ما قبل التوفير** - يتم إنشاء المنتج في API مع تكوين التوفير، وكتابة واختبار كتب لعب Ansible 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 التوافر (`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 معينة (أي رقم هاتف، SIM مثل، أي بطاقة)
6. **بدء التوفير** - عند النقر على زر "توفير"، يقوم بإنشاء وظيفة توفير API

تدفق التكامل المفصل بين Ansible وAPI

عند توفير خدمة، تحدث التسلسل التالي:

الخطوة 1: إنشاء وظيفة التوفير

يتلقى API طلب التوفير وينشئ وظيفة توفير مع:

- `provisioning_play` - اسم كتاب لعب Ansible (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

يمزج خدمة التوفير المتغيرات من مصادر متعددة بهذا الترتيب:

1. `provisioning_json_vars` للمنتج (الافتراضات من تعريف المنتج)

2. معلمات جسم الطلب (يمكن أن تتجاوز الافتراضات الخاصة بالمنتج)
3. المتغيرات المضافة بواسطة 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 - تُضاف 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 إضافته بواسطة النظام
    "initiating_user": 5        # تمت إضافته بواسطة ال��نظام
}
```

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 كتاب اللعب (مستخرج من YAML)
- `provisioning_status` - رمز الحالة (يتم تعيينه في البداية إلى 1 = قيد التشغيل، ثم يتم تحديثه إلى 0 = نجاح، 2 = فشل، أو قد يبقى 1 إذا كان لا يزال قيد التقدم)
- `product_id`، `customer_id`، `service_id` - مراجع السياق

الخطوة 4: تنفيذ كتاب اللعب في الخلفية

يولد API خيطًا في الخلفية يقوم بـ:

1. تحميل YAML من كتاب اللعب من `OmniCRM-API/Provisioners/plays/{playbook_name}.yaml`
2. استدعاء `ansible_runner.run()` مع:
   - `playbook` - المسار إلى ملف YAML المحمل
   - `extravars` - جميع المتغيرات المدمجة (تمريرها إلى Ansible)
   - `inventory` - تعيينه إلى `'localhost,'` (تنفيذ محلي)
   - `event_handler` - معالج مخصص لالتقاط أحداث تنفيذ المهام
3. مراقبة تنفيذ كتاب اللعب في الوقت الحقيقي

الخطوة 5: التقاط الأحداث والتسجيل (`ProvisioningEventHandler`)

عند تنفيذ كل مهمة Ansible، يتم التقاط الأحداث وتخزينها كسجلات `Provision_Event`:

- `event_name` - اسم المهمة من كتاب اللعب
- `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 كتاب اللعب `ignore_errors: true`)
- `provisioning_result_json` - نتائج المهمة مع بيانات حساسة محجوبة

يقوم معالج الأحداث تلقائيًا بإزالة كلمات المرور، المفاتيح، الأسرار، وغيرها من البيانات الحساسة من السجلات.

الخطوة 6: تنفيذ كتاب لعب Ansible (`Provisioners/plays/*.yaml`)

يعمل كتاب لعب Ansible محليًا وعادةً ما يقوم بهذه الإجراءات:

1. طلب GET إلى `/crm/product/product_id/{{ product_id }}` باستخدام `{{ access_token }}` - **استرداد تعريف المنتج**
2. طلب GET إلى `/crm/customer/customer_id/{{ customer_id }}` - **استرداد معلومات العميل**
3. طلب GET إلى `/crm/inventory/inventory_id/{{ inventory_id }}` (إذا لزم الأمر) - **معالجة عناصر المخزون** 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 الفوترة، خطط الأسعار
    - خوادم ENUM لتخطيط أرقام الهواتف
    - المعدات الشبكية (الموجهات، المفاتيح، إلخ.)
5. **إضافة تكاليف الإعداد** - POST إلى `/crm/transaction/` لتسجيل (إذا لزم الأمر) الرسوم لمرة واحدة
6. **شحن العميل** - POST إلى OCS/CGRateS لشحن 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": "Mobile SIM - 0412345678",
  "service_uuid": "generated-uuid-for-ocs",
  "service_status": "Active",
  "service_type": "mobile",
  "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 }}` لوضع المخزون كـ "مخصص" للخدمة

11. **إرسال الإشعارات** (اختياري) - بريد إلكتروني أو رسالة نصية قصيرة للعميل مع تفاصيل الخدمة

الخطوة 7: الاكتمال وتحديث الحالة

عند اكتمال كتاب اللعب:

- **نجاح**: يتم تحديث `Provision.provisioning_status` (نجاح) إلى **0**
- **فشل حرج**: يتم تحديث `Provision.provisioning_status` إلى **2** (فشل)، ويتم إرسال بريد إلكتروني بالفشل إلى `crm_config.provisioning.failure_list`
- **الفشل غير الحرج**: يتم وضع المهام التي تفشل مع `ignore_errors: true` في الحالة **3** (فشل ولكن تم تجاهله) ولا توقف التوفير

الخدمة المقدمة الآن مرئية في CRM (إذا نجح التوفير) ونشطة للعميل.

الاختلافات الرئيسية: التوفير المستقل مقابل الإضافة مقابل التوفير المجمّع

المنتجات المستقلة (`category: standalone`):

- تتلقى `customer_id` و `product_id`
- عادةً ما تتطلب عناصر المخزون (أرقام الهواتف، المودمات، بطاقات SIM)
- تنشئ سجل خدمة **جديد** عبر API PUT `/crm/service/`
- توفر موارد جديدة على الأنظمة الخارجية (المعدات الشبكية، OCS، HSS)
- مثال: تفعيل SIM جديد المحمول الجديدة، اتصال الإنترنت

منتجات الإضافة (`category: addon`):

- تتلقى `customer_id`، `product_id` و **`service_id`** (الخدمة الموجودة للتعديل)
- 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 بالأم)
- مثال: إعادة شحن خطة البيانات الشهرية، حزمة التجوال الدولية، رصيد إضافي

المنتجات المجمعة (`category: bundle`):

- مشابهة للإضافات من حيث المتغيرات المستلمة

- قد تتطلب بعض عناصر المخزون (مثل المودم لحزمة المنزل)
- تنشئ **عدة** سجلات خدمة مرتبطة (إنترنت + تلفاز + هاتف)
- توفر موارد متعددة عبر أنظمة مختلفة
- تربط الخدمات معًا في CRM للفوترة/الإدارة الموحدة
- مثال: حزمة المنزل (إنترنت + IPTV + هاتف VoIP)

متطلبات كتاب لعب التوفير

لكي يعمل كتاب اللعب بشكل صحيح، يجب أن:

1. **يكون موجودًا في** `OmniCRM-API/Provisioners/plays/{playbook_name}.yaml`
2. **يقبل المتغيرات** عبر `extravars` لـ Ansible (يتم الوصول إليها كـ `{{ variable_name }}`)
3. **يصادق على استدعاءات API** باستخدام `Authorization: Bearer {{ access_token }}` في الرأس
4. **يتعامل مع الأخطاء بشكل جيد** حيثما `ignore_errors` و `rescue` كان ذلك مناسبًا
5. **ينشئ سجل الخدمة** للمنتجات المستقلة، أو يعدل الخدمة الموجودة للإضافات
6. **يخصص المخزون** إذا تم اختيار عناصر المخزون
7. **يرجع رسائل خطأ ذات مغزى** عبر وحدة `fail` عند حدوث أخطاء حرجة

المتغيرات الشائعة المتاحة في كتب اللعب

كل كتاب لعب يتلقى 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 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 ال��وفير

يمكن أن تستخدم كتب اللعب هذه لـ:

- 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 المعاملات: `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 لعب إضافة بسيطة

لإعادة شحن بيانات الهاتف المحمول:

1. يتلقى كتاب اللعب: `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. شحن العميل في OCS: `POST إلى CGRateS` لخصم retail_cost من الرصيد
5. إضافة رصيد البيانات في OCS: `POST إلى CGRateS` لإضافة رصيد البيانات مع انتهاء الصلاحية
6. تسجيل المعاملة في CRM: `POST /crm/transaction/` مع تفاصيل الشحن
7. الاكتمال بنجاح

`Provision` و `Provision_Event` لأغراض التصحيح يتم تتبع العملية بأكملها في جداول والتدقيق.

مشاركة OCS

`OCS` (نظام الفوترة عبر الإنترنت)، المنفذ عبر CGRateS، يتعامل مع جميع الرسوم في الوقت الحقيقي وتتبع الاستخدام للخدمات. سجل خدمة CRM يعمل كمرجع لحساب OCS، الذي يدير:

- **الرسوم المتكررة** - الرسوم الشهرية، إيجار DID، رسوم الاشتراك
- **الرسوم المعتمدة على الاستخدام** - مكالمات صوتية بالدقيقة، بيانات لكل ميغابايت، رسوم لكل رسالة قصيرة
- **إدارة الأرصدة** - الأرصدة النقدية (رصيد مدفوع مسبقًا) والأرصدة الوحدوية (بيانات GB، دقائق الصوت، عدد الرسائل القصيرة)
- **تحويل الأرصدة** - تحويل الأرصدة النقدية إلى أرصدة وحدوية (مثل، إنفاق $30 للحصول على حزمة بيانات 10 GB)
- **حالة الحساب** - نشط، معلق، معطل بناءً على حدود الائتمان و Thresholds

يحتوي سجل الخدمة في CRM على بيانات التعريف والتكوين (العميل، المنتج، التسعير، الرؤية)، بينما يحتوي OCS على الحالة الفوترة الحية (الأرصدة، الاستخدام، الرسوم).

استرداد استخدام الخدمة والأرصدة

يتم استرداد معلومات استخدام الخدمة من OCS/CGRateS وعرضها على العملاء والموظفين في الوقت الحقيقي.

كيفية استرداد الاستخدام

عند طلب استخدام الخدمة (عبر واجهة المستخدم أو API)، يحدث التدفق التالي:

1. **طلب API** - يستدعي الواجهة الأمامية `GET /crm/service/{service_id}` أو يعرض 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 استدعائين إلى CGRateS:

    i. **Get_Balance(service_uuid)** - يسترد رصيد الحساب مع `BalanceMap`

- يُرجع الأرصدة منظمة حسب النوع: بيانات، صوت، رسائل قصيرة، نقدية، بيانات دوانجل
- يتضمن كل رصيد: ID، الوزن، تاريخ انتهاء الصلاحية، القيمة، DestinationIDs
- يضيف النظام حقول قابلة للقراءة البشرية: `custom_Name_hr`, `custom_Expiration`, `custom_Description_String`

ii. **Get_ActionPlans(service_uuid)** - يسترد خطط العمل النشطة للتجديد التلقائي (المغطاة في القسم التالي)

4. يتم دمج بيانات CGRateS في استجابة الخدمة - **دمج الاستجابة**:

```
{
  "service_id": 123,
  "service_name": "Mobile Service",
  "service_uuid": "abc-123-def",
  "cgrates": {
    "BalanceMap": {
      "DATA": [{
        "ID": "DATA_10GB",
        "Value": 5368709120,
        "ExpirationDate": "2025-02-01T00:00:00Z",
        "custom_Name_hr": "10GB Data Pack",
        "custom_Expiration": "Feb 1, 2025",
        "custom_Description_String": "5 GB remaining"
      }],
      "VOICE": [{
        "ID": "VOICE_UNLIMITED",
        "Value": 999999999,
        "custom_Name_hr": "Unlimited Calls",
        "custom_Description_String": "Unlimited minutes"
      }],
      "MONETARY": [{
        "ID": "PREPAID_CREDIT",
        "Value": 25.50,
        "custom_Description_String": "$25.50 credit"
      }]
    },
    "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 لتكون قابلة للقراءة

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 (5 = 5368709120 GB)
    - للصوت: ثوانٍ (3600 = 1 ساعة)
    - للرسائل القصيرة: عدد (100 = 100 رسالة)
    - للنقدية: وحدات العملة (25.50 = 25.50$)
- `ExpirationDate` - طابع زمني ISO 8601 عندما تنتهي صلاحية الرصيد
- `Weight` - أولوية استهلاك الرصيد (يتم استهلاك الوزن الأعلى أولاً)
- `DestinationIDs` - الوجهات التي ينطبق عليها هذا الرصيد (مثل، ["AU", "INTERNATIONAL"])

الحقول القابلة للقراءة البشرية (مضافة 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 GB أو "10 GB المتبقية" الكلية"
    - الصوت: "60 دقيقة المتبقية" أو "غير محدود"
    - الرسائل القصيرة: "50 SMS المتبقية"
    - النقدية: "25.50$ رصيد"

التحكم في رؤية الاستخدام

يتم التحكم في رؤية استخدام الخدمة بواسطة حقلين:

- **service_visible_to_customer** - إذا كانت false، يتم إخفاء الخدمة تمامًا من بوابة الرعاية الذاتية للعميل
- **service_usage_visible_to_customer** - إذا كانت false، تكون الخدمة مرئية ولكن تفاصيل الاستخدام/الرصيد مخفية (يمكن للعميل رؤية أنه�� يمتلكون الخدمة، ولكن ليس مقدار ما استخدموه)

هذا يسمح للمشغلين بـ:

- إخفاء الخدمات الداخلية/الاختبارية عن العملاء
- عرض وجود الخدمة دون الكشف عن الاستخدام (مفيد للخطط غير المحدودة أو الخدمات الحساسة)
- عرض الاستخدام بشكل شفاف تمامًا (افتراضي)

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 fetches البيانات الحية من CGRateS
- **الكفاءة**: يتم تحديث فقط عرض الخدمات النشطة؛ تستخدم عرض القوائم البيانات المخزنة

هذا يضمن أن العملاء والموظفين يرون تحديثات الرصيد في الوقت القريب كما يحدث الاستخدام.

الرسوم المتكررة / التجديد التلقائي

`OCS` أولاً كإجراءات داخل DID تُنشأ الرسوم المتكررة، مثل رسوم الخدمة الشهرية، أو رسوم لكل `Action_ServiceUUID_ServiceName_WhatitDoes` وتأخذ الشكل.

لنفترض خدمة GPON 1TB بقيمة $60 شهريًا تتضمن 1 من الاستخدام، ستبدو الإجراء كالتالي:

`Action_kj49-adsf-1234-9742_60_GPON_1TB_MonthlyExpiry`

1. إعادة تعيين الرصيد النقدي إلى 0$
2. لتوفير شيء ما على CRM إرسال POST HTTP إلى `/simple_provision`
3. إضافة رصيد لاستخدام 1 TBينتهي في شهر واحد

إذا أردنا جعل MRC متكررًا (نعم نفعل) فسوف ننشئ `ActionPlan` باسم `ActionPlan_{{ service_uuid }}_Monthly_Charge` والذي سيكون له الوقت مضبوطًا على *شهري* ليتم تشغيله كل شهر، وتعيين `ActionPlan` للحساب.

يمكننا تعيين تاريخ انتهاء الصلاحية بناءً على معلمة *السنة / الأشهر / الأيام* للتاريخ الذي سيتوقف فيه MRC أيضًا، على سبيل المثال لخدمة ثابتة لمدة 12 شهرًا التي توقفت بعد هذه النقطة.

فريدة لكل خدمة، فإنها لا تشارك أي شيء مع أي خدمات `ActionPlans`نظرًا لأن الإجراءات و أخرى.

هذا يعني أنه يمكننا توفيرها بقيم معدلة، ولن يؤثر ذلك على أي خدمات أخرى.

الإضافات والإضافات

تُعالج الإضافات / الإضافات مثل شراء بيانات إضافية، حزم التجوال، دقائق دولية، إلخ، بنفس الطريقة تقريبًا. يتم إنشاء إجراء للقيام بما هو مطلوب، مثل فرض قيمة نقدية ثم منح رصيد وحدوي مع انتهاء محدد.

لإضافتها تلقائيًا إلى الحساب، نقوم فقط بتفعيل `ActionPlans` بدلاً من استخدام `ExecuteAction` للإجراء الذي أنشأناه للتو مرة واحدة من داخل Ansible.

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

تُستخدم `ThresholdS` على كل حساب لتعيين 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.

يمكن إزالة `ActionPlans` حسب الحاجة من API CRM، وتنفيذها عبر كتب لعب Ansible. يمكن إضافة `ActionPlans` عن طريق إضافة إضافة/خدمة وتنفيذها عبر كتب CRM، حسب الحاجة، من لعب Ansible.

يمكن تعطيل حسابات `OCS`، مما سيوقف تنفيذ `ActionPlans` ومنع الخدمات من الاستهلاك.

بالنسبة لحدود الائتمان، يتم تعيين قيمة `ThresholdS` وفقًا للسياسة للمنتج.

عرض وإدارة ActionPlans في CRM

تُعرض ActionPlans (تكوينات التجديد التلقائي) وتُدار 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
    - هذا يستدعي داخليًا `Get_ActionPlans(service_uuid)` على CGRateS
2. **هيكل بيانات ActionPlan** - يحتوي كل ActionPlan على تم إرجاعه:

```
{
    "ActionPlanId": "ServiceID_abc-123-
def__ProductID_456__...",
    "PlanName": "Monthly_Renewal_Plan",
    "NextExecTime": "2025-02-01T00:00:00Z",
    "custom_NextExecTime_hr": "in 11 days",
    "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:

   - **اسم المنتج** `ProductID` من ActionPlanId - يتم استرداده من خلال البحث عن
   - **التكلفة** `retail_cost` من تعريف المنتج - تعرض
   - **تاريخ التجديد** `custom_NextExecTime_hr` (قابل للقراءة البشرية) - تعرض
   - **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 تجديد تلقائي 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 الأمام`DELETE /crm/oam/remove_action_plan/{action_plan_id}`ة
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:
   - ستخراج product_id من ActionPlanId
   - إنشاء وظيفة توفير جديدة لذلك المنتج
   - يوفر الإضافة على الفور (تشغيل كتاب لعب التوفير)
   - تتلقى الخدمة فوائد الإضافة دون الانتظار للتجديد المجدول
4. تظهر نافذة حالة التوفير تقدم
5. عند النجاح، يتم تحديث الأرصدة على الفور

إضافة التجديد التلقائي:

يتم تمكين التجديد التلقائي من خلال توفير منتج إضافة يحتوي على `auto_renew` مضبوطة:

- المنتجات مع `auto_renew = "true"` - تنشئ تلقائيًا ActionPlans أثناء التوفير

- تسأل العميل إذا كانوا يريدون التجديد - `auto_renew = "prompt"` المنتجات مع التلقائي (نافذة حوار)
- لا تنشئ أبدًا - `auto_renew = "false"` المنتجات مع ActionPlans (شراء لمرة واحدة)

ينشئ كتاب لعب التوفير ActionPlan في CGRateS مع:

- فريدة تشفر معرفات الخدمة والمنتج والعميل ActionPlanId
- جدول التجديد (شهري، سنوي، فترة مخصصة)
- إجراء للتنفيذ (عادةً إعادة توفير نفس الإضافة)
- تاريخ انتهاء الصلاحية (إذا كان العقد له فترة محددة)

تنسيق تسمية ActionPlan

تتبع ActionPlans تنسيق تسمية موحد لترميز البيانات التعريفية:

التنسيق:

```
ServiceID_{service_uuid}__ProductID_{product_id}__CustomerID_{custome
```

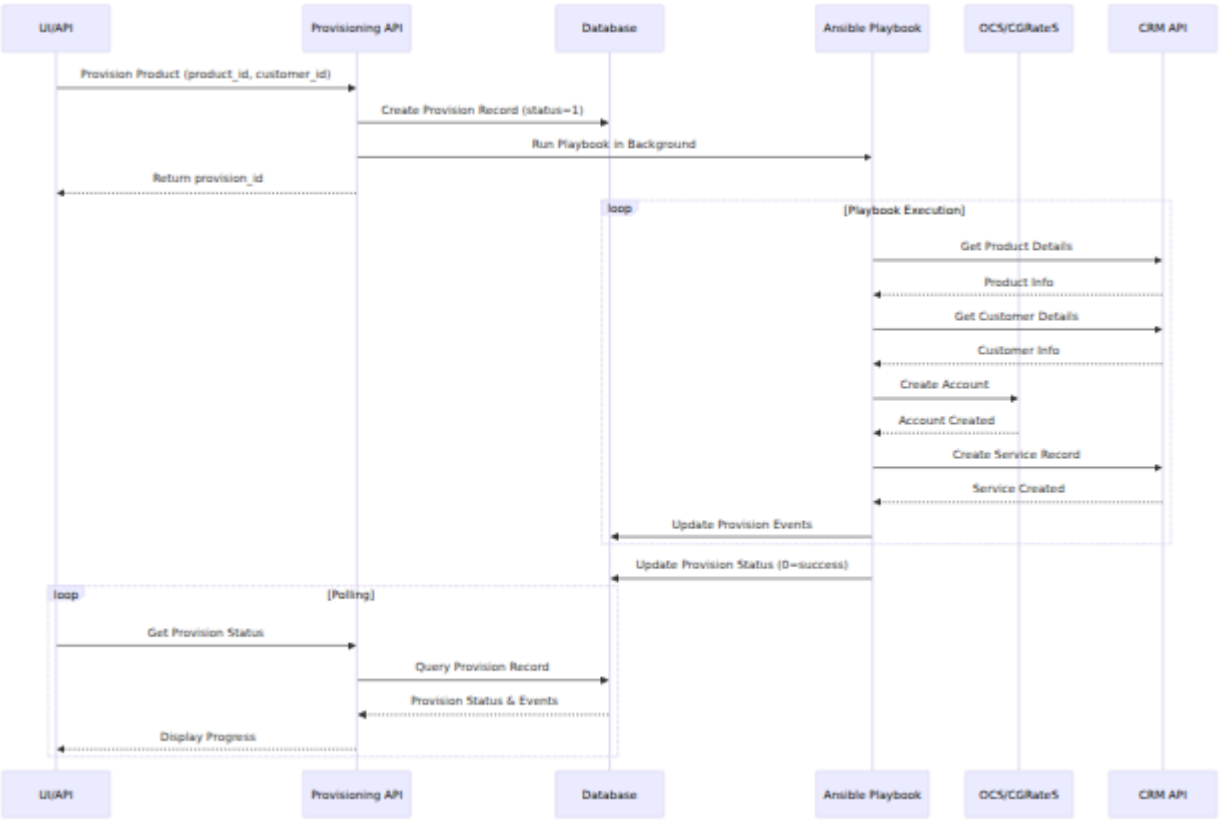
المثال:

```
```

نظام التوفير

تستخدم OmniCRM **Ansible** لأتمتة التوفير، والتكوين، وإلغاء توفير خدمات العملاء. تم تصميم نظام التوفير ليكون مرنًا، مما يسمح بتدفقات عمل معقدة مع الحفاظ على الاتساق والموثوقية.

للحصول على `SIM Card Provisioning <concepts_sim_provisioning>` انظر أيضًا: معلومات مفصلة حول توفير بطاقات SIM، بما في ذلك بطاقات SIM المحمولة والفعلية وeSIMs.



ملاحظة

للحصول على دليل كامل لرحلة المنتج إلى الخدمة مع أمثلة مفصلة على دفاتر تشغيل Ansible، واستراتيجيات التسعير، والسيناريوهات الواقعية، انظر `Complete Product Lifecycle Guide <guide_product_lifecycle>`.

نظرة عامة

عند طلب منتج أو الحاجة إلى تكوين خدمة، تقوم OmniCRM تنفذ دفتر **وظيفة توفير** بإنشاء تشغيل Ansible (OCS/CGRateS، واحد أو أكثر. تتفاعل هذه الدفاتر مع أنظمة خلفية مختلفة لتوفير الخدمة بالكامل (معدات الشبكة، واجهات برمجة ��لتطبيقات، إلخ.

يدعم نظام التوفير تدفقين رئيسيين:

1. **التوفير القياسي** - يتم تشغيله بواسطة الموظفين أو العملاء من خلال UI/API
2. **التوفير البسيط** - يتم تشغيله بواسطة أنظمة خارجية مثل OCS للعمليات الآلية

قيم حالة التوفير

يمكن أن تحتوي وظائف التوفير والمهام الفردية على الحالات التالية:

- **الحالة 0 (نجاح)** - اكتملت وظيفة التوفير بنجاح
- **الحالة 1 (جارية)** - وظيفة التوفير أو المهمة قيد التنفيذ حاليًا
- **الحالة 2 (فشلت - حرجة)** - حدث فشل حرج تسبب في فشل التوفير
- **الحالة 3 (فشلت - تم تجاهلها)** - فشلت مهمة ولكن كانت تحتوي على `ignore_errors: true`، لذا استمر التوفير

عندما تفشل وظيفة التوفير، ترسل OmniCRM إشعارات عبر البريد الإلكتروني إلى قائمة إشعارات الفشل المكونة مسبقًا مع معلومات خطأ مفصلة.

كيف تدفع المنتجات التوفير

تعريف **المنتج** هو المخطط لما يتم توفيره وكيف. عندما يختار المستخدم منتجًا للتوفير، يقرأ النظام عدة حقول رئيسية من تعريف المنتج لتحديد ما يجب القيام به.

حقول المنتج المستخدمة في التوفير

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 (مثل `['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 وظيفة التوفير

عند بدء التوفير، 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 إعداد لـ `refresh_token`

3. **تعيين المخزون** بناءً على `inventory_items_list`

اختيار المخزون UI/API قبل تشغيل دفتر التشغيل، يطلب:

- **SIM Card** - يختار المستخدم SIM متاح من المخزون
- **Mobile Number** - يختار المستخدم رقم هاتف متاح

مع نوع المخزون كمفتاح `extra_vars` تتم 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، تكون 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 يحتوي دفتر التشغيل على معرفات المخزون، يسترجع تفاصيل المخزون الكاملة من واجهة برمجة التطبيقات:

```
- 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 لدفتر التشغيل بعد ذلك استخدام هذه القيم لـ:

- توفير بطاقة SIM على HSS مع IMSI
- تكوين رقم الهاتف في نظام الفوترة
- تعيين عناصر المخزون للعميل
- إنشاء سجل الخدمة مع هذه التفاصيل

مثال من العالم الحقيقي: توفير بطاقة SIM المحمولة

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. يحدد تعريف المنتج `provisioning_play: "play_psim_only"`
2. يتطلب المنتج `inventory_items_list: ['SIM Card', 'Mobile Number']`
3. يختار المستخدم عناصر المخزون أثناء التوفير
4. يتم تمرير معرفات المخزون إل�ى دفتر التشغيل كـ `extra_vars`
5. يسترجع دفتر التشغيل تفاصيل المخزون الكاملة من واجهة برمجة التطبيقات
6. يستخدم دفتر التشغيل بيانات المخزون لتكوين معدات الشبكة
7. يحدد دفتر التشغيل المخزون على أنه مخصص للعميل

التراجع والتنظيف: نمط أفضل الممارسات

أفضل ممارسة حرجة: يجب أن يتعامل نفس دفتر التشغيل مع كل من التراجع عن التوفير الفاشل وإلغاء التوفير المتعمد باستخدام هيكل `block` و `rescue` في 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 التوفير ---
        - 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 التوفير ...

      rescue:
        # --- مهام التنظيف ---
        # يتم تشغيل هذا القسم عندما:
        # 1. (تراجع) تفشل أي مهمة في الكتلة
        # 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. لا تكرار في الكود

تتعامل نفس مهام التنظيف مع كلا السيناريوهين:

- إذا فشلت أي مهمة في `block`، يتم تنفيذ قسم `rescue` تلقائيًا: **فشل التوفير (التراجع)**

- **إلغاء التوفير المتعمد**: عند استدعائه مع `action: "deprovision"`، ينتقل التشغيل على الفور إلى `rescue` دفت��

2. ضمان التنظيف الكامل

عندما يفشل التوفير 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 المشتركين في HSS أو تعيينهم إلى حالة خاملة
- عدم بقاء أي توفير 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
- تستهلك موارد الشبكة

3. معالجة الفشل بشكل سلس مع ignore_errors

لاحظ أن كل مهمة تنظيف تستخدم `ignore_errors: True`. هذا مقصود لأنه:

- أثناء التراجع، قد لا تكون بعض الموارد قد تم إنشاؤها بعد
- نريد محاولة جميع مهام التنظيف حتى لو فشل بعضها
- يحدد التأكيد النهائي النجاح/الفشل العام

على سبيل المثال، إذا فشل التوفير عند "إنشاء حساب في OCS"، ستقوم عملية التنظيف بمحاولة:

- حذف حساب OCS (ستفشل، لكن سيتم تجاهلها)
- إزالة ملفات تعريف السمات (ستفشل، لكن سيتم تجاهلها)
- إعادة المخزون (تنجح)
- حذف المشترك في HSS (قد لا يوجد، يتم تجاهله)

4. تمييز إلغاء التوفير 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 كان ``action == "deprovision"``**: ينجح التأكيد، ينجح دفتر التشغيل (الحالة 0)
- **إذا لم يتم تعيين ``action`` أو != "deprovision"**: يفشل التأكيد، يفشل دفتر التشغيل (الحالة 2)

لذا فإن نفس كود التنظيف ينتج عنه حالات وظائف توفير 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 الخاملة**: عندما تكون بطاقات SIM قابلة لإعادة الاستخدام ويجب أن تبقى في HSS (`deprovision_subscriber: false`)

كيفية استخدام هذا النمط

للتوفير:

```
{
  "product_id": 1,
  "customer_id": 456,
  "provisioning_play": "play_psim_only"
}
```

إذا فشل التوفير، يحدث التراجع التلقائي عبر `rescue`.

لإلغاء التوفير:

```
{
  "service_id": 123,
  "service_uuid": "Service_abc123",
  "action": "deprovision",
  "provisioning_play": "play_psim_only"
}
```

يتخطى دفتر التشغيل مباشرة إلى قسم `rescue`، وينجح، وينفذ جميع عمليات التنظيف.

ملخص الفوائد

مصدر واحد للحقيقة: يتعامل دفتر تشغيل واحد مع التوفير وإلغاء التوفير ��� **عمليات** ذرية: إما تم توفيرها بالكامل أو تم تنظيفها بالكامل **لا موارد يتيمة**: التوفير الفاشل لا يترك أثر **صيانة أسهل**: التغييرات في منطق التوفير تنطبق تلقائيًا على التنظيف **تقليل الأخطاء**: لا فرصة لتزامن كود التوفير وإلغاء التوفير **قابل للاختبار**: يمكن اختبار منطق إلغاء التوفير عن طريق التشغيل مع `action: "deprovision"`

يجب اتباع هذا النمط في جميع دفاتر تشغيل التوفير لضمان الموثوقية والاتساق.

تجاوز متغيرات المنتج

يمكن تجاوز `provisioning_json_vars` المنتج في وقت التوفير. على سبيل المثال، قد يعرف:

```
{
  "provisioning_json_vars": "{\"monthly_cost\": 50,
\"data_limit_gb\": 100}"
}
```

ولكن عند التوفير، يمكنك تجاوز هذه القيم:

```
{
  "product_id": 1,
  "customer_id": 456,
  "monthly_cost": 45,
  "data_limit_gb": 150
}
```

ستستخدم `extra_vars` لـ هذا يسمح. المدمجة القيم المتجاوزة:

- تسعير مخصص للعملاء المحددين
- حدود بيانات مختلفة بناءً على العروض الترويجية
- اختبار مع معلمات مختلفة دون تعديل المنتج

منتجات بدون مخزون

لا تتطلب جميع المنتجات مخزونًا. على سبيل المثال، قد تحتوي إضافة بيانات أو مفتاح 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 لإضافة البيانات إليها
    "data_gb": 10,
    "access_token": "eyJ..."
}
```

يضيف دفتر التشغيل ببساطة البيانات إلى الخدمة الحالية دون الحاجة إلى أي عناصر مخزون.

تدفق التوفير القياسي

يتم بدء التوفير القياسي عندما:

- يضيف موظف خدمة إلى عميل من واجهة المستخدم
- يطلب عميل خدمة من خلال بوابة الخدمة الذاتية
- يتم استدعاء واجهة ب`PUT /crm/provision/`مجة التطبيقات مباشرة مع

"عندما تضغط على "توفير

إليك التدفق الكامل الذي يحدث عندما ينقر المستخدم على زر "توفير":

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 (مثل، `['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 من أنواع المخزون. يجب على المستخدم اختيار عناصر المخزون المتاحة قبل المتابعة.

3. تم النقر على زر التوفير

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. تتلقى واجهة برمجة التطبيقات الطلب

نقطة نهاية التوفير:

- 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 التوفير

تم إنشاء سجل قاعدة البيانات مع الحالة 1 (جارية):

```
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,  # جارية
    'task_count': 85,
    'initiating_user': 7,
    'created': '2025-01-10T14:30:00Z'
}
```

7. تم إنشاء خيط في الخلفية

يبدأ النظام خيطًا في الخلفية لتنفيذ دفتر التشغيل بشكل غير متزامن:

- دفتر التشغيل: `play_psim_only.yaml`
- المتغيرات: `extra_vars` المدمجة
- معرف التوفير: 456 (لتتبع الحالة)
- يتم تمرير رمز التحديث للتحديث أثناء التنفيذ

8. تعيد واجهة برمجة التطبيقات على الفور

يتم إعادة الاستجابة إلى واجهة المستخدم مع `provision_id`:

```
{
  "provision_id": 456,
  "provisioning_status": 1,
  "message": "تم إنشاء وظيفة التوفير"
}
```

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. ينفذ Ansible في الخلفية

تشغل دفتر التشغيل المهام بالتتابع:

- في قاعدة البيانات `Provision_Event` كل إكمال مهمة ينشئ سجل
- 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 الجارية حاليًا
- تظهر المهام الفاشلة رسائل خطأ في تفاصيل الحدث

التتبع في واجهة المستخدم:

بينما يتم تشغيل التوفير (الحالة 1)، يمكن للمستخدمين عرض:

- **صفحة تفاصيل الخدمة** - تظهر شارة حالة التوفير (جارية/نجاح/فشل)
- **سجل النشاط** - يسرد جميع أحداث التوفير مع الطوابع الزمنية
- **عرض تفاصيل التوفير** - يظهر تقدم المهمة تلو الأخرى مع توسيع/طي للتفاصيل

عرض مثال:

حالة التوفير: جارية (8 من 12 مهمة مكتملة)

✓ Get Product information from CRM API ✓ Fetch Customer details ✓ Assign SIM Card from inventory (ICCID: 89911012000003204510) ✓ Assign Mobile Number (555-0123) ⟳ Create account in OCS/CGRateS (قيد التقدم...) Configure network policies Create service record ...

11. يكتمل التوفير

تعيين الحالة النهائية:

- `provisioning_status: 0` - نجاح
- `provisioning_status: 2` - فشل (خطأ حرج)

تتوقف واجهة المستخدم عن الاستعلام وتعرض النتيجة:

- **نجاح**: علامة تحقق خضراء، الخدمة محددة نشطة، يمكن للمستخدم عرض تفاصيل الخدمة
- **فشل**: علامة X حمراء، تظهر رسالة خطأ، خيار لإعادة المحاولة أو الاتصال بالدعم
- **إشعار عبر البريد الإلكتروني**: إذا كان هناك فشل، يتم إرسال بريد إلكتروني إلى `provisioning.failure_list` في التكوين

المصادقة والتفويض

تتبع المستخدم

تتبع كل وظيفة توفير أي مستخدم بدأها:

- **بدأها المستخدم**: يتم تعيين حقل `initiating_user` إلى معرف المستخدم من رمز JWT الخاص بهم
- **مصادقة مفتاح API**: يستخدم أول معرف مستخدم إداري
- **مصادقة قائمة بيضاء IP**: يستخدم أول معرف مستخدم إداري

التحقق من الأذونات

يتحقق النظام من الأذونات قبل السماح بالتوفير:

- يحتاج الموظفون إلى إذن `CREATE_PROVISION`
- يمكن للعملاء فقط توفير الخدمات لحساباتهم الخاصة (إذن `VIEW_OWN_PROVISION`)

كيف تقوم Ansible بالمصادقة مع واجهة برمجة تطبيقات CRM

تحتاج دفاتر تشغيل Ansible إلى إجراء مكالمات واجهة برمجة التطبيقات المصرح بها مرة أخرى إلى CRM (لاسترجاع تفاصيل المنتج، وإنشاء خدمات، وتحديث المخزون، إلخ). تتم معالجة المصادقة من خلال **رموز Bearer** المرسلة إلى دفتر التشغيل.

يعتمد مصدر `access_token` على طريقة المصادقة المستخدمة لاستدعاء واجهة برمجة تطب��قات التوفير:

الطريقة 1: تسجيل دخول المستخدم (رمز Bearer)

عندما يقوم المستخدم بتسجيل الدخول عبر واجهة المستخدم على الويب:

1. يقوم المستخدم بالمصادقة: `POST /crm/auth/login`
2. يتلقى رمز JWT `access_token` (قصير الأمد، 15-30 دقيقة) و `refresh_token` (طويل الأمد)
3. يقوم بإجراء طلب التوفير مع رمز Bearer في الرأس
4. تستخرج واجهة برمجة التطبيقات رمز من `Authorization: Bearer ...` الرأس
5. تخزن الرمز للاستخدام أثناء التوفير
6. تمرره إلى Ansible كمتغير `access_token`

تدفق التنفيذ:

يستخرج النظام رمز Bearer من رأس المصادقة، يتحقق منه ويفك تشفيره، ثم يخزنه للاستخدام أثناء التوفير. يتم تمرير الرمز بعد ذلك إلى دفتر التشغيل كمتغير إضافي حتى يتمكن من المصادقة على مكالمات واجهة برمجة التطبيقات اللاحقة.

الطريقة 2: مفتاح API (رأس X-API-KEY)

بالنسبة للأنظمة الآلية التي تستخدم مفاتيح API:

1. يقوم النظام بإجراء الطلب: `PUT /crm/provision/` مع `X-API-KEY: your-api-key...` الرأس
2. تتحقق واجهة برمجة التطبيقات من مفتاح API مقابل `crm_config.yaml`
3. **تولد رمز JWT جديد في الوقت الفعلي** لأول مستخدم إداري
4. تخزن الرمز لاستخدامه في التوفير
5. تمرره إلى Ansible

لماذا توليد رمز؟

تتوقع دفاتر التشغيل استدعاء نقاط نهاية واجهة برمجة API مفاتيح وليست JWTs. هي سلاسل، لذا:

- تحقق من مفتاح API
- قم بتوليد JWT مؤقت، إذا كان صالحًا وله دور `admin`
- استخدم معرف أول مستخدم إداري كموضوع JWT
- يسمح الرمز لدفتر التشغيل بإجراء مكالمات واجهة برمجة التطبيقات المصرح بها

تدفق التنفيذ:

يتحقق النظام من مفتاح API يولد JWT مقابل المفاتيح المكونة. إذا كان صالحًا وله دور إداري، فإنه مؤقتًا باستخدام هوية مستخدم إداري، ثم يخزنه للاستخدام في التوفير.

الطريقة 3: قائمة بيضاء IP

لأنظمة داخلية موثوقة على الشبكات الخاصة:

1. يقوم النظام بإجراء الطلب من IP (مثل 192.168.1.100) مدرج في القائمة البيضاء
2. تتحقق واجهة برمجة التطبيقات من IP العميل مقابل `ip_whitelist` في `crm_config.yaml`
3. إذا كانت في القائمة البيضاء، **تولد رمز JWT** لأول مستخدم إداري **جديد**
4. تخزن الرمز لاستخدامه في التوفير
5. تمرره إلى Ansible

تدفق التنفيذ:

يتحقق النظام من IP العميل مقابل القائمة البيضاء. إذا كان مسموحًا، فإنه يولد JWT لمستخدم إداري ويخزنه للاستخدام في التوفير.

استخدام الرمز في دفاتر التشغيل

تتضمن كل مكالمة واجهة برمجة التطبيقات في دفتر التشغيل الرمز:

```
- 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

انتهاء الصلاحية 15-30) `access_token` قد تتجاوز دفاتر التشغيل طويلة الأمد (5-10 دقائق). بالنسبة للتوفير الذي يبدأه المستخدم، يمرر النظام كل 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 لمدير دفتر التشغيل:

1. اكتشاف استجابة 401 غير مصرح بها
2. استدعاء `POST /crm/auth/refresh` مع `refresh_token`
3. تلقي `access_token` جديد
4. إعادة محاولة الطلب الفاشل

بالنسبة لمصادقة مفتاح API/IP القائمة البيضاء، يمكن أن تحتوي الرموز المولدة على انتهاء صلاحية أطول (1-2 ساعة) نظرًا لأن هذه أنظمة آلية موثوقة.

عملية التوفير

1. **إنشاء الوظيفة**

   عند استلام طلب توفير، يقوم النظام:

   - يتحقق من الطلب ويتحقق من الأذونات
   - يحمل دفتر التشغيل Ansible المحدد في تعريف المنتج
   - ينشئ سجل `Provision` (جارية) 1 في قاعدة البيانات مع الحالة
   - يستخرج المتغيرات من تعريف المنتج وجسم الطلب
   - يلتقط رموز المصادقة للوصول إلى واجهة برمجة التطبيقات

2. **معالجة الرموز**

   تحتاج دفاتر تشغيل Ansible إلى المصادقة مع واجهة برمجة تطبيقات CRM لاسترجاع البيانات وإجراء التغييرات. يتعامل نظام التوفير مع ذلك بطريقتين:

   - **رمز Bearer (JWT)**: للتوفير الذي يبدأه المستخدم، يتم استخدام `refresh_token` من الطلب لتوليد رموز وصول جديدة أثناء تنفيذ دفتر التشغيل
   - **مفتاح API/IP Auth**: بالنسبة للأنظمة الآلية، يتم تمرير `access_token` مباشرة إلى دفتر التشغيل

3. **التنفيذ في الخلفية**

   يتم تشغيل دفتر التشغيل في خيط خلفي. يسمح ذلك لواجهة برمجة التطبيقات بالعودة على الفور بينما يستمر التوفير بشكل غير متزامن.

   أثناء التنفيذ:

   - كل إ��مال/فشل مهمة ينشئ سجل `Provision_Event`
   - يراقب معالج الحدث 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 يقوم دفتر تشغيل Ansible بتنفيذ هذه العمليات:

- يسترجع معلومات المنتج من واجهة برمجة التطبيقات
- يسترجع معلومات العميل من واجهة برمجة التطبيقات
- يعين عناصر المخزون (بطاقات SIM، عناوين IP، إلخ، أرقام الهواتف.)
- ينشئ حسابات في OCS/OCS
- يقوم بتكوين معدات الشبكة
- ينشئ سجل الخدمة في واجهة برمجة تطبيقات CRM
- يضيف معاملات تكاليف الإعداد
- يرسل رسائل بريد إلكتروني/SMS ترحيبية للعملاء

5. **معالجة الأخطاء**

   تستخدم دفاتر تشغيل Ansible أقسام `block` و `rescue` للتراجع:

   - إذا فشلت مهم�� حرجة، يقوم قسم الإنقاذ بإزالة التوفير الجزئي
   - يتم وضع علامة 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 التوفير 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
```

يظهر هذا دفتر التشغيل التدفق النموذجي:

1. استرجاع تفاصيل المنتج من واجهة برمجة التطبيقات CRM
2. خدمة فريدة UUID توليد
3. إنشاء حساب الفوترة في OCS
4. إنشاء سجل الخدمة عبر واجهة برمجة التطبيقات CRM
5. إضافة معاملات تكاليف الإعداد
6. إذا فشل أي شيء، يقوم قسم `rescue` بإزالة حساب OCS

تدفق التوفير البسيط

تم تصميم التوفير البسيط للأنظمة الآلية التي تحتاج إلى تحفيز التوفير دون تفاعل المس��خدم. أكثر حالات الاستخدام شيوعًا هي تحفيز OCS لتوفير الإضافات عبر ActionPlans.

نقاط نهاية التوفير البسيط

توفر OmniCRM نقطتي نهاية للتوفير البسيط:

- `POST /crm/provision/simple_provision_addon/service_id/<id>/product_id/<id>`

  للتوفير الآلي للإضافات (مثل الرسوم المتكررة، الشحنات التلقائية)

- `POST /crm/provision/simple_provision_addon_recharge/service_id/<id>/product_id/<id>`

  لعمليات الشحن السريعة التي تحتاج إلى ردود فورية

المصادقة للتوفير البسيط

تستخدم نقاط نهاية التوفير 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 توليد رمز وصول ويتم تمريره إلى دفتر التشغيل

مثال: استدعاء ActionPlan من OCS

يمكن تكوين OCS لاستدعاء نقطة نهاية التوفير 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 يحفز دفتر التشغيل المرتبط (مثل `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
```

يؤدي هذا دفتر التشغيل إلى:

1. الحصول على تفاصيل الخدمة والمنتج من واجهة برمجة التطبيقات
2. استرجاع طريقة الدفع الخاصة بالعميل
3. شحن العميل عبر واجهة برمجة التطبيقات Stripe
4. إضافة رصيد نقدي إلى OCS
5. تسجيل المعاملة في CRM
6. إرسال رسالة SMS تأكيدية (مع `ignore_errors: True` حتى لا تفشل الأخطاء الوظيفة)

سلاسل التوفير

سلاسل OmniCRM بالنسبة للمنتجات المعقدة التي تتطلب خطوات توفير متعددة، تدعم **التوفير**. تنفذ سلسلة متعددة من دفاتر التشغيل بالتتابع، مع تمرير السياق بينها.

حالة استخدام مثال: خدمة مجمعة توفر:

1. خدمة الإنترنت الأساسية (تنشئ سجل الخدمة الرئيسي)
2. إضافة IPTV (تستخدم `service_id` من الخطوة 1)
3. إضافة IP ثابت (تستخدم `service_id` من الخطوة 1)

تقوم خدمة التوفير 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 التوفير، يلتقط النظام معلومات مفصلة للمساعدة في تشخيص المشكلة.

سيناريوهات الفشل الشائعة

فشل المهام الحرجة (الحالة 2)

تسبب هذه الفشل في وظيفة التوفير بالكامل:

- مكالمات واجهة برمجة التطبيقات التي تعيد رموز حالة غير متوقعة
- فشل التأكيدات (مثل `assert: that: response.status == 200`)
- العناصر المخزونة المطلوبة المفقودة
- معدات الشبكة غير قابلة للوصول
- بيانات اعتماد غير صالحة أو رموز منتهية الصلاحية
- أخطاء OCS/OCS

الفشل المتجاهل (الحالة 3)

تسجل هذه ولكن لا تفشل الوظيفة:

- البريد الإلكتروني الاختيارية/SMS فشل إشعارات
- عمليات البحث عن البيانات غير الحرجة (المعلمة بـ `ignore_errors: True`)
- عمليات التنظيف أثناء إلغاء التوفير

الأخطاء الفتاكة

تمنع هذه تشغيل دفتر التشغيل على الإطلاق:

- أخطاء بناء جملة YAML في دفتر التشغيل
- متغيرات غير معرفة في دفتر التشغيل
- ملفات دفتر التشغيل المفقودة
- فشل الاتصال بوحدة التحكم Ansible

عند حدوث خطأ فتاك، ينشئ النظام حدث خطأ خاص يحتوي على:

- رمز خروج Ansible
- stdout (يحتوي على تفاصيل خطأ بناء الجملة) الكامل
- stderr (يحتوي على أخطاء وقت التشغيل) الكامل
- قائمة بالأسباب الشائعة لذلك النوع من الفشل
- جميع المتغيرات المرسلة إلى دفتر التشغيل

رسائل البريد الإلكتروني لإشعار الأخطاء

عندما يفشل التوفير (الحالة 2)، يتم إرسال بريد إلكتروني تلقائيًا إلى قائمة إشعارات الفشل (المكونة مسبقًا في `crm_config.yaml` `provisioning.failure_list`).

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
- رسائل الخطأ، والأسباب الشائعة، Ansible بالنسبة للأخطاء الفتاكة: إخراج

مراقبة وظائف التوفير

توفر OmniCRM مقاييس شاملة لمراقبة أداء التوفير. للحصول على تفاصيل كاملة حول مقاييس التوفير بما في ذلك مدة الوظيفة، ومعدلات النجاح، وعدد المهام، وتتبع الأخطاء، انظر المراقبة والمقاييس.

واجهة برمجة تطبيقات حالة التوفير

للتحقق من حالة وظيفة التوفير:

```
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": "{...}"
    }
  ]
}
```

سرد وظائف التوفير

للحصول على قائمة مرتبة من جميع وظائف التوفير:

```
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 التوفير الجزئي
- **استخدم ignore_errors بحكمة**: فقط للعمليات الاختيارية حقًا
- **سجل المتغيرات المهمة**: اس��خدم مهام `debug` لتسجيل القيم الرئيسية لأغراض تصحيح الأخطاء
- **تحقق من الاستجابات**: استخدم `assert` للتحقق من أن استجابات واجهة برمجة التطبيقات كما هو متوقع
- **العمومية**: صمم دفاتر التشغيل لتكون قابلة للتشغيل مرة أخرى بأمان

المصادقة

- **التوفير الذي يبدأه المستخدم**: استخدم دائمًا `refresh_token` لـ دفاتر التشغيل طويلة الأمد
- **التوفير الآلي**: استخدم قائمة بيضاء IP أو مفاتيح API مع رموز وصول مولدة
- **انتهاء صلاحية الرمز**: يضمن `refresh_token` تجديد رموز الوصول حسب الحاجة

معالجة الأخطاء

- **قدم السياق**: قم بتضمين `customer_id`، و`service_id`، وتفاصيل العملية في رسائل الخطأ
- **أبلغ بشكل مناسب**: تؤدي الفشل الحرج إلى تحفيز رسائل البريد الإلكتروني، ولكن لا تزعج للفشل المتوقع
- **معلومات تصحيح الأخطاء**: التقاط أجسام الطلب/الاستجابة الكاملة في سجلات `Provision_Event`

الأمان

- **تحقق من المدخلات**: تحقق من `customer_id`، و`product_id`، و`service_id` قبل التوفير
- **تحقق من الأذونات**: تحقق من أن المستخدمين يمكنهم فقط توفير خدمات للعملاء المصرح لهم
- **البيانات الحساسة**: استخدم نظام الحذف لإزالة كلمات المرور/المفاتيح من السجلات
- **قائمة بيضاء IP**: قيد نقاط نهاية `simple_provision` على الأنظمة الموثوقة فقط

الأداء

- **التنفيذ في الخلفية**: لا تحظر استجابات واجهة برمجة التطبيقات في انتظار التوفير
- **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 التوفير العامة لـ Ansible
- `concepts_sim_provisioning` - توفير بطاقة SIM (و eSIMالفعلية)
- `concepts_api` - مصادقة واستخدام واجهة برمجة التطبيقات CRM
- `concepts_products_and_services` - تعريفات المنتجات والخدمات
- `administration_inventory` - إدارة المخزون للتوفير

- المراقبة والمقاييس - مقاييس أداء التوفير والمراقبة

سجل نشاط العميل

يتم تسجيل كل تغيير تم إجراؤه على عميل أو جهة اتصال أو موقع أو خدمة أو معلمات مالية، مثل المعاملات والفواتير وطرق الدفع، في سجل النشاط.

يسمح لنا ذلك بتتبع التغييرات التي تم إجراؤها على النظام، ومن قام بها، ومتى تم ذلك، وهو مفيد لتدقيق التغييرات، وتتبع المشكلات، على سبيل المثال، إذا قال عميل إنه لم يتلقَ فاتورة، يمكننا التحقق من سجل النشاط لمعرفة ما إذا كانت قد أُرسلت، أو إذا تمت إزالة جهة اتصال، يمكننا معرفة من أزالها ومتى.

سجل النشاط هو قائمة زمنية بالتغييرات، مع أحدث التغييرات في الأعلى، والتغييرات الأقدم في أسفل القائمة، والتي يمكن تصفيتها من علامات التبويب.

لا يمكن حذف سجلات النشاط، ولكن يمكن تصفيتها، ويمكن عرض التفاصيل لمعرفة ما تم تغييره، ومن قام بذلك.

إضافة خدمة

الخدمات هي الأشياء التي نقوم بفوترة العميل من أجلها، يمكن أن تكون خدمات الإنترنت المنزلية، خدمات الهاتف المحمول، أو حتى خدمات مجردة مثل تأجير شبكة فرعية أو توفير كهرباء مقاسة لرف.

الخدمة هي مجرد حالة من `Product <concepts_products_and_services>` لعميل معين، يتم اختيارها من كتالوج المنتجات وتوفيرها للعميل.

إذا لم تكن قد `أنشأت عميلًا <basics_create_customer>` بالفعل، فسيتعين عليك القيام بذلك أولاً، حيث ترتبط الخدمات بالعملاء، وستحتاج أيضًا إلى تحديد طريقة دفع للعميل، حيث تولد الخدمات رسومًا تحتاج إلى الدفع.

يمكن للعملاء توفير خدماتهم الخاصة (إذا سمحنا لهم بذلك)، أو يمكن لموظفي خدمة العملاء توفير الخدمات للعميل.

استنادًا إلى تعريف المنتج، هناك قواعد تتعلق بمن يمكنه شراء منتج، مثل السماح فقط للعملاء التجاريين بشراء منتج تجاري، أو السماح فقط للعملاء الذين لديهم خدمة هاتف محمول بشراء إضافة موبايل.

يمكن أن تحتوي الخدمات أيضًا على مكونات استخدام، مثل استخدام البيانات، دقائق المكالمات، أو رسوم استخدام أخرى، ويمكن أن يكون لها رسوم متعددة مرتبطة بها، مثل الرسوم الشهرية، الرسوم لمرة واحدة، أو رسوم الاستخدام، يمكننا عرض ذلك من زر "الاستخدام".

تدعم العديد من الخدمات الإضافات، على سبيل المثال، تحتوي خدمة الهاتف المحمول على جميع الشحنات المتاحة من قائمة الإضافات، وقد تحتوي خدمة الإنترنت المنزلية على عنوان IP ثابت أو بيانات إضافية متاحة كإضافة، مرة أخرى، كل هذا يتم تعريفه في `product catalog<concepts_products_and_services>`.

سير عمل تخصيص الخطط

تتيح ميزة تخصيص الخطط للموظفين **توفير الخدمات للعملاء** من خلال اختيار المنتجات من الكتالوج وبدء عملية التوفير. هذه هي الطريقة الأساسية لإنشاء خدمات جديدة عندما لا يقوم العملاء بالتوفير الذاتي.

نظرة عامة

يتضمن تخصيص 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 تخصيص الخطة

من كتالوج المنتجات:

من صفحة العميل:

من صفحة الإضافات:

تفتح واجهة تخصيص 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 شريط تمرير أو شبكة:

انقر على **"تخصيص للعميل"** للمتابعة.

الخطوة 2: اختيار العميل

إذا لم تكن بالفعل على صفحة العميل، سيُطلب منك اختيار عميل.

بحث عن العملاء:

> ابحث عن العملاء: [جون ▼]
>
> العملاء المتطابقون: • جون دو (ID: 456) • جون سميث (ID: 789) • مؤسسات جونسون (ID: 123)

اكتب للبحث عن:

- اسم العميل
- ID العميل
- عنوان البريد الإلكتروني
- رقم الهاتف

اختر العميل من القائمة المنسدلة.

تحذير

تأكد من أنك قد اخترت العميل الصحيح قبل المتابعة. يتطلب تخصيص خطة للعميل الخطأ تدخلًا يدويًا لتصحيح ذلك.

الخطوة 3: تكوين المخزون (إذا لزم الأمر)

تظهر أدوات المخزون، إذا كان المنتج يتطلب عناصر مخزون (محددة في `inventory_items_list`) اختيار المخزون.

مثال على أداة اختيار 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
- تُظهر العناصر المتاحة فقط (الحالة: "متوفر" أو "جديد")
- يتم تحميل القوائم المنسدلة ديناميكيًا بناءً على أسماء قوالب المخزون
- بمجرد اختيارها، يتم حجز العناصر مؤقتًا

ماذا يحدث:

- يتم تمرير عناصر المخزون المحددة كمتغيرات إلى كتاب التشغيل الخاص بالتوفير

- أثناء التوفير، يتم تعيين العناصر للخدمة والعميل
- "تتغير حالة العنصر من "متوفر" إلى "مخصص

الخطوة 4: تكوين التجديد التلقائي (اختياري)

بالنسبة للخدمات المتكررة، قد يُطلب منك تعيين التجديد التلقائي:

> هل ترغب في تمكين التجديد التلقائي لهذه الخدمة؟
>
> عند التمكين، 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 ممارسة: افتراضيًا "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 موبايل - 20 جيجابايت (ID: 123) العميل: جون سميث
>
> رقم الهاتف المحمول: +44 7700 900123 • بطاقة SIM: SIM-00123 (ICCID: 8944...0001) المخزون:
>
> 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 الحقيقي:

> ✓ التحقق من حساب العميل ✓ تخصيص بطاقة SIM (ICCID: 8944...0001) ✓ تخصيص رقم الهاتف المحمول (+44 7700 900123) ⟳ تكوين حساب OCS (قيد التقدم...) إنشاء سجل الخدمة إرسال بريد الترحيب
>
> التقدم: 3 من 6 مهام مكتملة

تستعلم النافذة عن واجهة برمجة التطبيقات الخاصة بالتوفير كل 0.2 ثانية للحصول على تحديثات الحالة.

مؤشرات التقدم:

- ☑ مكتمل بنجاح
- ⟳ قيد التشغيل حاليًا
- ⏸ معلق (لم يبدأ)
- ✗ فشل (إذا حدثت أخطاء)

الخطوة 7: الاكتمال

النجاح:

> تم توفير الخدمة بنجاح لجون سميث
>
> الخدمة: 789 اسم الخدمة: موبايل - +44 7700 900123 الحالة: نشط ID
>
> [إغلاق] [عرض الخدمة]

انقر على **"عرض الخدمة"** لفتح صفحة تفاصيل الخدمة.

الفشل:

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] [إعادة 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 تخصيص إضافة (فئة: "إضافة") لعميل لديه خدمة بالفعل:

1. **الكشف التلقائي عن الخدمة:**
   - يجد النظام خدمات العميل الحالية
   - يتم تصفية 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 تخصيص الإضافة ضد service_id الموجود
   - يستخدم حساب OCS للخدمة الحالية
   - يتم تعيين المخزون (إن وجد) للخدمة الحالية
3. **كتاب تشغيل التوفير:**
   - كتاب تشغيل مختلف عن الخدمات المستقلة
   - عادةً ما يضيف رصيدًا أو ميزات أو معدات إلى الحساب الحا��ي

مثال:

> خدمات موبايل الحالية: • موبايل - +44 7700 900123 (ID: 789) • موبايل - +44 7700 900456 (ID: 790)
>
> أي خدمة يجب أن تتلقى هذه الإضافة؟ [موبايل - +44 7700 900123 ▼]
>
> [متابعة] [إلغاء]

التوفير لعملاء الأعمال

قد يكون لدى عملاء الأعمال متطلبات إضافية:

- **اختيار الموقع** - اختر أي موقع تجاري يتلقى الخدمة
- **تعيين جهة الاتصال** - تعيين جهات الاتصال للفوترة/التقنية
- **حقول مخصصة** - 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]

تخصيص الخدمة بالجملة

لتخصيص نفس الخطة لعدة عملاء (مثل الهجرات الجماعية):

1. استخدم استيراد CSV (إذا كان متاحًا)
2. أو خصص بشكل فردي مع إعدادات القالب
3. يجب أن يكون المخزون متاحًا بالجملة
4. راقب قائمة التوفير لتجنب التحميل الزائد

سير العمل الشائعة

سير العمل 1: خدمة موبايل جديدة

1. يدخل العميل إلى المتجر ويريد خدمة موبايل
2. يفتح الموظف **المنتجات ← الخطط**
3. يتبدل إلى نوع العميل **الفردي**
4. يختار "**موبايل - خطة 20 جيجابايت**"
5. ينقر على "**تخصيص للعميل**"
6. يبحث عن العميل برقم الهاتف: "+1234567890"
7. يختار **جون سميث** من النتائج
8. يختار بطاقة SIM من أداة اختيار المخزون
9. يختار رقم الهاتف المحمول المتاح
10. يمكّن **التجديد التلقائي**
11. يؤكد ويخصص
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 ��لتبويب **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/مواقع متعددة)
7. اختر مودم CPE من المخزون
8. حدد تاريخ التثبيت (إذا لزم الأمر)
9. خصص الخدمة
10. أنشئ تذكرة التثبيت

سير العمل 3: تخصيص 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 الموبايل
5. يختار "زيادة بيانات 5 جيجابايت"
6. يتم تفويض الدفع
7. يتم تخصيص الإضافة على الفور
8. يتلقى العميل زيادة البيانات الفورية

استكشاف الأخطاء وإصلاحها

"لا توجد منتجات متاحة"

- **السبب:** الفلاتر تستبعد جميع المنتجات
- **الحل:**
  - تبديل نوع العميل (فردي مقابل تجاري)
  - تحقق من أن كتالوج المنتجات يحتوي على منتجات مفعلّة

- تحقق من أن المنتجات تتطابق مع أهلية العميل

"لا يوجد مخزون متاح"

- **السبب:** عناصر المخزون خارج المخزون أو جميعها مخصصة
- **الحل:**
  - أضف المزيد من عناصر المخزون إلى النظام
  - تحقق من حالة العنصر (يجب أن تكون "جديدة" أو "متاحة")
  - تحقق من أن أسماء قوالب المخزون تتطابق مع متطلبات المنتج

"لم يتم العثور على العميل"

- **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

يتوقف التوفير

- **السبب:** خطأ في كتاب التشغيل أو النظام الخارجي غير متاح
- **الحل:**
  - انتظر حتى انتهاء المهلة (عادةً 5 دقائق)
  - تحقق من سجلات التوفير للخطأ المحدد
  - تحقق من أن أنظمة OCS والشبكة متصلة بالإنترنت
  - أعد محاولة التوفير بعد إصلاح المشكلة

ينجح التوفير ولكن الخدمة لا تعمل

- **السبب:** تم إنشاء حساب OCS ولكن الشبكة لم يتم تحديثها
- **الحل:**
  - تحقق من أن OCS لديه حساب
  - تحقق من أن بطاقة SIM مفعلة في HLR/HSS
  - تحقق من توفير الشبكة (RADIUS، DPI، إلخ)
  - راجع مهام كتاب التشغيل للخطوات المفقودة

أفضل الممارسات

قبل التخصيص:

- تحقق من أن العميل لديه وسيلة دفع صالحة في السجلات
- تأكد من أهلية العميل للمنتج (سكنية مقابل تجارية)
- تأكد من توفر المخزون المطلوب
- راجع شروط المنتج والأسعار مع العميل

أثناء التخصيص:

- تحقق مرة أخرى من اختيار العميل قبل التأكيد
- اختر عناصر المخزون الصحيحة (تحقق من أرقام السلاسل)
- قم بتمكين التجديد التلقائي للراحة (ما لم يعترض العميل)
- راقب تقدم التوفير حتى الاكتمال

بعد التخصيص:

- تحقق من ظهور الخدمة في قائمة خدمات العميل
- تحقق من أن حالة الخدمة "��شطة"
- تأكد من تعيين المخزون بشكل صحيح
- أرسل بريد الترحيب أو التعليمات للعميل
- اختبر الخدمة إذا كان ذلك ممكنًا (قم بإجراء مكالمة اختبار، تحقق من البيانات)

بالنسبة للإضافات:

- تأكد من أن الإضافة متوافقة مع الخدمة الحالية
- اشرح الفوترة (مرة واحدة مقابل متكررة)
- تحقق من الدفع قبل التوفير
- تحقق من تحديث الرصيد على الفور بعد تخصيص 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

تعديل خدمة

يمكن تعديل الخدمات من قبل العميل النهائي عبر `Self-Care Portal <self_care_portal>`، 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 للعميل.

إضافة ملحقات الخدمة

تعزز الملحقات الخدمات الحالية بميزات إضافية، أو حصة بيانات، أو أجهزة. تشمل أنواع الملحقات الشائعة:

- **تعبئة البيانات** - حصة بيانات إضافية (مثل، "5GB Data Boost")
- **ترقيات الميزات** - قدرات إضافية (مثل، "المكالمات الدولية")
- **تأجير الأجهزة** - إضافات المعدات (مثل، "مودم WiFi 6")
- **الخدمات المميزة** - ميزات محسنة (مثل، "عنوان IP ثابت")

الوصول إلى كتالوج الملحقات

من صفحة تفاصيل الخدمة:

1. انتقل إلى الخدمة التي تريد تحسينها
2. انقر على زر **"إضافة ملحق"** أو **"تصفح الملحقات"**
3. يفتح كتالوج الملحقات، مصفاة لعرض الملحقات المتوافقة فقط

التصفية التلقائية:

يقوم النظام بتصفية الملحقات تلقائيًا بناءً على:

- **نوع الخدمة** - يعرض فقط الملحقات المطابقة لنوع الخدمة (محمول، إنترنت، VoIP، إلخ.)
- **نوع العميل** - تصفية حسب العميل السكني مقابل العميل التجاري
- **توافق الخدمة** - يتحقق مما إذا كانت الخدمة تلبي متطلبات الملحق
- **التوافر** - يعرض فقط المنتجات المفعلة

على سبيل المثال، إذا كنت تشاهد خدمة موبايل سكنية، ستظهر لك فقط الملحقات المخصصة للعملاء السكنيين.

عرض الملحقات

تُعرض الملحقات في دوار تفاعلي يظهر:

عرض بطاقة المنتج:

المعلومات المعروضة:

- أيقونة المنتج
- اسم المنتج
- قائمة الميزات (نقاط مرقمة)
- تكلفة الإعداد
- التكلفة الشهرية/الدورية
- رابط الشروط والأحكام
- زر "إضافة إلى الخدمة"

توفير ملحق

الخطوة 1: اختيار الملحق

انقر على بطاقة الملحق المرغوب، ثم انقر على **"إضافة إلى الخدمة"**

الخطوة 2: اختيار المخزون (إذا لزم الأمر)

إذا كان الملحق يتطلب مخزونًا ماديًا (مثل، تأجير الأجهزة)، تظهر أداة اختيار 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 الدورية، قد يُطلب منك:

> هل ترغب في تفعيل التجديد التلقائي لهذا الملحق؟
>
> [نعم] [لا]

- **نعم** - يتم تجديد الملحق تلقائيًا في كل فترة فواتير
- **لا** - شراء لمرة واحدة، يجب على العميل التجديد يدويًا

الخطوة 4: تأكيد وتوفير

راجع تفاصيل الملحق وانقر على **"تأكيد"**

تظهر نافذة التوفير تعرض التقدم في الوقت الحقيقي:

> ✓ التحقق من الدفع ✓ تعيين المخزون ⟳ تكوين الخدمة (قيد التقدم...) إنشاء المعاملة إرسال بريد إلكتروني للتأكيد

بمجرد الانتهاء، يظهر الملحق في قائمة الملحقات الخاصة بالخدمة.

التوفير خلف الكواليس:

عند إضافة ملحق، يقوم النظام:

1. بالتحقق من إمكان��ة شراء العميل للملحق
2. بتشغيل كتاب التشغيل الخاص بالملحق (`provisioning_play`)
3. بتعيين أي عناصر مخزون مطلوبة للخدمة
4. بإنشاء سجل المعاملة للفوترة
5. بتحديث تكوين الخدمة (OCS، إلخ.أنظمة الشبكة،)
6. بإرسال إشعار تأكيد للعميل

للحصول على تفاصيل فنية حول توفير الملحقات، راجع `Complete Product Lifecycle Guide - Adding Addons <guide_product_lifecycle>`.

عرض الملحقات النشطة

تظهر الملحقات النشطة في صفحة تفاصيل الخدمة في قسم **"الملحقات"**:

عرض المثال:

> 5GB Data Boost
> أضيفت: 10 يناير 2025 تنتهي: 17 يناير 2025 الحالة: نشطة التكلفة: £5.00
>
> تأجير مودم WiFi 6
> 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. ابحث عن الملحق في قسم **"الملحقات النشطة"**
3. انقر على **"إزالة"** أو **رمز سلة المهملات** بج��ار الملحق
4. أكد الإزالة

ماذا يحدث:

- **الملحقات المادية** - يتم تشغيل كتاب التشغيل لإلغاء التوفير، ويتم وضع علامة على المخزون للعودة
- **الملحقات الافتراضية** - تتم إزالة الفوائد على الفور
- **الملحقات التي تجدد تلقائيًا** - يتم إلغاء التجديدات المستقبلية
- **تم إنشاء معاملة** - يتم إضافة أي مبلغ مقسوم إذا كان ذلك مناسبًا

تحذير

إعادة المعدات. سيقوم النظام بوضع (المودم، CPE، إلخ) عادةً ما تتطلب إزالة الملحقات المادية علامة على المخزون للعودة وقد يرسل تعليمات العودة للعميل.

سيناريوهات الملحقات الشائعة

السيناريو 1: نفاد البيانات للعميل

1. يتصل العميل بالدعم: "لقد استخدمت حصتي الشهرية"
2. يتنقل الموظف إلى خدمة الهاتف المحمول للعميل
3. ينقر على "إضافة ملحق"
4. يختار "5GB Data Boost"
5. يتم التوفير على الفور
6. يتلقى العميل تعبئة بيانات فورية

السيناريو 2: يحتاج العمل إلى IP ثابت

1. يطلب العميل التجاري VPN ثابت لـ IP
2. يفتح الموظف خدمة الإنترنت للعميل
3. يتصفح الملحقات، يختار "عنوان IP ثابت"
4. يقوم النظام بتوفير IP (المخزون) من المجموعة المتاحة
5. يقوم بتكوين التوجيه في معدات الشبكة
6. يتلقى العمل تفاصيل تكوين IP

السيناريو 3: تأجير المعدات 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 قدرها £10 شهريًا إلى الحساب

استكشاف الأخطاء وإصلاحها

"لا توجد ملحقات متوافقة متاحة"

- **السبب:** لا تتطابق أي ملحقات مع نوع الخدمة أو نوع العميل

- **الإصلاح:** تحقق من أن كتالوج المنتجات يحتوي على ملحقات تتطابق مع إعدادات `service_type` و `residential`/`business`

فشل توفير الملحق

- **السبب:** خطأ في كتاب التشغيل للتوفير أو عدم توفر المخزون
- **الإصلاح:**
  - تحقق من سجلات التوفير للخطأ المحدد
  - تحقق من أن عناصر المخزون متوفرة (إذا لزم الأمر)
  - راجع سجلات كتاب التشغيل في نظام التوفير

أداة اختيار المخزون لا تظهر أي عناصر

- **السبب:** لا توجد عناصر مخزون متاحة من النوع المطلوب
- **الإصلاح:**
  - 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
- **الإصلاح:**
  - تحقق من أن الخدمة تحتوي على `auto_renew: true`
  - تحقق من أن العميل لديه طريقة دفع صالحة في الملف
  - راجع الوظ��ئف 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 المادية

إدارة الخدمة

واجهة إدارة الخدمة توفر أدوات شاملة لعرض وتصنيف وإدارة خدمات العملاء عبر مؤسستك.

المستندات ذات الصلة: `إضافة خدمات <csa_add_service>`, `استخدام الخدمة <csa_service_usage>`, `تعديل الخدمات <csa_modify>`, `المنتجات والخدمات <concepts_products_and_services>`.

الوصول إلى قائمة الخدمات

انتقل إلى:

أو مباشرةً:

هذا يعرض جميع الخدمات مع قدرات التصنيف، والترتيب، والإدارة الجماعية.

نظرة عامة على قائمة الخدمات

تظهر قائمة الخدمات الخدمات في جدول مع:

الأعمدة:

- **معرف الخدمة** - معرف فريد

- **اسم الخدمة** - اسم وصفي (مثل "محمول - +44 7700 900123")
- **اسم العميل** - العميل المرتبط (رابط قابل للنقر)
- **نوع الخدمة**، VoIP، إنترنت، IPTV - محمول
- **الحالة** - نشط، غير نشط، معلق
- **تاريخ البدء** - متى بدأت الخدمة
- **تاريخ الانتهاء** - متى تنتهي/تنتهي الخدمة
- **التكلفة الشهرية** - السعر بالتجزئة
- **الإجراءات** - عرض، تعديل، إضافات، استخدام، حذف

تصنيف الخدمات

علامات الحالة

قم بتصفية الخدمات بسرعة حسب الحالة باستخدام شريط علامات التبويب:

- **جميع النتائج** - يعرض جميع الخدمات بغض النظر عن الحالة
- **نشط** - الخدمات النشطة حاليًا (عرض افتراضي)
- **غير نشط** - الخدمات الملغاة أو المنتهية
- **معلق** - الخدمات المعطلة مؤقتًا (عدم الدفع، الاحتيال، إلخ)

النقر على علامة التبويب يحدث القائمة على الفور. يتم تمييز علامة التبويب النشطة.

تصفية نوع الخدمة

تصفية حسب نوع الخدمة:

اختر نوعًا واحدًا أو عدة أنواع لعرض الخدمات المطابقة فقط.

حالات الاستخدام:

- **محمول** - عرض جميع خدمات SIM المحمولة
- **إنترنت** - عرض الألياف، DSL، واللاسلكي الثابت
- **IPTV** - عرض خدمات الاشتراك في التلفزيون
- **VoIP** - قائمة خدمات الصوت عبر الإنترنت

تصفية العملاء

ابحث وقم بتصفية حسب العميل:

ابدأ بكتابة اسم العميل لرؤية النتائج المطابقة:

اختر عميلًا لعرض خدماته فقط.

حالة الاستخدام: عرض جميع الخدمات لعميل محدد 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 (تاريخي)
- تاريخ الانتهاء (تاريخي)
- التكلفة الشهرية (رقمي)

اتجاه الفرز:

- انقر مرة واحدة: تصاعدي (A-Z، الأدنى-الأعلى، الأقدم-الأحدث)
- انقر مرتين: تنازلي (Z-A، الأعلى-الأدنى، الأحدث-الأقدم)
- العمود النشط يظهر 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 (المودمات، إلخ، SIM بطاقات)
3. **المعاملات** - الرسوم، الاعتمادات، 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:

إجراءات جماعية

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

تحذير

الحذف الجماعي دائم وسيفعل:

- إلغاء جميع الخدمات المحددة
- 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 لتضييق 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 رسوم
- لا يمكن للعميل استخدام الخدمة
- تظهر بشارة رمادية
- المخزون محدد للإرجاع/إعادة التدوير

معلق

- الخدمة معطلة مؤقتًا
- قد تطبق رسوم أو لا (قابل للتكوين)
- لا يمكن للعميل استخدام الخدمة
- تظهر بشارة برتقالية/صفراء
- الأسباب الشائعة: 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 تم تكوينها)
- **نشط → معلق:** يعطل الخدمة في OCS/السجل بالشبكة ولكن يحتفظ
- **معلق → نشط:** يعيد تفعيل الخدمة، ويستأنف 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 مسبقًا
- **استخدام البيانات** - المستخدم مقابل المخصص (شريط التقدم)
- **استخدام ا��صوت** - 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 متاحًا)

التجديدات التلقائية القادمة

تقدم عرض التجديدات التلقائية القادمة واجهة مركزية لمراقبة وإدارة جميع تجديدات الخدمات المجدولة عبر مؤسستك.

الوصول إلى التجديدات التلقائية

انتقل إلى:

أو مباشرةً:

هذا يعرض جميع الخدمات مع التجديد التلقائي المجدول، مرتبة حسب تاريخ التجديد التالي.

نظرة عامة على التجديدات التلقائية

تظهر 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
- **التكلفة** - تكلفة التجديد (من سعر المنتج بالتجزئة)
- **يتجدد في** - تاريخ ووقت التجديد التالي بصيغة قابلة للقراءة البشرية
- **الحالة** - حالة الخدمة (نشطة، معلقة، إلخ.)
- **الإجراءات** - تجديد الآن أو إزالة التجديد التلقائي

مثال على العرض:

كيفية عمل التجديد التلقائي

يتم جدولة التجديد التلقائي في CGRateS (نظام الفوترة) عند إنشاء خدمة أو تعديلها. النظام:

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 تلقائيًا - **تنفيذ التجديد**
4. ينشئ معاملات، ويحدث تواريخ الخدمة، ويسجل النشاط - **تحديث السجلات**

مصدر البيانات:

تستعلم CGRateS عرض التجديدات التلقائية القادمة مباشرة باستخدام واجهة برمجة التطبيقات `ApierV1.GetScheduledActions` وتغني 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 وظيفة تخصيص وتنفيذها
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 التجديد المبكر
- 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
- تبقى حالة الخدمة دون تغيير
- ستنتهي الخدمة في تاريخ انتهاءها الحالي
- يجب على العميل تجديدها يدويًا أو ستتوقف الخدمة

حالات الاستخدام:

- العميل يقوم بإلغاء الخدمة
- الانتقال إلى عملية التجديد اليدوي
- تغيير خطة الخدمة
- طلب العميل إيقاف الرسوم التلقائية

تحذير

إزالة التجديد التلقائي تعني أن الخدمة ستنتهي ما لم يتم تجديدها يدويًا. لن يتلقى العملاء استمرار الخدمة التلقائي.

فهم العرض

عمود العميل:

- يظهر اسم العميل كرابط قابل للنقر
- يربط إلى صفحة نظرة عامة على العميل
- يظهر "غير متاح" إذا لم تكن الخدمة مرتبطة بعميل (خدمة يتيمة)

عمود الخدمة:

- يظهر اسم الخدمة/UUID
- يربط إلى صفحة نظرة عامة على العميل

- يظهر UUID الخدمة إذا لم يتم تعيين الاسم
- يظهر "غير متاح" إذا لم يمكن العثور على الخدمة في قاعدة البيانات

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:

- التكلفة بالتجزئة من جدول المنتج في CRM
- المبلغ الذي سيتم فرضه عند التجديد
- يظهر "غير متاح" إذا لم يتم العثور على المنتج في قاعدة البيانات

عمود يتجدد في:

- التاريخ والوقت الدقيق للتجديد التالي
- الوقت النسبي القابل للقراءة البشرية (مثل "بعد 3 ساعات"، "بعد يومين")
- مدرك للمنطقة الزمنية (يستخدم منطقة زمنية خادم CGRateS)

عمود الحالة:

- الحالة الحالية للخدمة من قاعدة بيانات CRM
- **نشط** (أخضر) - الخدمة تعمل
- **معلق** (برتقالي) - الخدمة معطلة مؤقتًا
- **غير معروف** (رمادي) - لم يتم العثور على الخدمة في قاعدة البيانات أو لم يتم تعيين الحالة

استكشاف الأخطاء وإصلاحها

الخدمات تظهر "غير متاح" للعميل/الخدمة

- **السبب:** UUID الخدمة في CGRateS لا يتطابق مع service_uuid في قاعدة بيانات CRM
- **الإصلاح:**
  - تحقق من وجود الخدمة في قاعدة البيانات

    - تحقق من أن تنسيق service_uuid يتطابق: `ServiceType_UUID` (مثل، `FixedWireless_75628fa5`)
    - ولكن لا تزال مجدولة في CRM قد تكون الخدمة قد حذفت من CGRateS

التكلفة تظهر "غير متاح"

- **السبب:** معرف المنتج في CRM لا يوجد 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

التاريخ "لا يظهر في" يتجدد

- **السبب:** NextRunTime غير موجود في استجابة CGRateS
- **الإصلاح:**
    - تحقق من تكوين خطة إجراء CGRateS
    - تحقق من أن ActionTiming حيح�� تم تكوينه بشكل
    - تحقق من سجلات CGRateS للأخطاء

"تجديد الآن" يفشل

- **السبب:** أخطاء تخصيص متنوعة
- **استكشاف الأخطاء:**
    - تحقق من نافذة حالة التخصيص للحصول على تفاصيل الأخطاء
    - تحقق من أن خطط Ansible تم تكوينها بشكل صحيح
    - تحقق من الاتصال بـ 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 بالفعل في CGRateS
    - تحقق من اتصال واجهة برمجة التطبيقات CGRateS

تقسيم الصفحات

تُعرض الخدمات في صفحات لأغراض الأداء:

> ◄ السابق 1 [2] 3 4 5 التالي ►
>
> العناصر في الصفحة: [10 ▼]
>
> - 10
> - 25
> - 50
> - 100

التحكم:

- **السابق/التالي** - التنقل بين الصفحات
- **أرقام الصفحات** - الانتقال إلى صفحة معينة
- **العناصر في الصفحة** - ضبط عدد الخدمات المعروضة في الصفحة

نصيحة الأداء: استخدم الفلاتر لتقليل النتائج الإجمالية بدلاً من زيادة العناصر في الصفحة.

شارات الخدمة ومؤشرات الحالة

تساعد المؤشرات المرئية في التعرف على 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 المتكاملة 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:

- بإنشاء حساب الخدمة في CGRateS
- تكوين السمات (معرفات الحساب، ملفات تعريف السرعة، إلخ، MSISDN، IMSI)
- 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 حركة المرور (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:

- **التغييرات السريعة** - تعديل سمة واحدة دون الانتظار للتخصيص
- **الاختبار** - تجربة تكوينات مختلفة
- **دعم العملاء** - حل المشكلات أثناء مكالمات الدعم
- **التكوينات المخصصة** - تطبيق إعدادات خاصة بالخدمة غير المغطاة بواسطة القوالب

تحذير

سير عمل التخصيص القياسي. تأكد من فهمك لتأثير CGRateS تجاوز التغييرات اليدوية لتكوين 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 ثلاث�� أقسام قابلة للطي في CGRateS، عند عرض أو تعديل خدمة تم تخصيصها في أسفل 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`، يكون معرف ملف السمة هو `ATTR_ACCOUNT_Mobile_SIM_c2880638`

إدارة السمات:

1. افتح عرض تعديل الخدمة
2. قم بتوسيع قسم **"سمات CGRateS"**
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 (مثل `*req.Account`, `*req.IMSI`)
- **النوع** - كيفية تعيين القيمة:

    - `*constant` - تعيين قيمة ثابتة
    - `*variable` - التقاط القيمة من حقول الحدث باستخدام RSRParser
    - `*composed` - إضافة القيمة بدلاً 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 للتغييرات 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}`

يكون معرف الفلتر هو `Mobile_SIM_c2880638`، UUID بالنسبة للخدمة ذات **مثال:** `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 (مثل `~*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 الفلتر:

```
{
  "Element": "~*req.Account",
  "Type": "*string",
  "Values": ["Mobile_SIM_474a380a"]
}
```

حالات الاستخدام الشائعة:

- ضمان تطبيق السمات 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 قا��لة للفوترة الأخرى الجارية 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

المعلومات المعروضة للجلسة:

- **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:

- **المعلومات الأساسية:**
  - 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

يتم تطبيق الفلتر (UUID الخدمة). يتم عرض الجلسات التي تتطابق مع حساب هذه الخدمة فقط تلقائيًا `*string:~*req.Account:{service_uuid}`.

واجهة برمجة تطبيقات CGRateS Proxy

تستخدم جميع عمليات CGRateS (السمات، الفلاتر، الجلسات) نقطة نهاية واجهة برمجة التطبيقات OmniCRM:

نقطة النهاية: `POST /crm/ocs/proxy`

الحقول المطلوبة:

- `method` - طريقة واجهة برمجة تطبيقات CGRateS (مثل `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 إلى 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 واجهة برمجة التطبيقات CGRateS غير GET تلقائيًا في تغذية نشاط العميل، بما في ذلك:

- طريقة واجهة برمجة ��لتطبيقات المستدعاة
- 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:** المستخدم يفتقر إلى إذن `cgrates_api_access`
- **الإصلاح:** منح الإذن لدور المستخدم (عادةً ميزة خاصة بالمسؤولين فقط)

السمات أو الفلاتر لا يتم تحميلها

- **السبب:** مشكلة في الاتصال بـ CGRateS أو الملف الشخصي غير موجود
- **الإصلاح:**
  - تحقق من اتصال خادم CGRateS في التكوين
  - تحقق من أن تكوين المستأجر صحيح
  - تحقق من وحدة التحكم في المتصفح بحثًا عن أخطاء واجهة برمجة التطبيقات
  - قد لا يكون الملف الشخصي موجودًا بعد (سيظهر نموذجًا فارغًا)

التغييرات لا تحفظ

- **السبب:** خطأ في التحقق أو خطأ في واجهة برمجة التطبيقات CGRateS
- **الإصلاح:**
  - تحقق من الحقول المطلوبة (Path، Type، Element)
  - تحقق من أن تنسيق JSON صحيح
  - تحقق من سجل النشاط للحصول على تفاصيل الأخطاء
  - راجع سجلات CGRateS

لا تظهر جلسات نشطة

- **السبب:** لا توجد جلسات نشطة حاليًا لهذه الخدمة
- **الإصلاح:**
  - هذا طبيعي إذا كانت الخدمة غير مستخدمة
  - حاول التحديث بعد بدء جلسة (مكالمة، بيانات، إلخ)
  - تحقق من أن UUID الخدمة يتطابق مع الحساب في CGRateS

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

سجل النشاط لا يظهر تغييرات CGRateS

- **السبب:** يتم تسجيل العمليات غير GET (القراءات لا تسجل) فقط
- **الإصلاح:** هذا مصمم - فقط الكتابات/التعديلات تنشئ 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 المنتهية

1. انقر على علامة **"نشط"**
2. رتب حسب **"تاريخ الانتهاء"** (تصاعدي)
3. تظهر الخدمات التي تنتهي قريبًا أولاً
4. اتصل بالعملاء للتجديد

سير العمل 3: تنظيف خدمات الاختبار

1. ابحث عن "اختبار" في مربع البحث
2. راجع النتائج لتأكيد 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. قم بتصفية حسب **نوع الخدمة: محمول**
2. انقر على اسم الخدمة لفتح التفاصيل
3. انقر على علامة **"الاستخدام"**
4. راجع استهلاك البيانات/الصوت/SMS
5. حدد المستخدمين الكثيفين أو التجاوزات

استكشاف الأخطاء وإصلاحها

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:**
  - حاول البحث الجزئي (مثل "0770" بدلاً من الرقم الكامل)
  - استخدم فلتر العميل بدلاً من ذلك
  - تحقق من الأخطاء المطبعية

زر الحذف الجماعي معطل

- **السبب:** لا توجد خدمات مختارة
- **الإصلاح:** تحقق من المربعات بجوار الخدمات التي تريد حذفها

الفرز لا يعمل

- **السبب:** العمود غير قابل للفرز
- **الإصلاح:** فقط الأعمدة التي تحتوي على أيقونات ▲▼ قابلة للفرز

تحميل الصفحة ببطء

- **السبب:** عدد كبير جدًا من الخدمات للعرض
- **الإصلاح:**

    - تطبيق الفلاتر لتقليل مجموعة النتائج
    - تقليل العناصر في الصفحة
    - استخدم البحث لتضييق النتائج

المستندات ذات الصلة

- `csa_add_service` - إضافة خدمات جديدة
- `csa_modify` - تعديل الخدمات وإضافة الإضافات
- `concepts_products_and_services` - مفاهيم المنتج والخدمة
- `basics_customers` - إدارة العملاء

تتبع استخدام الخدمة والرصيد

يوفر نظام تتبع استخدام الخدمة **مراقبة في الوقت الحقيقي** لاستهلاك العملاء للبيانات، الصوت، الرسائل القصيرة، والأرصدة النقدية. تتكامل هذه الميزة مع نظام الشحن عبر (OCS لعرض الاستخدام الحالي، والبدلات المتبقية، ومعلومات انتهاء صلاحية الرصيد لكل من (الإنترنت عبر) العملاء `Self-Care Portal <self_care_portal>`) والموظفين.

نظرة عامة

يمكن لتتبع الاستخدام:

- **عرض الرصيد في الوقت الحقيقي** - عرض الاستخدام الحالي والبدلات المتبقية
- **أنواع رصيد متعددة** - تتبع البيانات، الصوت، الرسائل القصيرة، والأرصدة النقدية في وقت واحد
- **مراقبة انتهاء الصلاحية** - رؤية متى تنتهي صلاحية الأرصدة
- **تحليل ا��رصيد** - عرض تفصيلي لجيوب 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.

عرض الاستخدام

تظهر واجهة الاستخدام بطاقات ملخص وأشرطة تقدم مفصلة لكل نوع من أنواع الرصيد.

بطاقات الملخص

يعرض الصف العلوي بطاقات عرض سريعة لكل نوع من أنواع الرصيد:

معلومات البطاقة:

- **نوع الرصيد** - أيقونة وملصق (بيانات، صوت، رسائل قصيرة، نقدي)
- **المبلغ المتبقي** - الرصيد الحالي بالوحدات المناسبة
- **وقت انتهاء الصلاحية** - الأيام/الساعات حتى انتهاء صلاحية الرصيد
- **زر المزيد من المعلومات** - انقر لتوسيع التحليل التفصيلي

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 (مثل "12.5GB / 20GB")
- **ترميز الألوان**:
  - الأخضر: >50% المتبقية
  - الأصفر: 20-50% المتبقية
  - الأحمر: <20% المتبقية
- **قابل للنقر** - انقر لتوسيع ��لتحليل التفصيلي

أنواع الرصيد

رصيد البيانات

يتتبع استهلاك بيانات الإنترنت.

الوحدات: جيجابايت (GB) أو ميجابايت (MB)

تنسيق العرض:

> التقدم: 12.5GB / 20GB (62%)

السيناريوهات الشائعة:

- 5 - **خطط البيانات المحمولة** بدلات شهرية 5GB، 10GB، 20GB
- **اللاسلكي الثابت** (500GB، 1TB) - غير محدود أو محدد بمبالغ عالية
- **إعادة الشحن** - بيانات إضافية تم شراؤها في منتصف الدورة
- **خدمات الدونجل** - بيانات مسبقة الدفع لأجهزة النقاط الساخنة

جيوب متعددة:

غالبًا ما تحتوي الخدمات على أرصدة بيانات متعددة:

- بدل شهري (تنتهي شهريًا)
- بيانات إضافية (تنتهي بعد فترة الحملة)
- بيانات إعادة الشحن (انتهاء أسرع، تستهلك أولاً)

رصيد الصوت

يتتبع دقائق المكالمات الهاتفية.

الوحدات: دقائق (min)

تنسيق العرض:

> التقدم: 125 min / 500 min (25%)

تتبع مدة المكالمات:

- المكالمات الوارد (إذا كانت مشحونة)
- المكالمات الصادرة
- المكالمات الدولية (جيب منفصل إذا كان ذلك مناسبًا)
- الأرقام المميزة

الحساب:

يتم حساب استخدام الصوت حسب مدة المكالمة بالنانoseconds داخليًا، ويتم تحويلها إلى دقائق 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 الرصيد المسبق الدفع أو رصيد الحساب.

الوحدات: عملة (£، $، €، إلخ.)

تنسيق العرض:

> التقدم: £15.50 / £20.00 (77%)

الاستخدام:

- تستخدم الحسابات المسبقة الدفع الرصيد النقدي لدفع مقابل الاستخدام
- يتم تخفيض الرصيد مع استخدام العميل للخدمات
- يمكن إعادة شحنه عبر الدفع أو القسيمة
- قد تنتهي صلاحيته إذا لم يتم استخدامه خلال فترة الصلاحية

تحليل الرصيد التفصيلي

انقر على **"المزيد من المعلومات"** على أي بطاقة أو انقر على شريط التقدم لتوسيع التحليل التفصيلي.

عرض موسع:

 بدل شهري 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

الإجمالي المتبقي: 17.5 GB

حقول جيب الرصيد:

- **الاسم** - معرف لجيب الرصيد/**ID**
- **المتبقي** - المبلغ المتبقي في هذه الجيب المحدد
- **المستخدم** - المبلغ المستهلك من هذه الجيب
- **تاريخ انتهاء الصلاحية** - متى تنتهي صلاحية هذا الرصيد
- **الوزن** - ترتيب الأولوية (يتم استهلاك الوزن الأعلى أولاً)

نظام الوزن

تحدد ترتيب الاستهلاك **وزن** تمتلك الأرصدة قيمة:

- **وزن أعلى = يتم استهلاكه أولاً**
- **وزن أقل = يتم استهلاكه أخيرًا**

أمثلة على الأوزان:

- بيانات إعادة الشحن: الوزن 30 (يتم استهلاكه أولاً، انتهاء أقصر)
- بيانات إضافية: الوزن 20 (يتم استهلاكها ثانيًا)
- بدل شهري: الوزن 10 (يتم استهلاكه آخرًا، انتهاء أطول)

هذا يضمن أن الأرصدة التي تنتهي صلاحيتها تُستخدم قبل الأرصدة الأطول عمراً.

للحصول على تفاصيل كاملة حول قواعد استهلاك الرصيد، واستراتيجيات الوزن، وكيفية إعطاء الأولوية للأرصدة في CGRateS، انظر إجراءات CGRateS وسلوكيات إعادة الشحن.

التحديثات في الوقت الحقيقي

تتجدد بيانات الاستخدام تلقائيًا كل **3 ثوانٍ** عبر الاستطلاع.

ما الذي يتم تحديثه:

- المبالغ الحالية للرصيد
- أشرطة تقدم الاستخدام
- مؤقتات انتهاء الصلاحية
- تفاصيل الجيب الفردية

تجربة المستخدم:

- لا حاجة لإعادة تحميل الصفحة
- تحديثات سلسة دون وميض
- تحميل تغطي أثناء التحديث
- شارة الحالة تظهر حالة الخدمة الحالية

حالات الاستخدام:

- مراقبة استخدام العميل أثناء المكالمة
- مشاهدة انخفاض الرصيد في الوقت الحقيقي أثناء استخدام العميل للخدمة
- التحقق من إعادة الشحن مباشرة بعد الشراء

الاستخدام في أنواع الخدمة المختلفة

الخدمات المحمولة

عرض جميع أنواع الرصيد الأربعة:

- البيانات (GB)
- الصوت (دقائق)
- الرسائل القصيرة (رسائل)
- النقدي (عملة)

مثال:

> البيانات: 12.5GB المتبقية الصوت: 125 min المتبقية الرسائل القصيرة: 45 msgs المتبقية
> النقدي: £15.50 المتبقية

اللاسلكي الثابت / الإنترنت

عادةً ما يظهر فقط:

- البيانات (GB أو TB)
- النقدي (إذا كان مسبق الدفع)

مثال:

> البيانات: 450GB / 500GB (رصيد مسبق الدفع) المتبقية النقدي: £45.00

خدمات النقاط ال��اخنة / الدونجل

يظهر تتبع البيانات الخاص بالدونجل:

- البيانات (المستهلكة مقابل المسبقة الدفع)
- النقدي (رصيد مسبق الدفع)

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 مهيأة في OCS
  - تحقق من اتصال API OCS
  - مراجعة تعيين UUID الخدمة

الاستخدام لا يتحدث

- غير متاح OCS **السبب:** توقف الاستطلاع أو
- **الإصلاح:**
  - قم بتحديث الصفحة
  - تحقق من وحدة تحكم المتصفح للبحث عن الأخطاء
  - متصل API OCS تحقق من أن

الأرصدة تظهر مبالغ غير صحيحة

- أو مشكلة في التخزين المؤقت OCS **السبب:** عدم تطابق بيانات
- **الإصلاح:**
  - فرض تحديث رصيد OCS
  - تحقق من المعاملات المعلقة
  - من تكوين ��تحق OCS

تاريخ انتهاء الصلاحية مفقود

- **السبب:** الرصيد ليس له تاريخ انتهاء محدد
- **الإصلاح:**
  - بعض الأرصدة محددة بأنها لا تنتهي أبدًا (صلاحية غير محدودة)
  - تحقق من تكوين الرصيد في OCS

الأرصدة المتعددة مربكة

- **السبب:** تمت إضافة عدة إعادة شحنات أو بيانات إضافية
- **الإصلاح:**
  - استخدم عرض التحليل التفصيلي لرؤية جميع الجيوب
  - قم بالفرز حسب الوزن لرؤية ترتيب الاستهلاك
  - راجع تواريخ انتهاء الصلاحية الفردية

التكامل مع OCS/CGRateS

تأتي بيانات الاستخدام من **OCS (نظام الشحن عبر الإنترنت)**، عادةً CGRateS. للحصول على معلومات شاملة حول كيفية عمل الأرصدة، بما في ذلك أنواع الأرصدة، وقواعد الاستهلاك، وسلوك إعادة الشحن، وتكوين الإجراءات، انظر إجراءات وسلوكيات إعادة الشحن CGRateS.

تدفق البيانات:

1. يفتح المستخدم عرض الاستخدام
2. يس��دعي OmniCRM `GET /crm/service/{service_id}`
3. يستعلم API OCS عبر UUID الخدمة
4. يعيد 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. يقوم 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:

يعيد OCS بيانات الرصيد مع بادئات الأنواع التي تتطابق مع عرض واجهة المستخدم:

- `*data` → بطاقة **DATA** (استخدام الإنترنت)
- `*voice` → بطاقة **VOICE** (دقائق المكالمات)
- `*sms` → بطاقة **SMS** (رسائل نصية)
- `*monetary` → بطاقة **MONETARY** (رصيد مسبق الدفع)

يمكن أن يحتوي كل نوع من أنواع الرصيد على جيوب متعددة (مثل بدل شهري + بيانات إضافية + بيانات إعادة الشحن)، جميعها معروضة في عرض التحليل التفصيلي.

تحويلات قيمة الرصيد:

- **البيانات:** بايت → GB (قسم على 1024^3)
- **الصوت:** نانوثانية → دقائق (قسم على 60×10^9)
- **الرسائل القصيرة:** عدد (لا تحويل)
- **نقدي:** سنتات → عملة (قسم على 100)

التجديد التلقائي وخطط الإجراءات

تحتوي الخدمات المفعلة للتجديد التلقائي على **خطط إجراءات** مجدولة في OCS. للحصول على معلومات تفصيلية حول كيفية عمل الإجراءات، وتلاعب الرصيد، وتكوين خطط الإجراءات، انظر إجراءات CGRateS وسلوكيات إعادة الشحن.

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 البدلات الشهرية
- تطبيق رسوم متكررة
- انتهاء صلاحية الأرصدة القديمة

كيف يعمل التجديد التلقائي:

1. **تهيئة الخدمة:**
   - عند إنشاء الخدمة مع `auto_renew = true`
   - يقوم كتاب تهيئة الخدمة بإنشاء خطة إجراء في OCS
   - يتم تكوين خطة الإجراء لتعمل 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 (مثل، شهريًا في اليوم الأول في 00:00 UTC)
   - **المبلغ** - مقدار الرصيد الذي يجب إضافته
3. **التنفيذ التلقائي:**
   - ينفذ OCS خطة الإجراء في الوقت المحدد
   - يضيف رصيدًا إلى الحساب (مثل، 20 GBصوت، 500 دقيقة، 500 بيانات)
   - يحدد تاريخ انتهاء صلاحية الرصيد الجديد (مثل، 30 يومًا)
   - يتم تحصيل رسوم من العميل عبر طريقة الدفع المسجلة
4. **عرض خطط الإجراءات:**
   - انتقل OCS إلى تفاصيل الخدمة في عرض
   - يتم سرد خطط الإجراءات مع وقت التنفيذ التالي
   - يظهر: اسم الخطة، تاريخ التشغيل التالي، تفاصيل الإجراء

مثال على خطة إجراء:

```
{
  "ActionPlanId":
"ProductID_MonthlyPlan__ProductName_20GB_Mobile__ActionPlan_Monthly_F
  "NextExecTime": "2025-02-01T00:00:00+00:00",
  "ActionName_hr": "التجديد الشهري",
  "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 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

لفريق الدعم:

- "تحقق من الاستخدام قبل الرد على مكالمات "لماذا خدمتي بطيئة؟
- تحقق من الرصيد بعد إعادة الشحن لتأكيد النجاح
- استخدم التحليل التفصيلي لتحديد الجيوب المنتهية
- راقب العملاء ذوي الاستخدام العالي لمنع التجاوزات

للعملاء (الرعاية الذاتية):

- تحقق من الاستخدام بانتظام لتجنب نفاد الرصيد
- أعد الشحن قبل انتهاء صلاحية الرصيد
- فهم نظام الوزن لمعرفة أي رصيد يتم استهلاكه أولاً
- اتصل بالدعم إذا بدا أن الاستخدام غير صحيح

للمسؤولين:

- تكوين انتهاء صلاحية الرصيد المناسبة
- تعيين قيم الوزن لإعطاء الأولوية للأرصدة التي تنتهي صلاحيتها
- مراقبة اتصال OCS للتقارير الدقيقة
- مراجعة تكوينات الرصيد لمطابقة عروض المنتجات

الوثائق ذات الصلة

- إجراءات CGRateS وسلوكيات إعادة الشحن - دليل كامل لأنواع الرصيد، قواعد الاستهلاك، الأوزان، سلوك إعادة الشحن، وتكوين 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 البدلات
- `concepts_products_and_services` - تكوين المنتج

خدمة العملاء

انتحال الهوية للدعم وحل المشكلات

يسمح الانتحال للموظفين المخولين بتسجيل الدخول مؤقتًا كعميل آخر لحل المشكلات، والتحقق من التكوينات، أو رؤية بالضبط ما يختبره العميل. هذه الميزة ضرورية لدعم العملاء ولكنها تتطلب أذونات مناسبة وتخضع لتدقيق كامل.

عند انتحال هوية عميل، يصل الموظفون إلى `Self-Care Portal <self_care_portal>` تمامًا كما يراه العميل، مما يسمح بحل المشكلات والدعم بدقة.

انظر أيضًا: `RBAC <rbac>` لتكوين الأذونات، `Customers <basics_customers>` لإدارة العملاء، `Self-Care Portal <self_care_portal>` لميزات بوابة العملاء.

الغرض

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** — عرض الميزات من منظور العميل
4. **الدعم** — مساعدة العملاء في التنقل في النظام دون الحاجة إلى مشاركة الشاشة
5. **سجل التدقيق** — يتم تسجيل جميع جلسات الانتحال لأغراض الأمان والامتثال

الأذونات المطلوبة

للانتحال كعملاء، 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. ابحث عن العميل في نظام إدارة علاقات العملاء — **انتقل إلى العميل**
2. عرض قائمة جهات الاتصال الخاصة بالعميل — **حدد جهات الاتصال**
3. يظهر زر بجوار كل جهة اتصال لديها حساب مستخدم — **انقر على "تسجيل الدخول كعميل"**
4. قد يطلب النظام تأكيد — **تأكيد الانتحال**
5. أنت الآن مسجل الدخول كذاك المستخدم — **بدء الجلسة**

عبر API:

بدء جلسة الانتحال:

نقطة النهاية: `POST /auth/impersonate`

الإذن المطلوب: `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"
}
```

تكون الرموز المستردة لجلسة المستخدم المنتحل.

ماذا يحدث أثناء الانتحال

عند انتحال هوية مستخدم:

- **السياق الكامل** — ترى النظام تمامًا كما يراه المستخدم المستهدف:
  - لوحة التحكم والتنقل الخاصة بهم
  - بيانات العميل الخاصة بهم (إذا كان مستخدم عميل)
  - أذوناتهم ووسائل التحكم في الوصول
  - خدماتهم، والفواتير، والاستخدام
- **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** — يتم تسجيل 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** — انتقل إلى قائمة المستخدم 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

يتم تسجيل جميع جلسات الانتحال في جدول `ImpersonationLog` مع:

- **impersonating_user_id** — عضو الفريق الذي قام بالانتحال
- **target_user_id** — العميل أو المستخدم الذي تم انتحاله
- **impersonation_start** — طابع زمني للبداية
- **impersonation_end** — طابع زمني للنهاية (عندما تم إيقاف الجلسة)
- **impersonation_duration** — المدة بالثواني

هذا يوفر المساءلة الكاملة لجميع جلسات الانتحال ويسمح بـ:

- **تدقيق الأمان** — مراجعة من انتحل من ومتى
- **تقارير الامتثال**— إثبات الاستخدام الصحيح للوصول المرتف
- **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 بالانتحال)
- المستخدم المستهدف (من تم انتحاله)
- نطاق التاريخ
- المدة

أفضل الممارسات

1. **تقليل المدة** — انتحل فقط طالما كان ذلك ضروريًا لحل المشكلة
2. **توثيق الغرض** — لاحظ لماذا كان الانتحال مطلوبًا 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 الانتحال عبر حسابهم
5. **مراجعة السجلات** — تدقيق سجلات الانتحال بانتظام للبحث عن أنماط غير عادية
6. **تقييد الأذونات** — منح `can_impersonate` فقط لموظفي الدعم الذين يحتاجون إليها
7. **التدريب** — التأكد من فهم الموظفين للمسؤولية وآثار التدقيق

اعتبارات الأمان

- **الوصول الكامل** — يمنح الانتحال وصولًا كاملاً إلى حساب المستخدم المستهدف
- **لا حاجة لكلمة مرور** — يتجاوز الانتحال المصادقة (مبني على الأذونات فقط)

- **الإجراءات المسجلة** — تُنسب جميع الإجراءات أثناء الانتحال إلى المستخدم المستهدف في سجلات التطبيق (لكن سجلات التدقيق تظهر الفاعل الحقيقي)
- **عزل الجلسة** — ينشئ الانتحال جلسة جديدة؛ لا يؤثر على الجلسات النشطة للمستخدم المستهدف
- **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 المسؤولين الآخرين
- **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 يمكن تعطيل أو إخفاء تسجيل الانتحال

حل المشكلات الشائعة

المشكلة: زر "تسجيل الدخول كعميل" لا يظهر

- 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 الأذونات

- 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: ا��إجراءات لا تسجل بشكل صحيح

- الحل: تحقق من أن جلسة الانتحال نشطة (تحقق من اللافتة)
- الحل: مراجعة سجلات التدقيق - يتم تسجيل الإجراءات مع كلا معرفي المستخدم

تعريف المنتجات وتكوين CGRateS

تشرح هذه الدليل كيفية تعريف المنتجات في OmniCRM جنبًا إلى جنب مع تكوين الفوترة المرتبط بـ CGRateS.

عادةً ما يتم تعريف المنتجات وإجراءات Python الخاصة بها معًا في نصوص CGRateS التي تقوم بتكوين كل من CRM و OCS (نظام الشحن عبر الإنترنت).

نظرة عامة

يتضمن تعريف منتج كامل تكوينًا في **نظامين**:

1. **CGRateS/OCS** - تعريف سلوك الفوترة (الإجراءات، الأسعار، الوجهات)
2. **OmniCRM** - تعريف بيانات المنتج الوصفية (السعر، الاسم، الميزات)

هذه الأنظمة مرتبطة بشكل غير مباشر عن عمد، مما يسمح لك بالتفاعل مع CGRateS عبر ألعاب Ansible التي تحددها.

CGRateS مهم: قدرات

CGRateS هو محرك تقييم قوي ومرن للغاية. يوضح هذا الدليل **نمط تكوين شائع واحد** لمنتجات OmniCRM، لكن CGRateS يمكنه القيام بالكثير:

- **سيناريوهات تقييم متقدمة**: أسعار حسب الوقت، مستويات العملاء، توجيه المكالمات بناءً على الجودة، فوترة مفاجئة
- **منطق تجوال معقد**/SMS الصادرة مقابل الواردة عند التجوال: أسعار مختلفة لمكالمات التجوال، تسعير محدد للشبكة
- **دعم البروتوكولات القديمة**: CAMEL 2 لتجوال 2G/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 نظام توفير OmniCRM. للحصول على تكوينات CGRateS المتقدمة، راجع وثائق CGRateS أو انظر إجراءات CGRateS لأساليب التوازن واستراتيجيات التقييم وسلوكيات الشحن.

سير عمل تعريف المنتج الكامل

الخطوة 1: المصادقة على واجهة برمجة تطبيقات CRM

```
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 م��زون 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"
})

# التحقق مما إذا كان قالب المخزون موجودًا، وإنشاءه 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
    )

    # التحقق 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 تواجه العميل
- **allow_dropdown_staff** - عرض في قوائم المنسدلة للموظفين
- **allow_dropdown_customer** - عرض في قوائم المنسدلة للعملاء
- **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

الوجهات تحدد أين يمكن استخدام الأرصدة. يجب تعريفها قبل إنشاء الإجراءات التي تشير إليها.

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 الصوت/SMS** → استخدم الوجهات الجغرافية (الرقم الذي يتم الاتصال به)
- **أرصدة البيانات** → استخدم وجهات PLMN (الشبكة التي يتصل بها العميل)

لأمثلة كاملة على تعريف الوجهات بما في ذلك:

- الوجهات الجغرافية (محمول، خط ثابت، مجاني، دولي)
- وجهات PLMN (على الشبكة، تجوال، مناطق)
- قواعد تنسيق PLMN (`mccXXX.mncYYY`)
- متى تستخدم كل نهج

انظر: إجراءات CGRateS - تعريف الوجهات

الخطوة 5: تعريف إجراءات CGRateS

الإجراءات هي الآلية لإضافة أرصدة إلى حسابات العملاء. قبل أن تتمكن من ربط منتج CRM بإج��اء، يجب تعريف الإجراء في CGRateS.

حرج: يتم تعريف الإجراءات أثناء التكوين الأولي للنظام (ليس أثناء التوفير). تحدد ما الأرصدة التي يجب إضافتها، أوقات انتهاء الصلاحية، سلوك التراجع، إلخ.

انظر إجراءات CGRateS للحصول على الوثائق الكاملة حول وسلوكيات الشحن:

- كيفية تعريف الإجراءات باستخدام Python (`OCS_Obj.SendData`)
- فهم `*topup` مقابل `*topup_reset` (سلوك التراجع)
- حسابات الوحدات (بايت للبيانات، نانوثانية للصوت، عدد لـ SMS)
- مرجع الحقول (BalanceId، BalanceType، DestinationIDs، Units، ExpiryTime، Weight)
- أمثلة عمل كاملة (خطط متعددة الأرصدة، إضافات البيانات، إضافات الصوت)
- قواعد وأولوية استهلاك الأرصدة

مرجع سريع للتراجع:

- **`*topup`** - يضيف إلى الرصيد الحالي (يمكّن التراجع). لـ: إضافات مسبقة الدفع، حزم البيانات
- **`*topup_reset`** - يستبدل الرصيد الحالي (لا تراجع). 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 تنشيطها 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 بيانات في أستراليا، 3000 دقيقة 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 بيانات في أستراليا', '3000 دقيقة في أستراليا', '6 SMS في أستراليا', '3000 بيانات تجوال GB']",
    "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": "20GB بيانات في أستراليا",
    "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": "['20GB بيانات في أستراليا']",
    "wholesale_cost": 22,
    "wholesale_setup_cost": 0,
    "auto_renew": "False",
    "customer_can_purchase": True
})
```

أوصاف 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) - كتاب اللعب الخاص بـ 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 يوجد عقد، 30 = شهري)
- **retail_cost** (مطلوب) - التكلفة الشهرية/الدورية التي تواجه العميل
- **retail_setup_cost** - رسوم إعداد لمرة واحدة يتم تحصيلها من العميل

- **product_slug** (مطلوب) - **معرف فريد يرتبط بإجراء CGRateS** (القاعدة: `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 Python من العناصر المطلوبة في المخزون
- **icon** - فئة أيقونة Font Awesome
- **features_list** - سلسلة قائمة Python من ميزات الرصاص
- **wholesale_cost** - تكلفتك الدورية
- **wholesale_setup_cost** - تكلفتك لمرة واحدة
- **auto_renew** - سلسلة "True" أو "False" للتجديد التلقائي
- **customer_can_purchase** - يمكن للعميل الشراء عبر بوابة الخدمة الذاتية

انظر 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']}")

    # التحقق 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 للمنتج الموجود
        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 **قاعدة التسمية** هو:

```
# 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"
```

كيفية تنفيذ كتب اللعب للإجراءات

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** CGRateS وإلا ستفشل التنفيذ مع "الإجراء غير موجود" خطأ.

انظر أيضًا: دليل كتب Ansible

مثال كامل: تعريف منتج جديد

دعونا نتناول إنشاء منتج جديد كامل "حزمة بيانات 50GB":

1. تعريف الإجراء في CGRateS

انظر إلى إضافة بيانات بسيطة المثال 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"` (يمكّن 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": "50 حزمة بيانات GB",
    "comment": "50GB يومًا 30 من البيانات عالية السرعة صالحة لمدة",
    "provisioning_json_vars": "",
    "terms": "تنتهي صلاحية البيانات بعد 30 يومًا. يتم نقل البيانات غير المستخدمة إذا تم إعادة 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 التراجع ','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. يشتري العميل "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. يحصل العميل على رصيد 50GB

انظر أيضًا: الشحن والمدفوعات من كتب اللعب

أنماط المنتج الشائعة

تظهر هذه الأنماط تكوينات المنتج النموذجية في CRM وكيف ترتبط بإجراءات CGRateS. للحصول على تعريفات الإجراءات المقابلة، انظر 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": "خطة شهرية متميزة",
    "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 متعددة من الأرصدة. انظر مثال 1: خطة شهرية متعددة الأرصدة (Python).

الإجراء: يستخدم `*topup_reset`

النمط 2: إضافة بيانات بسيطة (تراجع)

حالة الاستخدام: شراء بيانات لمرة واحدة تتراجع إذا تم إعادة شحنها مرة أخرى

منتج CRM:

```
{
    "product_slug": "10gb-addon",  # يرتبط بـ Action_10gb-addon
    "product_name": "إضافة بيانات 10GB",
    "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 صلاحية 30 يومًا', 'بيانات مع تراجع']"
}
```

الإجراء: يستخدم `*topup` لسلوك التراجع. انظر إجراءات CGRateS - التراجع.

النمط 3: خطة شهرية ثابتة (بدون ت��اجع)

حالة الاستخدام: خطة شهرية تعيد ضبط نفسها دائمًا إلى نفس المبلغ بالضبط

منتج CRM:

```
{
    "product_slug": "fixed-50gb-monthly",  # يرتبط بـ Action_fixed-50gb-monthly
    "product_name": "خطة شهرية ثابتة 50GB",
    "category": "addon",
    "retail_cost": 35.00,
    "auto_renew": "True",
    "contract_days": 30,
    "provisioning_play": "play_topup_charge_then_action",
    "features_list": "['50GB بدون تراجع', 'تجدد شهريًا', 'بيانات']"
}
```

سلوك إعادة - CGRateS لمنع التراجع. انظر إجراءات `*topup_reset` **الإجراء:** يستخدم الضبط.

متقدم: الشواحن وخطط التقييم

سيناريوهات CGRateS **مهم:** تغطي هذه القسم التقييم الأساسي للدفع مقابل الاستخدام. يدعم تقييم معقدة للغاية تشمل:

- الصادرة مقابل الواردة عند SMS/**أسعار تجوال مختلفة**: أسعار مختلفة لمكالمات التجوال
- **شحن محدد بالبروتوكول**: CAMEL 2 لـ3G/2G (SMS الصادر والوارد، USSD)، Diameter 4 لـ5G
- **تسعير يعتمد على السياق**: أسعار تعتمد على الوقت، الموقع، شريحة العملاء، جودة الشبكة
- **تقييم متعدد الأبعاد**: دمج نوع الشبكة، حالة التجوال، الوجهة، ووقت اليوم

تظهر الأمثلة أدناه تكوينات PAYG بسيطة ومسطحة مناسبة لمعظم نشرات OmniCRM.

للفوترة بالدفع عند الاستخدام (حيث تكلف الاستخدامات التي تتجاوز الأرصدة المضمنة المال)، تحتاج إلى تعريف خطط تقييم تحدد المبالغ بالدولار التي يجب تحصيلها.

كيفية عمل التقييم

عندما يستخدم العميل خدمة (صوت، بيانات، SMS):

1. يتحقق CGRateS من الأرصدة الوحدوية (مثل GB البيانات المضمنة)
2. إذا كانت الرصيد الوحدوي موجودًا، 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 التكلفة المحسوبة من الرصيد النقدي للعميل

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 توجد رسوم اتصال
            "Rate": 0.10,             # 0.10 دولار لكل وحدة
            "RateUnit": "1048576",    # 1 ميغابايت (1024 * 1024 بايت)
            "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 تفعله هذه:

- عندما لا يكون لدى العميل رصيد بيانات (أو ينفد)، تكلف استخدام البيانات **0.10 دولار لكل ميغابايت**
- يتم خصم الرسوم من رصيده النقدي (`*monetary`)
- يتم الفوترة بزيادات 1 ميغ��بايت (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",       # تدوير 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 (اخت��اري)

تطبق الشواحن التقييم على أنواع معينة من الأحداث. بالنسبة لمعظم حالات الاستخدام، فإن الشاحن الافتراضي كافٍ:

```
OCS_Obj.SendData({
    "method": "APIerSv1.SetChargerProfile",
    "params": [{
        "Tenant": tenant,
        "ID": "Charger_Default",
        "FilterIDs": [],
        "AttributeIDs": ["*constant:*req.RequestType:*none"],
        "Weight": 999
    }]
})
```

انظر أيضًا:

- إجراءات CGRateS (أساليب إدارة الأرصدة (وحدوية، نقدية، هجينة) - وسلوكيات الشحن وتكوين التقييم
- وجهات CGRateS - كيفية تعريف الوجهات الجغرافية وPLMN

أفضل الممارسات

1. استخدم قواعد تسمية متسقة

```
# 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 الإجراءات 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 = يتم استهلاكه أولاً)
"BalanceWeight": 1200  # أرصدة متميزة/محلية
"BalanceWeight": 1100  # أرصدة التجوال
"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

الخطأ: "الإجراء غير موجود"

الأعراض: يفشل كتاب اللعب مع "الإجراء غير موجود" أو "لا يوجد ActionsId"

السبب: الإجراء غير معرف في CGRateS، أو عدم تطابق في التسمية

الحل:

1. تحقق من وجود الإجراء: استعلام CGRateS عن الإجراء
2. تأكد من التسمية: تحقق من `ActionsId = "Action_" + product_slug`
3. تعريف الإجراء إذا كان مفقودًا

"الخطأ: "الوجهة غير موجودة

الأعراض: يتم تنفيذ الإجراء ولكن الأرصدة لا تُنشأ

السبب: DestinationIDs تشير إلى وجهة غير موجودة

حل��ل:

1. أولاً CGRateS تعريف الوجهة في
2. استخدم `*any` للوجهات العالمية
3. تحقق من تهجئة معرفات الوجهة

تم إنشاء المنتج ولكن لم يتم إضافة أرصدة

الأعراض: يتم توفير المنتج بنجاح ولكن العميل ليس لديه رصيد

السبب: الإجراء موجود ولكن ليس لديه عمليات إضافة أرصدة

الحل:

1. تحقق من أن الإجراء يحتوي على عمليات `*topup` أو `*topup_reset`
2. تحقق من أن قيمة الوحدات صحيحة (ليست 0)
3. تحقق من أن وقت انتهاء الصلاحية لم ينته بعد

الأرصدة لا تتراجع

الأعراض: استخدام التراجع 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

انظر أيضًا: إجراءات CGRateS وسلوكيات الشحن

ملخص

يوضح هذا 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، ما بعد الدفع/مسبقة الدفع مع خطط.

تذكر: CGRateS قادر على سيناريوهات أكثر تعقيدًا بكثير مما تم تغطيته هنا. إذا كنت بحاجة إلى ميزات متقدمة مثل:

- تسعير MO/MT وصوت SMS مختلف لمكالمات أثناء التجوال
- شحن قائم على CAMEL 2 للأجهزة القديمة 2G/3G
- تغييرات في الأسعار حسب الوقت أو الموسم
- تسعير يعتمد على شريحة العملاء (برونزي/فضي/ذهبي)
- شحن يعتمد على جودة الشبكة أو QoS
- توجيه أقل تكلفة مع موردين متعددين

استشر وثائق CGRateS مباشرة واعمَل على تكوين خطة التقييم بشكل مستقل عن تعريفات منتجات CRM...

الوثائق ذات الصلة

- إجراءات CGRateS وسلوكيات الشحن - دليل مفصل حول أنواع الإجراءات وسلوك الأرصدة
- دليل كتب Ansible - كيفية تنفيذ ��تب 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 الرسمية - مرجع الكامل CGRateS

نظام بث الخلايا

تمكين مشغلي الشبكات المحمولة من إرسال تنبيهات OmniCRM يمكن لنظام بث الخلايا في الطوارئ والتحذيرات العامة إلى الأجهزة المحمولة داخل مناطق جغرافية محددة. يُعتبر بث الخلايا وتحذيرات الطقس، وتنبيهات تسونامي، AMBER، ميزة حيوية للسلامة العامة تُستخدم لتنبيهات وغيرها من الإشعارات الطارئة.

القياسية، ستقوم رسائل بث الخلايا بتنبيه الهواتف SMS **الميزة الرئيسية:** على عكس رسائل صوتيًا حتى لو كانت في وضع الصامت، أو خارج الرصيد، أو في التجوال. نظرًا لأن هذه رسالة بث، فمن الممكن إرسال تنبيه إلى كل فرد من السكان الذين يحملون هاتفًا محمولًا في غضون ثوانٍ.

رابط تحذير أومني تاتش (OWL)

حلاً شاملاً لمتخصصي إدارة الكوارث ومشغلي (OWL) توفر منصة رابط تحذير ومني تاتش الشبكات المحمولة:

- **كيان بث الخلايا (CBE)** - تطبيق ويب آمن للمستخدمين المصرح لهم لإنشاء وبث رسائل تحذير الطوارئ
- **مركز بث الخلايا (CBC)** - مكون تكامل الشبكة المتوافق مع المعايير الذي يتصل لتوزيع الرسائل (2G/3G/4G/5G) بالشبكات الخلوية

للاستخدام على أي جهاز يحتوي على متصفح ويب OWL تم تصميم مثل أجهزة الكمبيوتر، واللابتوبات، والأجهزة اللوحية أو (Chrome/Firefox/Safari/Edge)، الهواتف المحمولة.

نظرة عامة

للمشغلين بـ (PWS المعروف أيضًا باسم نظام التحذير العام أو) يسمح بث الخلايا:

- **إرسال تنبيهات الطوارئ** - توزيع معلومات السلامة الحيوية على جميع الأجهزة في منطقة معينة
- **استهداف المناطق الجغرافية** - البث إلى مناطق تتبع محددة أو خلايا شبكة عن طريق التحكم في أي أبراج خلوية تبث الرسائل
- **دعم لغات متعددة** - توفير رسائل التحذير بعدة لغات في وقت واحد (حتى 500 حرف لكل لغة)

- **إدارة دورة حياة التحذير** - إنشاء، تحديث، الموافقة، مراقبة وحذف الرسائل المذاعة
- **التكامل مع الأنظمة الخارجية** (CBC) عبر - الاتصال بالبنية التحتية لمركز بث الخلايا واجهات شبكة خلوية متعددة
- **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

يتم استخدام بث الخلايا لـ:

- **تحذيرات الطوارئ** - الكوارث الطبيعية (الزلازل، التسونامي�� الفيضانات، الحرائق)
- **تنبيهات AMBER** - إشعارات اختطاف الأطفال
- **تنبيهات الطقس** - تحذيرات الطقس القاسي، تنبيهات الأعاصير
- **السلامة العامة** - تهديدات إرهابية، انسكابات كيميائية، إخلاءات
- **رسائل الاختبار** - اختبار النظام وحملات التوعية العامة (لا تنبه رسائل الاختبار الشهرية المستخدمين ولكن سيتم تلقيها)

يكون الاستهداف الجغرافي غالبًا ذا أهمية قصوى في حالات الطوارئ. على سبيل المثال، يجب ألا تُرسل تنبيهات تسونامي التي تنصح القريبين من المناطق الساحلية بالبحث عن مناطق مرتفعة 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 إجراء سريع 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 للبث إليها

3. **موافقة الرسالة** - تفويض/تحقق من هوية المشغل وموافقة الشخص الثاني (إذا لزم الأمر)
4. **مراجعة الرسالة** - تأكيد نهائي 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: يحتوي كل نموذج رسالة متاح في نظام OWL بالفعل على معرف الرسالة المناسب (MI) مضمن، لذا لا يحتاج مستخدمو النظام إلى تحديد ذلك يدويًا عند إنشاء رسالة تحذير.

نص الرسالة
نص الجسم محدود بـ 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 الحد الأقصى البالغ 500 حرف ينطبق على الرسالة الإجمالية بما في ذلك جميع اللغات.

قوالب الرسائل
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 والتكرارات
تمتلك الرسائل الطارئة عمرًا محددًا تكون فيه ذات صلة. عند تعريف الرسالة:

- **تنتهي (بالدقائق)** - كم من الوقت ستستمر الرسالة في البث

- **تكرارات الرسالة** - عدد المرات التي سيتم إعادة بثها

ستظهر الرسالة على الهاتف للمستخدم مرة واحدة فقط. ومع ذلك، ستستمر مواقع الخلايا في نقل الرسائل حتى يتم الوصول إلى وقت انتهاء الصلاحية لضمان تلقي الأشخاص الذين يدخلون منطقة التغطية من الخارج الرسالة.

المرحلة 2: الاستهداف

يتم إرسال رسائل بث الخلايا على مستوى برج الخلية، ويمكن تحديد النطاق الجغرافي عن طريق اختيار أي الأبراج تبث الرسالة.

الاستهداف الاختياري
هذه الخطوة اختيارية. عدم إدخال أي معلومات استهداف يعني أن جميع أبراج الخلايا ستنقل رسالة التحذير الطارئة.

المناطق المستهدفة المحددة مسبقًا
قاعدة بيانات لجميع أبراج الخلايا ويمكنه تحديد المناطق المستهدفة على OWL متلك نظام��
الخريطة. يمكن استهداف المناطق باستخدام مناطق محددة مسبقًا (تم تحديدها مسبقًا للاختيار السريع) أو عن طريق 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 متوافقة أخرى 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 الاستجابة السريعة 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 تفصيلية حول إعداد 2FA، انظر `المصادقة الثنائية` `<2fa>`.

إذا قمت بتغيير الأجهزة المحمولة أو توقف التطبيق عن المزامنة مع حساب OWL الخاص بك، فاتصل بمدير النظام للحصول على المساعدة. يمكن للمسؤولين إعادة تعيين 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 خطر سوء الاستخدام.

أدوار المستخدمين الدقيقة
يمكن تكوين أدوار المستخدمين الفردية لـ:

- السماح فقط لمستخدمين معينين بإرسال رسائل محددة مسبقًا
- تقييد استهداف الرسائل لمناطق معينة
- طلب سير عمل موافقة إضافية

المرحلة 4: المراجعة

بمجرد اكتمال مراحل تعريف الرسالة، والاستهداف، والموافقة، يجب على المشغل مراجعة الرسالة قبل البث النهائي. بمجرد أن يكونوا راضين عن تفاصيل الرسالة، يمكنهم 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.

التعليقات الشبكية
ترجع 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 الموقف
- معدلة بمحتوى نص الرسالة الجديد
- مسترجعة/متوقفة 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 الزلازل، 4371 لتحذير التسونامي ETWS)

- **الفئة** - فئة التحذير (عادية، طارئة، عالية، قصوى)
- **فترة التكرار** - ثوانٍ بين تكرارات البث
- **عدد البثوث** - 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** - رمز مشغل الشبكة المحمولة (MCC-MNC)
- **نوع RAT** - تقنية الوصول اللاسلكي (LTE، 5G، UMTS، GSM)

يمكن تحديد مناطق تتبع متعددة لتغطية مناطق أكبر أو مشغلين متعددين.

إنشاء رسالة بث خلايا

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، قصوى)
    - فترة التكرار (عادة 5-60 ثانية)
    - عدد البثوث (999 للمتواصل، 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 (GSM7، أقل لـ Unicode بحد أقصى 1395 حرفًا لـ)
   - 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/`

الإذن المطلوب: `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 الموجودة

عرض جميع الرسائل

نقطة النهاية: `GET /crm/cbc/`

الإذن المطلوب: `VIEW_CBC_MESSAGE`

يعيد قائمة بجميع رسائل CBC مع حالتها، والطوابع الزمنية، والتكوين.

تحديث رسالة

نقطة النهاية: `PATCH /crm/cbc/{cbc_message_id}`

الإذن المطلوب: `UPDATE_CBC_MESSAGE`

يحدث محتوى الرسالة، أو مناطق التتبع، أو معلمات البث. يتم إعادة 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}`

الإذن المطلوب: `DELETE_CBC_MESSAGE`

يزيل الرسالة من قاعدة البيانات ويحاول إلغاءها على CBC.

سير عمل الموافقة

تدعم رسائل بث الخلايا سير عمل موافقة اختياري لتنبيهات عالية المخاطر:

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`
- يتم الاحتفاظ بالرسالة في حالة "معلقة الموافقة"
- يقوم المستخدم الموافق بمراجعة الرسالة إما بالموافقة أو الرفض
- ويتم بث الرسالة `approving_user` عند الموافقة، يتم تعيين

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** - تحذير الزلزال والتسونامي ETWS المدمج
- **4373-4378** - أنواع الطوارئ الأخرى ETWS
- **4379** - رسالة اختبار ETWS

التنبيهات الطارئة اللاسلكية (WEA) / نظام التنبيه المحمول التجاري (CMAS):

- **4352** - تنبيه رئاسي
- **4353-4355** - تنبيهات قصوى
- **4356-4359** - تنبيهات شديدة
- **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

يحتاج كيان بث الخلايا (CBE) إلى آلية لتسليم الرسائل إلى الشبكات الخلوية الفردية. يتصل مركز بث الخلايا OWL (CBC) بكل شبكة خلوية لإرسال رسائل التحذير الطارئة إلى الجمهور.

التكرار عبر الشبكات المتعددة
في حال عدم توفر مشغل شبكة خلوية واحد (انقطاع أو عدم تغطية)، إذا كانت شبكة خلوية أخرى تعمل، سيظل المستخدمون يتلقون 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 التطبيقات 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 التنبيه

حذف الرسالة:

عند حذف رسالة، يحاول CBE إلغاءها على CBC لإيقاف البث المستمر.

نقاط تكامل الشبكة الخلوية

تمتلك أجيال مختلفة من الشبكات الخلوية (2G/3G/4G/5G) واجهات فريدة للاتصال بمركز بث الخلايا. بناءً على التقنيات المستخدمة من قبل كل شبكة خلوية، يجب تكوين الواجهة الصحيحة.

وقد تم دمجه مع العديد من 5G و4G و3G واجهات بث الخلايا 2 OWL يدعم مركز بث الخلايا مكونات الشبكة الخلوية الشائعة الاستخدام.

CBSP - 2G/3G - وحدة التحكم في محطة القاعدة (BSC)

يربط واجهة بروتوكول خدمة بث الخلايا (CBSP) CBC بوحدة التحكم في محطة القاعدة (BSC) التي تتحكم في محطات القاعدة 2G (GSM).

- 2 تُستخدم لبث رسائل بث الخلايا 3وG في لوحدة التحكم في مشتر��ة مع نشرات G
  الشبكة اللاسلكية/وحدة التحكم في محطة القاعدة
- 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 تنبيه للإشارة إذا كان رابط

3 حددت :ملاحظةGPP بروتوكول بث منطقة الخدمة (SABP) للاستخدام في نشرات RNC
RNC/BSC على Gلخلايا 3 CBSP المستقلة. يمكن استخدام ذلك إذا لزم الأمر عندما لا يتم دعم
RNC. مشتركة، ومع ذلك قد تكون هناك حاجة لاختبارات إضافية ودعم من بائع

SBc-AP - 4G/5G غير المستقل - MME/IWF

يربط واجهة SBc-AP OWL CBC بـ MME (كيان إدارة الحركة) 4 الذي يخدم 5وG
eNodeB/gNodeBs.

- LTE 4G تُستخدم لشبكات
- 5 تُستخدم أيضًا لـ G (معظم النشر اعتبارًا من 2025) غير المستقل
- OWL CBC في الشبكة و MMEs يجب إنشاء اتصال بين جميع
- SBc-AP قد تعطل يتم مراقبة الواجهات باستمرار مع تنبيه للإشارة إذا كان رابط

N50 - 5G مستقل - AMF

5 وظيفة إدارة الوصول) AMF بـ N50 OWL CBC المستقلة، يربط واجهة Gبالنس��ة لشبكات
5 (والحركة التي تخدم Gs gNodeB.

- OWL CBC الواجهة موجودة في
- من طرف ثالث بسبب العدد القليل من شبكات AMFs لم يتم اختبارها بشكل موسع مع
  5G SA 2025 المتاحة تجاريًا في
- 5 سيتم دعمها بالكامل مع تزايد نشرات G SA

متطلبات الشبكة لمشغل الشبكة المحمولة

وشبكة مشغل الشبكة المحمولة للوصول إلى OWL CBC يجب أن تكون الشبكة موجودة بين
الواجهات الموضحة أعلاه.

:يتم التعامل مع هذا على أساس كل حالة على حدة، ولكن 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 الشائعة الاستخدام من كبار البائعين:

تكامل بيانات موقع الخلية

يدعم OWL جمع البيانات تلقائيًا من:

- **Nokia NetAct**
- **Huawei U2000 / U2020**
- **ZTE NetNumen / ZXPOS**
- **Ericsson ENM**

بدلاً من ذلك، يمكن توفير بيانات موقع الخلية لفريق عمليات أومني تاتش بشكل دوري عبر البريد الإلكتروني.

إدارة المستخدمين والتحكم في الوصول

التحكم في الوصول بناءً على الدور (RBAC)

يستخدم نظام OWL التحكم في الوصول بناءً 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 للمستخدم المسجل 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 والقيام به
2. **تناسب العمليات** - تعكس الأدوار وظائف العمل (مدير، منشئ رسالة، موافق على الرسالة)
3. **إدارة بسيطة** - منح الوصول عن طريق تعيين الأدوار؛ تجنب الإدارة الدقيقة لكل مستخدم

أذونات النظام

تتبع أذونات النظام عمومًا أنماط CRUD مع أربعة 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

أذونات CBC الأساسية:

- `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 مؤسستك.

للحصول على معلومات شاملة حول إدارة أدوار المستخدمين والأذونات، انظر `المستخدمون والأدوار <rbac>`.

إدارة حساب المستخدم

تسجيل الدخول لأول مرة

عند إعداد مستخدم جديد في النظام، يتم تزويده ببيانات اعتماد تسجيل الدخول من قبل مدير النظام:

1. انتقل إلى عنوان URL لتسجيل الدخول إلى OWL الخاص بك المقدم من مدير النظام
2. أدخل اسم المستخدم وكلمة المرور الخاصة بك
3. إذا كانت 2FA مفعلة لحسابك، ستتم مطالبتك بإعدادها (أعلاه 2FA انظر قسم 2)

أفضل ممارسة: يجب على المستخدمين تغيير كلمة المرور الأولية الخاصة بهم على الفور بعد تسجيل الدخول الأول لأسباب أمنية. انتقل إلى إعدادات ملف تعريف المستخدم الخاص بك لتغيير كلمة المرور الخاصة بك.

للحصول على مزيد من التفاصيل حول تدفقات المصادق`تدفقات المصادقة والتحكم الإداري <authentication_flows>`، انظر.

إعادة تعيين كلمة المرور

يمكن للمستخدمين إعادة تعيين كلمات المرور الخاصة بهم:

1. من صفحة تسجيل الدخول إلى النظام، اختر "نسيت كلمة المرور"
2. اتبع عملية التحقق عبر البريد الإلكتروني
3. قم بتعيين كلمة مرور جديدة

إعادة تعيين كلمة مرور المسؤول:

يمكن لمديري النظام إعادة تعيين كلمات المرور للمستخدمين من صفحة **المستخدمون والأدوار** **المستخدمون** →:

1. انتقل إلى **المستخدمون والأدوار** → **المستخدمون**
2. اختر 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. يتم عرض كلمة المرور المؤقتة للمسؤول
6. قدم كلمة المرور المؤقتة للمستخدم عبر قناة آمنة (هاتف أو شخصيًا، وليس عبر البريد الإلكتروني)
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، انظر `المستخدمون والأدوار <rbac>`.

إضافة مستخدمين جدد للنظام

يمكن لمديري 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 نظام جديد يتماشى مع السياسات والإجراءات الحالية لمؤسستك. أضف المستخدمين فقط بعد الحصول على الموافقة المناسبة.

تحرير ملفات تعريف المستخدمين

يمكن لمديري النظام تغيير الأذونات والوصول للمستخدمين الحاليين:

1. انتقل إلى **المستخدمو�� والأدوار ← المستخدمون**
2. اختر المستخدم الذي تريد تحريره
3. عدل الأدوار المعينة حسب الحاجة
4. احفظ التغييرات

حذف مستخدمي النظام

يمكن لمديري النظام حذف المستخدمين من **المستخدمون والأدوار ← المستخدمون**.

تحذير: بمجرد حذف مستخدم من نظام OWL، لا يمكن التراجع عن هذا الإجراء. كن حذرًا جدًا قبل تنفيذ هذه المهمة.

عرض جميع مستخدمي النظام

في أي وقت، يمكن لمدير النظام عرض قائمة بجميع المستخدمين المسجلين ورؤية الأذونات والأدوار التي يمتلكها كل منهم من خلال 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** لعرض:

- جميع الرسائل المرسلة سابقًا
- حالة الرسالة (نشطة، منته��ة، ملغاة)
- الطابع الزمني 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:

الإذن المطلوب: `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 الأبراج** - 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** - تحديد 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، تقوم الخريطة تلقائيًا بتحميل مواقع أبراج الخلايا من `/cellSites.txt`. 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 مخصصة
    - انقر لإضافة نقاط
    - انقر نقرًا مزدوجًا لإكمال
    - ينشئ حدود تغطية دقيقة
2. انقر على **الدائرة** لرسم مناطق دائرية
    - انقر على نقطة المركز
    - اسحب لتعيين نصف القطر
    - مفيد للتنبيهات السريعة المعتمدة على نصف القطر

3. انقر على **المستطيل** لرسم مناطق مستطيلة
    - انقر على زاوية واحدة
    - اسحب إلى الزاوية المقابلة
    - فعال لتغطية متوافقة مع الشبكة

الخطوة 3: مراجعة المواقع المغطاة

بعد الرسم، يقوم النظام:

- بحساب المواقع الخلوية التي تقع ضمن المنطقة
- بعرض القائمة في الجدول أسفل الخريطة
- عرض أسماء المواقع وإحداثياتها
- توفير عدد الأبراج المتأثرة

جدول التغطية المثال:

الخطوة 4: تصدير بيانات التغطية

استخدم بيانات التغطية لـ:

- **تحديد رموز مناطق التتبع** للبث
- **تقدير مدى التنبيه** بناءً على عدد الأبراج
- **تخطيط الاستجابة للطوارئ** من خلال فهم المناطق المتأثرة
- **حفظ تعريفات المناطق** للاستخدام في المستقبل

تكامل سير العمل

يتكامل تصور الخريطة مع إنشاء البث:

تخطيط تنبيه:

1. **فتح خريطة CBC** لتصور المنطقة المتأثرة
2. **رسم منطقة التغطية** باستخدام أداة المضلع
3. **مراجعة المواقع المغطاة** وضبط الحدود
4. **تدوين رموز مناطق التتبع** لمواقع الخلايا المتأثرة
5. **إنشاء رسالة CBC** مع الاستهداف المناسب

مثال: تحذير تسونامي ساحلي

إدارة المناطق

حفظ المناطق:

يمكن حفظ مناطق التغطية المحددة مسبقًا لسيناريوهات شائعة:

- **المنطقة الساحلية** - لتحذيرات التسونامي/ارتفاع الأمواج
- **مناطق حرائق الغابات** - لتنبيهات الإخلاء
- في المناطق المأهولة AMBER - لتنبيهات **المراكز الحضرية**
- **الممرات السريعة** - لتنبيهات المرور/الطقس

تنسيق بيانات المنطقة:

يتم تخزين 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. يتم عرض المضلع على الخريطة تلقائيًا
3. يتم تحديث قائمة المواقع لعرض الأبراج المغطاة
4. تعديل المضلع إذا لزم الأمر

تكوين الخريطة

واجهة برمجة تطبيقات خرائط Google:

تتطلب الخريطة مفتاح واجهة برمجة تطبيقات خرائط Google تم تكوينه في البيئة:

```
REACT_APP_GOOGLE_API_KEY=your_google_maps_api_key_here
```

بيانات موقع الخلية:

قم بتحديث `public/cellSites.txt` بمواقع أبراج الخلايا لشبكتك:

```
# 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 مواقع الخلايا من:

- أدوات تخطيط الشبكة
- سجلات تركيب الأبراج
- المسوحات الميدانية باستخدام GPS
- تكوين وحدة التحكم في الشبكة اللاسلكية (RNC)

تصميم الخريطة:

يمكن تكوين أنماط الخريطة المخصصة في `mapStyles.js` لـ:

- تسليط الضوء على خدمات الطوارئ
- عرض الميزات الطبوغرافية
- التأكيد على مراكز السكان
- مطابقة العلامة التجارية للمنظمة

حالات الاستخدام

تخطيط الطوارئ:

- تخطيط مناطق التغطية مسبقًا لمناطق الخطر المعروفة

- اختبار استراتيجيات الاستهداف المختلفة
- تقدير مدى الوصول إلى السكان
- التنسيق مع خدمات الطوارئ

التحقق من التنبيه:

- تأكيد أن الرسالة ستصل إلى المنطقة المستهدفة
- تحديد فجوات التغطية
- تجنب التنبيه الزائد في المناطق المجاورة
- التحقق من رموز مناطق التتبع

تحليل الشبكة:

- تصور توزيع أبراج الخلايا
- تحديد تداخلات التغطية
- تخطيط نشر الأبراج لتحسين تغطية التنبيه
- تحسين كفاءة البث

أفضل الممارسات

بأفضل الممارسات التالية لجميع Omnitouch استنادًا إلى الخبرة مع العملاء حول العالم، توصي نشرات OWL.

محتوى الرسالة:

- حافظ على الرسائل مختصرة وواضحة (أقل من 360 حرفًا لتنبيهات صفحة واحدة)
- استخدم جميع الأحرف الكبيرة لتنبيهات الطوارئ (يحسن من قابلية القراءة)
- تضمين عناصر العمل المحددة ("الإخلاء فورًا"، "ابحث عن مأوى")
- تجنب المصطلحات الفنية
- اختبار الرسائل مع الأجهزة الفعلية قبل الطوارئ

دعم اللغة:

- دائمًا قدم الرسائل باللغة الأساسية للمنطقة
- تضمين لغات إضافية للمناطق متعددة الثقافات
- ضمان أن تكون الترجمات مناسبة ثقافيًا

- اختبار الأحرف الخاصة ودعم Unicode

الاستهداف الجغرافي:

- استخدم أصغر مناطق التتبع اللازمة لتجنب إرهاق التنبيه
- اعتبر كثافة السكان عند تعيين فترة التكرار
- اختبار الاستهداف الجغرافي قبل الطوارئ
- الحفاظ على وثائق دقيقة لمناطق التتبع

الاختبار:

- استخدم معرفات الرسائل الاختبارية (4379، 4380-4381) للتدريبات
- جدولة اختبارات النظام بانتظام
- يعمل CBC التحقق من أن تكامل
- تدريب الموظفين على إجراءات الطوارئ

إرهاق التنبيه:

- استخدم فقط لحالات الطوارئ الحقيقية
- تجنب البث الزائد
- تعيين فترات التكرار المناسبة
- استخدام مستويات الشدة بشكل مناسب

أمان حساب المستخدم

أمان رمز 2FA:

- حماية الرموز المادية 2FA مثل بطاقات الوصول إلى المباني
- الإبلاغ عن الرموز المفقودة أو المسروقة على الفور
- اختبار 2FA بانتظام للتأكد من أنها تعمل
- حفظ رموز النسخ الاحتياطي عند إعداد 2FA وتخزينها بشكل آمن في وضع عدم الاتصال
- الاتصال بمدير النظام إذا كنت بحاجة إلى إعادة تعيين 2FA

للحصول على مزيد من المعلومات حول إعداد 2FA واستعادته، انظر `المصادقة الثنائية <2fa>`.

صيانة البيانات

المناطق المستهدفة المحددة مسبقًا:

مع تغير الحدود الجغرافية، ووقوع التطورات، وتحول مناطق الخطر، هناك حاجة لمراجعة المناطق المستهدفة المحددة مسبقًا. تقترح Omnitouch مراجعة هذه البيانات **سنوياً من قبل NDMO**، مع دعم من وكالات التنبيه الأخرى عند الاقتضاء.

يمكن تعريف التحديثات للحدود باستخدام عدة منصات شائعة أو GIS، مثل Google Earth، ثم توفيرها عبر البريد الإلكتروني إلى فريق عمليات Omnitouch الذي سيقوم بإجراء التغييرات على النظام.

قوالب الرسائل المحددة مسبقًا:

يجب مراجعة قوالب الرسائل المحددة مسبقًا **سنويًا على الأقل من قبل NDMO**، مع دعم من وكالات التنبيه الأخرى عند الاقتضاء، لضمان:

- أن محتوى الرسالة لا يزال يعكس بدقة الخطر والدعوة للعمل
- أن معلومات الاتصال والتعليمات محدثة
- أن الترجمات اللغوية لا تزال دقيقة
- أن نبرة الرسالة وعجلتها مناسبة
- أن الإشارات إلى خدمات الطوارئ أو الإجراءات محدثة

يمكن توفير التحديثات عبر البريد الإلكتروني إلى فريق عمليات Omnitouch الذي سيقوم بإجراء التغييرات على النظام.

بيانات موقع الخلية:

- مراجعة وتحديث بيانات موقع الخلية كلما حدثت تغييرات في الشبكة
- تكرار التحديث النموذجي: شهري أو ربع سنوي
- التنسيق مع فرق تخطيط الشبكة لمشغل الشبكة المحمولة
- التحقق من الدقة بعد التحديثات أو التوسعات الكبيرة في الشبكة

بنية النظام

تم تصميم جميع منتجات Omnitouch لدعم النشر الجغرافي الموزع.

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
- **معدن عاري** - نشر مباشر على الأجهزة

البنية الموزعة

تسمح البنية الموزعة بـ:

- مكاتب إدارة الكوارث المحلية للوصول إلى OWL CBE وتوزيع الرسائل حتى إذا أصبحت منطقة معزولة عن الشبكة الوطنية
- كيانات بث خلايا متعددة ومراكز بث خلايا متعددة لكل مشغل/دولة
- وكالات الاستجابة المحلية للكوارث (الحكومة البلدية، الشرطة، الإطفاء، إلخ) لإصدار التنبيهات لمنطقتها حتى إذا فقد المكتب الوطني لإدارة الكوارث (NDMO) الوصول

هذا ذو قيمة خاصة عندما يكون لدى مشغلي الشبكات المحمولة موارد شبكة خلوية موزعة مع موارد BSC/MME محلية.

التواصل بين CBC وCBE

لضمان عدم وجود اتصال بين الشبكات المتنافسة، يمتلك كل مشغل شبكة متنقلة حالة CBC منفصلة (غير مشتركة).

متطلبات الشبكة بين CBE وCBC:

- يتم تشفير جميع حركة المرور بين CBE وCBC
- المصادقة تعتمد على الشهادات المتبادلة
- الاتصال من CBE إلى CBC عبر TLS على المنفذ TCP 443
- التنسيق مطلوب بين مشغلي الشبكات المحمولة و INDMO أو الوكالة المستضيفة لـ CBE

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 قابلاً للوصول في السيناريوهات غير المثالية مثل:

- انقطاعات واسعة النطاق لشبكات الطاقة العامة
- فشل شبكات الاتصالات
- الكوارث الطبيعية التي تؤثر على البنية التحتية

متطلبات النشر

متطلبات VM لـ CBE (NDMO / الحكومة)

3x آلات افتراضية:

- **2x CBE VMs** (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:** GB
- **نظام التشغيل:** نظام التشغيل الأساسي المقدم من Omnitouch
- **الشبكة:** السماح بحركة المرور إلى TCP 443 على المنفذ VMs CBC لحركة المرور TLS للتحكم في CBCs

متطلبات VM لـ CBC (MNO)

3x آلات افتراضية:

- **2x CBC VMs** (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:** GB
- **نظام التشغيل:** نظام التشغيل الأساسي المقدم من Omnitouch
- **الاتصال:** بكل من نقاط التكامل في الشبكة الخلوية (MME، RNC، BSC)

- **الشبكة:** السماح بحركة المرور من VMs CBE على المنفذ TCP 443 لحركة المرور التي يتم التحكم فيها بواسطة TLS CBE

خطوات التكامل

يتضمن نشر OWL الخطوات التالية:

1. **مصدر مشغل جديد/مشغلين** - تحديد مشغلي الشبكات المحمولة المشاركين
2. **إدارة المشروع (الإعداد)** - إنشاء حوكمة المشروع والإدارة
3. **تحديد موقع استضافة CBE** - تحديد مكان استضافة CBE
4. **تحديد المستخدمين وتدفقات/إجراءات الرسائل** - إنشاء سير عمل الموافقة وأدوار المستخدمين
5. **تحديد مضلعات لمناطق الاستهداف وقوالب الرسائل** - تكوين 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 Omnitouch
8. **نشر VMs CBE إلى NDMO** - تثبيت كيان بث الخلايا
9. **نشر VMs CBC إلى MNO** - تثبيت مركز بث الخلايا في كل مشغل
10. **تكوين عناصر الشبكة** - إعداد الاتصال بـ CBC من معدات الشبكة الخلوية
11. **التواصل بين VMs CBE وCBC** - إنشاء اتصال آمن
12. **التواصل بين VMs CBC وعناصر الشبكة** - الاتصال بـ BSC/MME/AMF
13. **إعداد الوصول إلى API لـ NMS** - تكوين تكامل بيانات موقع الخلية
14. **إعداد المراقبة واختبارها** - التحقق من المراقبة والتنبيه
15. **التحقق/اختبار التنبيهات الاختبارية** - 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 (الإنجليزية، الأحرف اللاتينية الأساسية) بت:

- 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 بـ 500 حرف
- تعرض واجهة الويب عدد الأحرف المتبقية وتحذر عند الاقتراب من الحدود

المراقبة والسجلات

يتم تسجيل نشاط بث الخلايا لأغراض التدقيق:

- أحداث إنشاء الرسالة، التحديث، والحذف
- المستخدمون المبادرون والموافقون
- الطوابع الزمنية ومعرفات الرسائل
- استجابات واجهة برمجة تطبيقات 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

تستقبل أجهزة المحمول المتوافقة رسائل بث الخلايا:

دعم الأجهزة:

- تدعم معظم الهواتف الذكية من عام 2015 فصاعدًا بث الخلايا
- قد تحتوي الهواتف البسيطة على دعم محدود
- يجب أن تكون الأجهزة متصلة بالشبكة (لا يلزم وجود رصيد بيانات/SMS)

- تعمل حتى أثناء ازدحام الشبكة عندما تفشل SMS

تجربة المستخدم:

- يتم عرض التنبيه كإشعار ملء الشاشة
- يتم تشغيل نغمة تنبيه فريدة
- يستمر التنبيه حتى يتم الاعتراف به
- لا يلزم اشتراك المستخدم
- لا يمكن حظره من قبل المستخدمين لتنبيهات رئاسية/قصوى

اختبار استقبال الجهاز:

للتحقق من أن الأجهزة يمكنها استقبال التنبيهات:

1. أرسل رسالة اختبار (معرف 4379 أو 4380-4381)
2. تأكد من أن الجهاز في منطقة التتبع المستهدفة
3. تحقق من أن الجهاز لديه بث الخلايا مفعل في الإعدادات
4. تحقق مع نماذج أجهزة متعددة وإصدارات نظام التشغيل

الوظائف الإضافية

يمكن توسيع منصة OWL مع ميزات اختيارية لتكملة رسائل بث الخلايا:

النص الجماعي / SMS

إرسال SMS إضافية لإشعارات أفراد ل��عادي.

- غالبًا ما يستخدم لتنبيهات منخفضة المستوى لإبلاغ الناس بأن التهديد الفوري قد زال، بطريقة أقل إزعاجًا
- أبطأ بكثير من بث الخلايا ولكن يمكن أن يتضمن آلية تأكيد للتحقق من استلام الرسالة
- مفيد للتواصل المستهدف للمتابعة

المكالمات الصوتية للخط الثابت

الاتصال تلقائيًا بأرقام الخط الثابت وتشغيل رسائل التحذير الطارئة.

- تشغيل رسالة مسجلة مسبقًا

- عرض نص إلى كلام لرسالة التحذير الطارئة
- يصل إلى السكان بدون هواتف محمولة
- يمكن التحقق من تسليم الرسالة من خلال إكمال المكالمة

النشر على وسائل التواصل الاجتماعي

نشر رسائل التحذير الطارئة تلقائيًا على القنوات الرسمية لوسائل التواصل الاجتماعي.

- يوسع 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
- عرض نص إلى كلام لرسالة التحذير الطارئة
- يصل إلى السكان أثناء انقطاع الشبكة
- يكمل بث الخلايا لتغطية شاملة

أجهزة التحذير الخارجية

يمكن لـ OWL CBC الاتصال بمجموعة متنوعة من المصادر الخارجية:

- منصات وسائل التواصل الاجتماعي
- واجهات برمجة التطبيقات العامة
- أنظمة المكالمات الصوتية
- أنظمة البث الإذاعي
- الإنذارات الفعلية والصفارات
- اللوحات 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 تعمل
- أن قوالب الرسائل محدثة وفعالة

الاختبارات الموصى بها:

- رسائل اختبار شهرية باستخدام المعرفات 4379، 4380-4381
- اختبارات نظام كاملة ربع سنوية بما في ذلك 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 Omnitouch لضمان دقة بيانات أداة التخطيط.

تكامل البيانات التلقائي

يدعم OWL جمع 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 Omnitouch بشكل دوري عبر البريد الإلكتروني بتنسيقات مختلفة.

تكرار التحديث: مراجعة وتحديث بيانات موقع الخلية كلما حدثت تغييرات في الشب��ة، عادةً شهريًا أو ربع سنويًا.

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 Google Earth شائعة أو GIS، ثم توفيرها لفريق عمليات Omnitouch.

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 توفير التحديثات عبر البريد الإلكتروني إلى فريق عمليات Omnitouch.

تدفقات الموافقة على الرسالة

تمتلك المناطق المختلفة متطلبات مختلفة بشأن إصدار الرسائل وتدفقات الموافقة.

قاعدة الشخصين: يُوصى باستخدام قاعدة الشخصين كلما كان ذلك عمليًا لضمان الإشراف في تقديم الرسالة.

أدوار المستخدمين الدقيقة: يمكن تكوين أدوار المستخدمين الفردية لـ:

- السماح فقط لمستخدمين معينين بإرسال رسائل محددة مسبقًا
- تقييد الاستهداف لمناطق معينة
- طلب خطوات موافقة إضافية

- تقليل خطر سوء الاستخدام

صيانة الأمان / 2FA

يمكن لفريق عمليات Omnitouch المساعدة في:

- إعادة تعيين رموز 2FA
- إعادة إصدار الرموز المفقودة/التالفة/المنتهية
- تدقيق الأمان وإدارة الرموز

إجراءات أمان الرمز:

عند إصدار الرموز، يتم تفصيل إجراء للخطوات التي يجب اتخاذها إذا فقد رمز أو لم يُحاسب عليه. 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 الموظفين المخولين
2. يتم تعطيل الرمز في النظام
3. يتم إجراء مراجعة أمنية
4. يتم إصدار رمز جديد بعد التحقق الأمني

التكامل مع واجهة برمجة التطبيقات الخارجية

تتوفر مجموعة كاملة من واجهات برمجة التطبيقات لكيان بث الخلايا للسماح للأنظمة من الطرف الثالث بالتكامل والتفاعل مع CBE.

قدرات واجهة برمجة التطبيقات:

- **التقارير/المراقبة** - التحقق من حالة ومدى 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** - استرجاع حالة الرسالة، وإحصائيات التسليم، ومقاييس النظام

انظر قسم مرجع واجهة برمجة التطبيقات أدناه للحصول على وثائق تفصيلية 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/` انظر وثائق Swagger في التفصيلية.

البحث العالمي

تقدم ميزة البحث العالمي واجهة بحث **موحدة** للعثور بسرعة على العملاء، جهات الاتصال، الخدمات، المخزون، والمواقع عبر قاعدة بيانات OmniCRM بأكملها.

انظر أيضًا: `Customers <basics_customers>`, `Inventory <administration_inventory>`, `Service Management <csa_service_management>`.

الوصول إلى البحث العالمي

من أي مكان في نظام إدارة علاقات العملاء:

انقر على أيقونة البحث في شريط التنقل العلوي أو انتقل إلى:

تظهر صفحة البحث العالمي مع مربع بحث كبير وخيارات تصفية.

كيف يعمل

يؤدي البحث العالمي **بحثًا عبر الكيانات** عبر خمسة أنواع من البيانات:

ما الذي يتم البحث عنه:

1. اسم العميل - **العملاء**
2. الاسم الأول، الاسم الأخير، عنوان البريد الإلكتروني، رقم الهاتف - **جهات الاتصال**
3. اسم الموقع - **المواقع**
4. أرقام السلاسل - **المخزون**، ICCIDs، المعرفات (itemtext1، itemtext2)
5. اسم الخدمة - **الخدمات**، UUID الخدمة

سلوك البحث:

- **مطابقة جزئية** - يبحث عن المصطلحات التي تحتوي على استعلامك (على سبيل المثال، "Smith" يتطابق مع "John Smith" و "Smithson")
- **غير حساسة لحالة الأحرف** - "john" يتطابق مع "John"، "JOHN"، و "john"
- **كيانات متعددة** - بحث واحد 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 السلسلة: `"8944"` أو فقط `"ICCID8944"`
- UUID الخدمة: `"e4567-e89b"` `123"`

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 عن:

- العملاء الذين لديهم `customer_status = "Closed"`
- جهات الاتصال، الخدمات، المواقع، والمخزون المرتبط بالعملاء المغلقين

حالات الاستخدام للبحث 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 المفسرة

نتائج العملاء:

النقر يفتح صفحة نظرة عامة على العميل تعرض جميع التفاصيل، الخدمات، جهات الاتصال، إلخ.

نتائج جهات الاتصال:

النقر يفتح صفحة العميل مع علامة التبويب جهات الاتصال نشطة، ويتوجه 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 (بدون customer_id)، فإنه يرتبط بقائمة المخزون الرئيسية بدلاً من ذلك.

نتائج الخ��مات:

النقر يفتح صفحة العميل مع علامة التبويب الخدمات نشطة، مع تسليط الضوء على الخدمة المحددة.

التنقل من النتائج

جميع نتائج البحث هي **روابط قابلة للنقر** التي 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

تحدد جزء الهاش (`#`) تلقائيًا علامة التبويب الصحيحة عند تحميل صفحة العميل.

تقسيم الصفحات

تُقسم النتائج إلى 10 عناصر لكل صفحة:

عرض النتائج 11-20 من 47

يمكن التنقل عبر الصفحات باستخدام:

- أزرار **السابق/التالي**
- **أرقام الصفحات** - انقر على صفحة معينة

- **لوحة المفاتيح** - الأسهم اليسرى/اليمنى (إذ�ً 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 نتيجة 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 تخصيص 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 إلكترونيًا إلى الدعم، يحتاج الموظف إلى العثور على حسابه.

> النتائج: • John Smith (جهة اتص��ل) - 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
- **جرب الاختلافات** - إذا لم تنجح "John"، جرب الهاتف أو البريد الإلكتروني
- **تضمين الحسابات المغلقة** - عند البحث عن البيانات التاريخية
- **كن محددًا للمعدات** - استخدم أرقام السلاسل 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 السلسلة MAC)
- itemtext2 (عادةً IMSI، معرف ثانوي)
- *ملاحظة: لا يتم البحث في itemtext3-20 أو ملاحظات المخزون*

الخدمات:

- اسم الخدمة
- UUID الخدمة

ما الذي 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)

الاستجابة:

```
{
  "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 (الخلفية):

تقوم 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

-- مخزون؟؟!
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 وإرجاعها.

اعتبارات الأداء

أداء البحث:

- تستخدم عمليات البحث استعلامات LIKE مع الأحرف البديلة (`%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 بمصطلحات البحث لتقليل مجموعة النتائج
- استخدم فلتر الحسابات المغلقة لتقليل نطاق البحث
- اعتبر إضافة فهارس قاعدة البيانات على الحقول التي يتم البحث عنها بشكل متكرر

استكشاف الأخطاء وإصلاحها

لم يتم العثور على نتائج (لكن السجل موجود)

- **السبب:** لا يتطابق مصطلح البحث 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
- **الإصلاح:** جرب الاختلافات، استخدم المطابقات الجزئية التي توجد بالتأكيد

البحث بطيء جدًا

- **السبب:** قاعدة بيانات كبيرة، استعلام معقد عبر جداول متعددة
- **الإصلاح:**
    - استخدم مصطلحات بحث أكثر تحديدًا
    - حصر البحث على الحسابات المفتوحة فقط (قم بإلغاء تحديد الحسابات المغلقة)
    - اتصل بالمسؤول حول فهرسة قاعدة البيانات

النتائج ترتبط بعميل خاطئ

- **السبب:** وجود عملاء/جهات اتصال متعددة بنفس الاسم
- **الإصلاح:** استخدم معرف العميل للتفريق، أو ابحث بواسطة معرف فريد (البريد الإلكتروني، الهاتف)

الحسابات المغلقة لا تظهر

- **السبب:** لم يتم تحديد مربع "تضمين الحسابات المغلقة"
- **الإصلاح:** تحقق من المربع وابحث مرة أخرى

الوثائق ذات الصلة

- `basics_customers` - إدارة العملاء
- `basics_navigation` - التنقل العام
- `administration_inventory` - عمليات البحث في المخزون

نظام التعبئة وإعادة الشحن

يوفر نظام التعبئة في OmniCRM للعملاء **بوابة إعادة شحن مسبقة الدفع للخدمة الذاتية** لإضافة رصيد أو تمديد صلاحية الخدمة عبر `Self-Care Portal <self_care_portal>`.
تُستخدم هذه الميزة عادةً لـ:

- بطاقات SIM مسبقة الدفع وخدمات البيانات فقط - **خدمات البيانات المحمولة**
- أجهزة WiFi المحمولة وأجهزة الإنترنت المحمولة - **خدمات النقاط الساخنة**
- الوصول إلى الإنترنت مسبق الدفع - **خدمات الإنترنت الثابتة**

لمعلومات حول `SIM Card Provisioning <concepts_sim_provisioning>` انظر أيضًا: توفير الخدمات المحمولة وإدارة مخزون SIM.

نظرة عامة

يسمح نظام التعبئة للعملاء بشراء أيام إضافية من الخدمة من خلال عملية دفع مبسطة متعددة الخطوات مع معال��ة دفع Stripe المدمجة.

الميزات الرئيسية:

- بوابة خدمة ذاتية للعملاء (لا حاجة لتدخل الموظفين)
- اختيار مرن للمدة (1-30 يومًا)
- عرض الاستخدام في الوقت الحقيقي قبل الشراء
- معالجة دفع آمنة مدعومة من Stripe
- استرداد تلقائي إذا فشلت التعبئة
- إنشاء الفواتير والمعاملات
- تكامل نظام التوفير لتفعيل الخدمة

الوصول إلى بوابة التعبئة

يتم الوصول إلى بوابة التعبئة عبر **رابط عام** يمكن للعملاء زيارته دون تسجيل الدخول إلى CRM:

كيف يصل العملاء إليها:

- عندما يكون الرصيد منخفضًا SMS رابط مباشر يُرسل عبر
- على المواد المطبوعة QR رمز
- رابط على بوابة الخدمة الذاتية
- مشاركة عبر دعم العملاء

تكتشف البوابة تلقائيًا خدمة العميل بناءً على عنوان IP أو IMSI الخاص بالطلب.

عملية التعبئة

تتكون عملية التعبئة من **4 خطوات**:

الخطوة 1: اختيار البيانات

يختار العملاء عدد ا��أيام من الخدمة التي يرغبون في شرائها.

الواجهة:

- **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 التعبئة

مثال على العرض:

تكوين التسعير:

- يتم تكوين سعر اليوم عبر متغير البيئة `REACT_APP_TOPUP_PRICE_PER_DAY`
- الافتراضي: 10 دولارات أمريكية في اليوم
- يتم تعيين العملة عبر `REACT_APP_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 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 مصادقة 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

غير OCS ،على سبيل المثال، خطأ في الشبكة) إذا نجح الدفع ولكن فشلت عملية توفير التعبئة
يقوم النظام تلقائيًا ببدء **استرداد كامل** لطريقة دفع العميل ،(متاح.

الخطوة 4: الإكمال

شاشة ال��جاح:

> تم تمديد خدمتك. تاريخ انتهاء الصلاحية الجديد: 17 يناير 2025
>
> معرف المعاملة: TXN-123456 <customer@example.com> :تم إرسال الإيصال إلى

شاشة الفشل:

إذا فشلت التعبئة، يعرض النظام خطأ ويقوم تلقائيًا بمعالجة استرداد:

> لم نتمكن من إتمام عملية التعبئة الخاصة بك. تم استرداد دفعتك.
>
> خطأ: غير قادر على الاتصال بنظام الفوترة
>
> يرجى المحاولة مرة أخرى أو الاتصال بالدعم.

معالجة الخلفية

عندما يكمل العميل الدفع، 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 التعبئة المزدوجة)

2. عملية التعبئة

```
- 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 السجلا��

ينشئ النظام سجلات قاعدة بيانات متعددة:

- **سجل HotspotTopup** - يتتبع معاملة التعبئة
  - payment_intent_id
  - service_uuid
  - imsi
  - الأيام المشتراة
  - topup_amount
  - الحالة (نجاح/فشل/مسترد)
- **سجل المعاملة** - معاملة مالية
  - العنوان: "Hotspot Topup - 7 Days"
  - المبلغ: topup_amount (إيجابي)
  - مرتبط بـ service_id و customer_id
- **سجل الفاتورة** - فاتورة الدفع
  - تحتوي على معاملة التعبئة
  - 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 تطبيقات 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 إعادة النجاح إلى العميل.

5. استرداد تلقائي عند الفشل

إذا فشلت 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 = "فشلت التعبئة. يتم استرداد الدفع..."
```

يتم معالجة الاسترداد تلقائيًا ويتم إبلاغ العميل على الشاشة.

نقاط نهاية واجهة برمجة التطبيقات

نقطة نهاية التعبئة

```
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
}
```

التحقق من الصحة:

- 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 والخدمة للعميل:

```
GET /oam/usage
```

الاستجابة:

```
{
  "imsi": "310120123456789",
  "service": {
    "service_uuid": "123e4567-e89b-12d3-a456-426614174000",
    "service_name": "Mobile Data - 0412345678",
    "service_status": "Active"
  },
  "balance": {
    "expiry": "2025-01-10T23:59:59Z",
    "unlimited": true
  },
  "requestingIp": "203.0.113.45"
}
```

تستخدم هذه الن��طة عنوان IP تلقائيًا لتحديد خدمة العميل المطلوب.

التكوين

متغيرات البيئة

قم بتكوين هذه في ملف OmniCRM-UI `.env`:

```
# تكوين بوابة التعبئة
REACT_APP_TOPUP_PRICE_PER_DAY=10
REACT_APP_CURRENCY_CODE=AUD
REACT_APP_SELF_CARE_NAME="YourCompany"

# تكوين Stripe
REACT_APP_STRIPE_PUBLISHABLE_KEY=pk_live_...
```

تكوين Stripe

يستخدم نظام التعبئة نوايا الدفع من Stripe:

1. **تمكين نوايا الدفع** في لوحة التحكم الخاصة بك في Stripe
2. **تكوين Webhook** (اختياري ولكن موصى به) لتلقي تحديثات حالة الدفع
3. **إعداد طرق الدفع** (البطاقات، المحافظ، إلخ)

4. استخدم مفاتيح الاختبار للتطوير - **وضع الاختبار**

```
# التطوير
REACT_APP_STRIPE_PUBLISHABLE_KEY=pk_test_...

# الإنتاج
REACT_APP_STRIPE_PUBLISHABLE_KEY=pk_live_...
```

تكوين Playbook

يجب تكوين playbook للتوفير `play_topup_hotspot.yaml` لـ:

- قبول متغير `days`
- الاتصال بواجهة برمجة تطبيقات OCS/CGRateS
- إضافة رصيد إلى الحساب
- تحديث تاريخ انتهاء الخدم

مثال playbook هيكل:

```yaml
- name: Top up hotspot service
  hosts: localhost
  tasks:
    - name: Add balance to 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 تفعيلها بواسطة أحداث OCS (`Action_Balance_Low`, `Action_Balance_Out`, `Action_Balance_Expired`):

إشعارات البريد الإلكتروني:

مُعدة في خطط عمل OCS/CGRateS لإرسال تنبيهات الرصيد.

محفزات الإشعار:

- `Action_Balance_Low` - الرصيد أقل من العتبة (على سبيل المثال، تبقى 2 يوم)
- `Action_Balance_Out` - الرصيد مستنفد
- `Action_Balance_Expired` - انتهت الخدمة

تتضمن كل إشعار رابط بوابة التعبئة لسهولة وصول العملاء.

استكشاف الأخطاء وإصلاحها

المشكلات الشائعة

"نظام الدفع غير متاح"

- **السبب:** فشل تحميل مكتبة Stripe أو مفتاح قابل للنشر غير صالح
- **الإصلاح:**
  - تحقق من تعيين `REACT_APP_STRIPE_PUBLISHABLE_KEY` بشكل صحيح
  - تحقق من أن حساب Stripe نشط
  - تحقق من وحدة تحكم المتصفح بحثًا عن أخطاء JavaScript

"فشلت التعبئة. يتم استرداد الدفع..."

- **السبب:** فشلت 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 تطبيقات OCS/CGRateS متاحة
  - 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 (على سبيل المثال، تحديث بعد النجاح)
- **الإصلاح:** هذا يعمل كما هو مصمم لمنع الفوترة المزدوجة. يجب على العميل بدء تعبئة جديدة إذا لزم الأمر.

"مبلغ نية الدفع لا يتطابق"

- **السبب:**�عدم تطابق بين حساب واجهة المستخدم والتحقق من الخلفي
- **الإصلاح:**
  - تحقق من أن `REACT_APP_TOPUP_PRICE_PER_DAY` يتطابق مع توقعات الخلفية (افتراضي 10 دولارات)
  - تحقق من أن تكوين العملة متسق
  - امسح ذاكرة التخزين المؤقت للمتصفح وحاول مرة أخرى

مراقبة التعبئة

عرض سجلات التعبئة:

استعلام عن جدول `HotspotTopup` لرؤية جميع محاولات التعبئة:

```
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 التعبئة.

لوحة تحكم Stripe:

راقب المدفوعات، والاستردادات، والمعاملات الفاشلة في لوحة تحكم Stripe الخاصة بك على <https://dashboard.stripe.com>

اعتبارات الأمان

أمان الدفع:

- يتم التعامل مع جميع بيانات البطاقة بواسطة Stripe (متوافق مع PCI المستوى 1)
- 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:

- يكتشف تكرار نية الدفع لمنع الفوترة المزدوجة
- تتبع عنوان IP لارتباط الاستخدام
- يضمن التحقق من IMSI أن التعبئة تذهب إلى الخدمة الصحيحة
- تحد من الاستردادات التلقائية التعرض المالي

التحكم في الوصول:

- بوابة التعبئة عامة (حسب التصميم - يحتاج العملاء إلى الوصول)
- تتطلب نقطة نهاية الاستخدام تحديد خدمة صالح (IMSI أو IP)
- يمنع التحقق من الخلفية التعبئة العشوائية للخدمات
- يمكن للمسؤول عرض جميع سجلات التعبئة عبر واجهة CRM

أفضل الممارسات

للمشغلين:

1. **اختبر تدفق الاسترداد** - اختبر بانتظام سيناريوهات الفشل لضمان عمل الاستردادات
2. **راقب التعبئات الفاشلة** - قم بإعداد تنبيهات لمعدلات الفشل العالية

3. **اجعل playbooks بسيطة** - يجب أن تكون playbooks التعبئة سريعة وموثوقة
4. **تحقق من اتصال OCS** - تأكد من أن واجهة برمجة تطبيقات OCS متاحة دائمًا
5. **راجع التسعير** - قم بتحديث `REACT_APP_TOPUP_PRICE_PER_DAY` حسب الحاجة

للعملاء:

1. **احفظ رابط التعبئة** - وصول سريع عند الحاجة
2. **احفظ إشعارات الرصيد المنخفض** - تحتوي رسائل SMS على رابط مباشر
3. **تحديث البريد الإلكتروني** - تُرسل الإيصالات إلى البريد الإلكتروني المسجل
4. **تحقق من انتهاء الصلاحية قبل السفر** - قم بالتعبئة قبل مغادرة منطقة التغطية

للمطورين:

1. **تعامل مع webhooks من Stripe** - نفذ معالجات webhook لتحديثات حالة الدفع
2. **نفذ idempotency** - تحقق دائمًا من payment_intent_id قبل المعالجة
3. **سجل بشكل مكثف** - تحتاج حالات فشل التعبئة إلى معلومات استكشاف الأخطاء التفصيلية
4. **اختبر مسارات الخطأ** - تحقق من أن أتمتة الاسترداد تعمل بشكل صحيح
5. **راقب الأداء** - يجب أن يكتم�� استطلاع التوفير في <5 ثوانٍ

الوثائق ذات الصلة

- `payments_process` - معالجة الدفع العامة
- `concepts_provisioning` - نظرة عامة على نظام التوفير
- `Payment System Guide <payment_system_guide>` - تفاصيل تكامل بائع الدفع
- `payments_transaction` - إدارة المعاملات
- `payments_invoices` - معالجة الفواتير

دليل دورة حياة المنتج الشامل

يوفر هذا الدليل جولة شاملة عبر دورة حياة المنتج في OmniCRM، بدءًا من إنشاء تعريف المنتج وحتى توفير الخدمات، وإضافة الإضافات، وإلغاء التوفير. سنغطي استراتيجية التسعير، ودمج 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 الإضافات/المدفوعات الإضافية، ويعدل الخدمة
4. **إلغاء التوفير** - يتم إنهاء الخدمة، وتحرير الم��ارد

فهم التسعير: الجملة مقابل التجزئة

كل منتج وخدمة في OmniCRM له بعدان للتسعير: **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 الذي يتم تحصيله من العميل.

تكاليف الإعداد

تحتوي كل من الجملة والتجزئة على متغيرات تكلفة إعداد لرسوم التوفير لمرة واحدة:

- `wholesale_setup_cost` - التكلفة الخاصة بك للتوفير
- `retail_setup_cost` - المبلغ المحصل من 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 منتج

المنتجات هي قوالب تحدد ما يتم توفيره وكيف يتم تحصيل الرسوم من العملاء.

إنشاء منتج بطاقة SIM موبايل

ننشئ منتج بطاقة SIM دعون�� موبايل مسبقة الدفع مع 20 جيجابايت من البيانات شهريًا.

الخطوة 1: الانتقال إلى إدارة المنتجات

من واجهة المسؤول، انتقل إلى **المنتجات** ← **إنشاء منتج**.

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 يظهر للعملاء في الكتالوج
- `product_slug` - معرف آمن للعنوان يستخدم في استدعاءات API والروابط
- `category` - "standalone" (مقابل الإضافة/الحزمة) تعني أن هذا ينشئ خدمة جديدة
- `service_type` - يجمع المنتجات ذات الصلة، يستخدم لتصفية الإضافات
- `enabled` - يجب أن يكون صحيحًا ليكون المنتج قابلًا للطلب
- `icon` - أيقونة FontAwesome المعروضة في واجهة المستخدم
- `comment` - مل��حظات داخلية للموظفين

الخطوة 3: تعيين التسعير

```
{
  "retail_cost": 15.00,
  "wholesale_cost": 5.00,
  "retail_setup_cost": 0.00,
  "wholesale_setup_cost": 1.00,
  "contract_days": 30
}
```

تفاصيل 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: تعريف 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 تاريخ انتهاء (عرض مستمر)

الخطوة 5: تكوين التجديد التلقائي

```
{
  "auto_renew": "prompt",
  "allow_auto_renew": true
}
```

- `"prompt"` - اسأل العميل إذا كانوا يريدون التجديد التلقائي عند الشراء
- `"true"` - تجديد تلقائي بدون سؤال
- `"false"` - عدم التجديد التلقائي أبدًا (إضافة 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 بنظام إدارة المخزون الخاص بك `Inventory Management System <administration_inventory>`.

```
{
  "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 التسلسل، عناوين MAC، أرقام الهواتف، إلخ.
- **الحالة** - في المخزون، مخصص، متقاعد، إلخ.
- **الموقع** - الموقع المادي أو المنطقي

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 `Inventory Template <administration_inventory>`.

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 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. يظهر نموذج اختيار المخزون

إذا كانت المخزون مطلوبة، يظهر مربع حوار مع قائمة منسدلة منفصلة لكل نوع من المخزون:

3. تصفية المخزون المتاح

تظهر القائمة المنسدلة لكل نوع من المخزون فقط 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 بواسطة موقع المستودع/المتجر

مثال على خيارات القائمة المنسدلة:

بالنسبة لنوع مخزون "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 للمضي قدمًا

قاعدة حاسمة: لا يمكن أن يستمر التوفير دون اختيار جميع عناصر المخزون المطلوبة.

- يت�� تعطيل زر "متابعة" حتى يتم اختيار جميع القوائم المنسدلة
- يجب على المستخدم اختيار عنصر واحد لكل نوع من 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:
# - بطاقة SIM 5001 :معرف المخزون
# - 5002 :رقم الهاتف معرف المخزون

# 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 SIM (ICCID، IMSI، إلخ) في المشترك لتوفير 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. وهذا 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
- **الرجوع عند الفشل**: إذا فشل البرنامج قبل تخصيص المخزون، يبقى المخزون غير مخصص ومتاح لإعادة الاستخدام

لماذا لا يتم التخصيص في وقت سابق؟

لا يتم تخصيص المخزون عند النقر على "توفير" لأن:

1. حتى يتم إنشاء الخدمة في البرنامج `service_id` **معرف الخدمة مطلوب**: لا يوجد
2. لا يحتاج، (مثل فشل إنشاء حساب OCS) **بساطة الرجوع**: إذا فشل التوفير مبكرًا المخزون إلى تنظيف
3. **المرونة**: يمكن للبرنامج أن يقرر عدم تخصيص المخزون بناءً على المنطق الشرطي

التعامل مع التوفيرات الفاشلة:

عندما يفشل التوفير بعد تخصيص المخزون، يجب أن يحرر كتلة الإنقاذ 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:

- **المنتجات الرقمية** - تراخيص البرمجيات، حسابات VPN
- **إضافات الخدمة** - مدفوعات إضافية للبيانات التي لا تحتاج إلى أجهزة جديدة
- **الخدمات الافتراضية** - التي لا تستهلك موارد قابلة للتتبع

مثال: إضافة "5GB Data Boost" تحتوي على `inventory_items_list: "[]"` لأنها تضيف فقط رصيدًا إلى خدمة موجودة دون الحاجة إلى أجهزة جديدة.

إعداد قالب المخزون

يجب عليك إنشاء قالب المخزون، `inventory_items_list` قبل استخدام نوع المخزون في:

1. انتقل إلى **الإدارة** → **المخزون** → **القوالب**
2. أنشئ قالبًا بنفس الاسم بالضبط (مثل "SIM Card")
3. حدد الحقول:
   - `itemtext1_label`: "ICCID"
   - `itemtext2_label`: "IMSI"
   - `itemtext3_label`: "PUK Code"

4. أضف عناصر المخزون من هذا النوع إلى المخزون

للحصول على تفاصيل كاملة حول إنشاء وإدارة قوالب المخزون، راجع `Inventory Management <administration_inventory>`.

عناصر متعددة من نفس النوع

بينما `inventory_items_list` هو 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 مميزة

```
# خدمة هاتف ثنائي الشريحة
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 ثنائية الشريحة
inventory_items_list: "['Dual SIM Kit', 'Mobile Number']"
```

حيث يحتوي قالب المخزون "Dual SIM Kit" على حقول لكلا الشريحتين (`itemtext1`: ICCID الأساسي، `itemtext2`: ICCID الثانوي، إلخ.).

سيناريوهات المخزون الشائعة

خدمة موبايل:

```
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 راوتر: 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، وكلمات المرور، وتكوينات الحساب **تُولد برمجيًا** بواسطة برنامج التوفير، وليس من اختيار المخزون.

GPON/الألياف:

```
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 ا��ألياف المادي من الشارع إلى المباني (يتم تتبعه لإدارة الأصول)

تأجير المعدات:

```
inventory_items_list: "['Rental Modem']"
```

- يتتبع أي مودم مع أي عميل
- مهم لاستعادة المعدات عند الإلغاء

لماذا تهم متطلبات المخزون

1. منع التخصيص المزدوج

بدون تتبع المخزون، يمكنك عن غير قصد:

- تخصيص نفس بطاقة SIM لعميلين
- تخصيص نفس عنوان IP لعدة خدمات
- شحن نفس رقم تسلسل المعدات إلى مواقع مختلفة

يضمن اختيار المخزون أن يتم تخصيص كل عنصر لخدمة واحدة فقط.

2. تتبع التدقيق

يخلق تخصيص المخزون تتبع 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 انخفاض بطاقات SIM
- إعادة الطلب قبل نفاد المخزون
- تخطيط جداول الفنيين بناءً على توفر CPE

- إدار�� IP تخصيص مساحة عنوان

4. تتبع التكلفة

ربط تكلفة الجملة بعنصر معين:

- تتبع تكلفة كل بطاقة SIM
- حساب استهلاك المعدات
- تحديد العناصر المفقودة أو المسروقة
- دقيقة (تكلفة السلع المباعة) COGS

5. إلغاء التوفير

عند إلغاء الخدمة، يمكن أن يكون المخزون:

- مُعاد إلى المخزون (بطاقات SIM، مودمات)
- مُتقاعد (معدات تالفة)
- مُعاد إلى البائع (معدات تأجير)
- مُحتفظ به لفترة السماح (أرقام الهواتف قبل الإفراج)

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 على أنها "تالفة" 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 "جديدة" أو "في المخزون"
5. تحقق من أن العناصر ليست مخصصة بالفعل (`service_id` يجب أن يكون NULL)

المشكلة: لا يظهر اختيار المخزون

الأسباب:

- `inventory_items_list` فارغة: `"[]"`
- `inventory_items_list` أو غير محدد NULL
- فئة المنتج هي "إضافة" وتورث مخزون الخدمة الأصلية

الحل:

- إذا كان المخزون مطلوبًا، قم بتعيين `inventory_items_list: "['Type1', 'Type2']"`
- تحقق من حفظ تعريف المنتج 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

للحصول على تفاصيل `Inventory Management <administration_inventory>` راجع كاملة حول إنشاء القوالب، وإضافة العناصر، وإدارة مستويات المخزون.

الخطوة 7: تعريف الميزات والشروط

الميزات والشروط هي محتوى تسويقي وقانوني موجه للعملاء 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 عدة وظائف تجارية حاسمة:

1. **تمييز المنتج** - يساعد العملاء على مقارنة المنتجات بسرعة واختيار المنتج المناسب
   - الميزات - "Prepaid Mobile 50GB" مقابل "Prepaid Mobile 20GB" توضح الفرق 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 الالتزامات
   - تدفع الميزات قرارات الشراء
3. **توقعات العملاء** - تحدد توقعات واضحة حول ما هو مدرج
   - تقلل من مكالمات الدعم ("هل يشمل هذا المكالمات؟" → مدرج 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 موبايل مكالمات غير محدودة" → يظهر المنتج
   - يحسن من قابلية البحث في كتالوج المنتجات

الشروط والأحكام - قانونية وامتثال:

تخدم الشروط أغراضًا قانونية وتشغيلية:

1. **الحماية القانونية** - تحمي الأعمال من النزاعات والمسؤولية
    - "Credit expires after 30 days" - لا يمكن للعميل المطالبة باسترداد بعد 31 يومًا
    - "Fair use policy applies" - تمنع الإساءة (توصيل مكتب كامل على خطة موبايل)
    - تخلق اتفاقًا ملزمًا
2. **إدارة التوقعات** - تمنع عدم رضا العملاء
    - "Valid only within expiry period" - يعرف العميل موعد انتهاء الاستخدام
    - "Cannot be refunded" (للإضافات) - تمنع عمليات الشراء الاحتيالية
    - تقلل من الشكاوى والاعتراضات
3. **الامتثال التنظيمي** - تلبي المتطلبات القانونية
    - تتطلب قوانين حماية ��لمستهلك شروطًا واضحة
    - تفرض لوائح الاتصالات الإفصاح
    - يمكن الإشارة إلى شروط GDPR/الخصوصية
4. **الحدود التشغيلية** - تحدد نطاق الخدمة والقيود
    - "Subject to network coverage" - غير مسؤول عن المناطق الميتة
    - "Speed may vary" - حتى "تدير التوقعات بشأن السرعات"
    - "Equipment must be returned" - تضمن استرداد المعدات المستأجرة
5. **تتبع التدقيق** - يثبت أن العميل تم إبلاغه
    - قبل العميل الشروط عند الشراء
    - يسجل النظام توقيت قبول الشروط
    - قابل للدفاع 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: "قلت غير محدودة!"
- المزود: "كنا نعني الاستخدام الشخصي..."
- العميل: "هذا ليس ما أعلنتم عنه!"
- النتيجة: نزاع، شكوى محتملة للجهات التنظيمية، ضرر ��لعلامة التجارية

مع الشروط: "تطبق 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 حاسم لأن **التنسيق غير الصحيح يكسر عرض واجهة المستخدم**. قد تظهر الميزات كسلسلة طويلة واحدة بدلاً من نقاط قائمة، 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 بالتقسيم على هذا الفاصل وتعرض كل ميزة كنقطة قائمة.

لماذا هذا التنسيق؟

- سهل التحرير - فقط اكتب الميزات مع نقاط بينهما
- لا أحرف خاصة للهروب
- يعمل بشكل موثوق عبر جميع مكونات واجهة المستخ��م
- 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 Mbps. تخضع للتوافر.")
- أسهل التوليد البرمجي من البرامج/API

- مستورد من كتالوجات المنتجات الخارجية التي تستخدم المصفوفات

مهم: يجب أن يكون هذا بناء جملة قائمة بايثون صالحًا كسلسلة. علامات الاقتباس الفردية حول كل عنصر، وعلامات الاقتباس المزدوجة حول السلسلة بالكامل.

أي تنسيق يجب استخدامه؟

- **مفصولة بنقطة** - لإنشاء المنتجات يدويًا في واجهة المستخدم (أبسط، أقل عرضة للأخطاء)
- **مصفوفة JSON** - لإنشاء المنتجات بناءً على API/(أكثر قوة للميزات المعقدة) البرامج

ينتج كلا التنسيقين 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، تُعرض الميزات كنقاط قائمة 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 قائمة)
- الشروط والأحكام (انظر أدناه)
- التوفر والأهلية

3. تأكيد التوفير

أثناء التوفير، تُظهر الميزات للمستخدم للمراجعة قبل التأكيد:

> 20 • :الميزاتGB High-Speed Data • Unlimited Calls & Texts • EU Roaming Included • No Contract • 30-Day Expiry
>
> التكلفة: £15.00/شهر الإعداد: £0.00
>
> [تأك��د وتوفير] [إلغاء]

4. تفاصيل الخدمة (بعد التوفير)

بعد أن تصبح الخدمة نشطة، تُعرض الميزات في صفحة تفاصيل الخدمة للرجوع إليها من قبل العميل.

تنسيق الشروط والأحكام:

حقل `terms` هو نص عادي يمكن أن يتضمن أسطر جديدة:

أين تظهر الشروط في واجهة المستخدم:

1. صفحة تفاصيل المنتج

تُعرض الشروط في قسم مُنهار يتوسع عند النقر عليه:

2. تأكيد الطلب

أثناء التوفير، يتطلب مربع اختيار من المستخدم قبول الشروط:

```
زر [توفير] معطل حتى يتم التحقق منه
```

3. الفواتير

يمكن أن تتضمن شروط الخدمة على الفواتير كحواشي توضيحية.

أفضل الممارسات:

- **الميزات:** احتفظ بها مختصرة (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 لإبقاء ال��ملاء على اطلاع

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 إضافي

تعمل الإضافات على تحسين أو تعديل الخدمات الحالية. تأتي في نوعين: **إضافات افتراضية** (لا موارد مادية) و **إضافات الأجهزة** (تتطلب المخزون).

مثال 1: إضافة افتراضية (5GB Data Boost)

إضافة رقمية تضيف 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: إضافة أجهزة (تأجير مودم)

إضافة توفر معدات مادية لخدمة الألياف الحالية:

```
{
  "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 للإضافات:

- `category: "addon"` - تطبق على الخدمة الحالية، وليست مستقلة
- `contract_days: 0` (افتراضية) - تكرار الفوترة (إيجار متكرر) `30` أو

- `inventory_items_list: "[]"` (افتراضية) أو `"['Rental Modem']"` (أجهزة) - موارد مادية
- `auto_renew: "false"` (لمرة واحدة) أو `"true"` (إيجار) - سلوك متكرر
- `relies_on_list: ""` - فارغ يعني أنه ينطبق على أي خدمة من نوع `service_type` المطابق

لماذا تحتاج إضافات الأجهزة إلى المخزون:

تتطلب إضافات الأجهزة `inventory_items_list` لأن:

1. **تتبع المعدات** - معرفة أي مودم مع أي عميل
2. **منع نفاد المخزون** - لا يمكن توفير الإضافة إذا لم يكن هناك مودمات في المخزون
3. **الاسترداد** - عند إلغاء العميل، معرفة أي معدات لاستردادها
4. **تتبع التكلفة** - ربط تكلفة الجملة برقم تسلسلي معين
5. **الاستهلاك** - تتبع قيمة المعدات على مدار فترة الإيجار
6. **الضمان** - تحديد الوحدات المعيبة برقم تسلسلي

تدفق توفير الإضافات مع المخزون:

عندما يضيف العميل "تأجير مودم WiFi 6" به إلى خدمة الألياف الخاصة به:

1. **تم اختيار الإضافة** - ينقر العميل على "إضافة إلى الخدمة"
2. **يظهر اختيار المخزون**: - تمامًا مثل الخدمات المستقلة
3. **تم معالجة الدفع** - يتم تحصيل 10.00£ إيجار شهري
4. **تم تخصيص المودم** - تم تحديث المخزون:
   - `service_id`: مرتبط بخدمة الألياف
   - `customer_id`: مرتبط بالعميل
   - `item_state`: "مخصص"
5. **تم الشحن** - تم إبلاغ نظام التنفيذ بشحن المودم
6. **التثبيت** - يتلقى العميل المودم، ويقوم بتوصيله بـ ONT
7. **الفوترة المتكررة** - يتم تحصيل 10£ شهريًا حتى يتم إلغاء الإضافة

إلغاء توفير الإضافات الأجهزة:

عند إلغاء العميل لتأجير المودم:

1. **تم بدء الإلغاء** - ينقر العميل على "إزالة الإضافة"

2. **تم بدء عملية الإرجاع**:
    - تم إرسال بريد إلكتروني مع تعليمات الإرجاع
    - تم إنشاء ملصق شحن مدفوع مسبقًا
    - فترة سماح لمدة 14 يومًا قبل فرض غرامة
3. **تمت إعادة المعدات**:
    - تم تحديث المخزون: `item_state` = "في المخزون" (بعد التجديد)
    - أو `item_state` = "تالف" (إذا كانت معيبة)
    - مرتبط بالعميل التالي بمجرد تجديده
4. **عدم الإرجاع**:
    - بعد 14 يومًا، يتم فرض غرامة استبدال بقيمة 150£
    - يتم وضع علامة على المخزون: `item_state` = "مفقود"
    - يتم استرداد تكلفة الجملة (45£) + قيمة الاستبدال

التسعير للإضافات:

يمكن تسعير الإضافات بشكل مختلف عن الخدمات المستقلة:

- عادةً ما لا تحتوي الإضافات الافتراضية على تكاليف إعداد
- قد تحتوي الإضافات الأجهزة على تكاليف إعداد جملة للمعدات
- تستخدم الإضافات الإيجارية `contract_days` لتكرار الفوترة

المرحلة 2: عملية التوفير

عندما يطلب العميل منتج "Prepaid Mobile 20GB"، تنظم OmniCRM التوفير من خلال Ansible.

مخطط تدفق التوفير

طلب العميل → اختيار المخزون → إنشاء وظيفة التوفير ↓ ↓ تفويض الدفع ← تجميع المتغيرات ← تخصيص المخزون → → إنشاء سجل الخدمة في OCS ← إعداد حساب ← 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 المنتجات
- تُعرض 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** - 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 "تشغيل" (status=1)
- دمج المتغيرات من المنتج + الطلب + اختيارات المخزون
- يطلق Ansible خيطًا في الخلفية لتنفيذ برنامج
- 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

دعونا نفحص 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
```

تفصيل البرنامج:

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: استرجاع 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 هذه المهمة:

- يستدعي `GET /crm/product/product_id/42`
- يسترجع تعريف المنتج الكامل
- يخزن في متغير `api_response_product`

لماذا: على الرغم من أن لدينا `provisioning_json_vars` من المنتج، فإننا نسترجع المنتج 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 هذه المهمة:

- يبحث عن معرف المخزون لبطاقة SIM 5001
- يسترجع تفاصيل SIM:
  - `itemtext1` = ICCID (رقم بطاقة SIM)
  - `itemtext2` = IMSI (هوية المشترك)
- يفعل نفس الشيء لرقم الهاتف المحمول (يسترجع رقم الهاتف)

لماذا يهم هذا:

- 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 هذه المهمة:

- يولد UUID عشوائي
- ينشئ service_uuid مثل `Local_Mobile_SIM_a3f2c1d8`

لماذا:

- service UUID هو المعرف الفريد في OCS/CGRateS
- يستخدم لجميع عمليات الشحن
- يجب أن يكون فريدًا عالميًا عبر جميع الخدمات

المهمة 6: إنشاء حساب في 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 هذه المهمة:

- يستدعي واجهة برمجة التطبيقات JSON-RPC لـ CGRateS
- ينشئ حسابًا جديدًا مع service_uuid

- يضبط الحساب على نشط (غير معطل)
- يمنع الرصيد السالب (وضع مسبق)

لماذا:

- هو المكان الذي تحدث فيه جميع عمليات الشحن OCS حساب
- هنا (المال، SMS، البيانات، الصوت) يتم تخزين الأرصدة
- يتم تتبع الاستخدام وتقييمه في الوقت الحقيقي

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 هذه المهمة:

- يضيف رصيد 20 جيجابايت إلى الحساب
- القيمة: 21474836480 بايت (20 * 1024 * 1024 * 1024)
- تنتهي صلاحيتها في 720 ساعة (30 يومًا)
- الوزن 10 (يتم استهلاكه أولاً)

المهمة 8: إضافة صوت غير محدود و SMS

```
- 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 هذه المهمة:

- يحدث 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 تخصيص المخزون المزدوج
- 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 هذه المهمة:

- يسجل تكلفة إعداد 0.00£ للعميل (تجزئة)
- يسجل تكلفة جملة 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 هذه المهمة:

- إذا فشلت أي مهمة، يتم تنفيذ كتلة الإنقاذ
- الذي تم إنشاؤه جزئيًا OCS يحذف حساب
- يحرر المخزون مرة أخرى إلى "في المخزون"
- يفشل وظيفة التوفير مع رسالة خطأ

لماذا:

- OCS يمنع وجود حسابات يتيمة في
- يضمن الرجوع النظيف عند حدوث أخطاء
- يحافظ على اتساق البيانات

اكتمل التوفير: 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 في 30 يومًا)

- صوت غير محدود (999M يومًا 30 في تنتهي، ثانية)
- SMS غير محدود (999M يومًا 30 في تنتهي، رسالة)

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"
- service UUID: `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 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 SMS
- إعادة الشحن أو إضافة إضافات
- عرض الاستخدام في الوقت الحقيقي

المرحلة 3: إضافة الإضافات والمدفوعات الإضافية

بعد أن تصبح الخدمة نشطة، يمكن للعملاء شراء الإضافات لتعزيز خدمتهم.

تدفق توفير الإضافات

لنفترض أن العميل قد استخدم 18 جيجابايت من رصيد 20 جيجابايت ويريد شراء إضافة "5GB Data Boost".

1. يتنقل العميل إلى الخدمة

- يفتح صفحة "Mobile - +44 7700 900123"
- يرى الاستخدام الحالي: 18 جيجابايت من 20 جيجابايت مستخدمة (90%)
- ينقر على "إضافة إضافة" أو "إعادة شحن"

2. يقوم النظام بتصفية الإضافات المتاحة

يظهر فقط الإضافات 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 الإضافة

- ينقر على "5GB Data Boost"
- يؤكد الشراء بمبلغ £5.00
- يلتقط النظام تفويض الدفع

4. يبدأ توفير الإضافة

يستدعي النظام `play_topup_charge_then_action.yaml` مع المتغيرات:

```
{
    "product_id": 43,  # 5 منتجGB Data Boost
    "customer_id": 123,
    "service_id": 1234,  # خدمة موجودة
    "access_token": "eyJ...",
    "data_gb": 5,  # من provisioning_json_vars
    "days": 7      # من provisioning_json_vars
}
```

الاختلاف الرئيسي عن الخدمة المستقلة:

- يتم تضمين `service_id` (الخدمة الموجودة لتعديلها)
- لا يتطلب المخزون
- لا يتم إنشاء الخدمة (يعدل الموجود)

استعراض برنامج توفير الإضافة

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 `service_uuid` الصحيح OCS لإضافة الرصيد إلى حساب
- يتحقق من وجود الخدمة وأنها نشطة
- يضمن أن الخدمة تنتمي إلى العميل

المهمة 2: شحن 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 هذه المهمة:

- يجد طريقة الدفع الافتراضية للعميل
- يشحن 5.00£ إلى البطاقة
- يسجل تكلفة الجملة 1.50£ لتتبع الهامش
- ينشئ معاملة مرتبطة بالخدمة
- يضيف إلى الفاتورة التالية
- يفشل التوفير إذا فشل الدفع

لماذا الشحن أولاً:

- لا يتم تسليم الرصيد حتى يتم تأكيد الدفع
- يمنع الاحتيال
- يطابق الدفع بتوفير الإضافة

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 هذه المهمة:

- يضيف 5 جيجابايت (5368709120 بايت) إلى الحساب
- تنتهي صلاحيتها في 168 ساعة (7 أيام)
- الوزن 20 (يتم استهلاكه أولاً - دفعة قبل الرصيد الشهري)

رصيد العميل بعد الإضافة:

- الرصيد الشهري الأصلي: 2 جيجابايت متبقية (تنتهي في 25 يومًا)
- الدفعة الجديدة: 5 جيجابايت (تنتهي في 7 أيام)
- الإجمالي المتاح: 7 جيجابايت
- ترتيب الاستخدام: يتم استهلاك الدفعة 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 هذه المهمة:

- يسجل £5.00 تكلفة للعميل
- يسجل £1.50 تكلفة جملة
- يربط المعاملة بالخدمة للتقارير

ملخص تدفق الإضافة الكامل

1. يختار العميل الإضافة من القائمة المصفاة

2. يتم تفويض الدفع وتحميله
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:

- تكلفة الخدمة الشهرية: 15£ تجزئة، 5£ جملة
- شراء الإضافة: 5£ تجزئة، 1.50£ جملة

التجديد التلقائي: الإضافات المتكررة

يمكن تعيين بعض الإضافات للتجديد التلقائي (خطط بيانات شهرية، اشتراكات، 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"
}
```

إنشاء البرنامج 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 هذه المهمة:

- ينشئ مهمة مجدولة في OCS
- ينفذ `Action_{{ product_slug }}` كل 30 يومًا
- يتم شحن العميل وإعادة تطبيق رصيد البيانات
- يستمر حتى يقوم العميل بالإلغاء

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.

مشغلات 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** - يصل العقد الثابت إلى تاريخ الانتهاء

تدفق إلغاء 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 "في انتظار 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 للبرنامج `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 هذه المهمة:

1. **تعطيل حساب OCS** - يوقف جميع الشحنات، ويمنع الاستخدام
2. **إزالة خطط العمل** - يلغي التجديدات التلقائية
3. **تحديث خدمة CRM** - الحالة "معطلة"، يتم تسجيل التاريخ
4. **تحرير المخزون** - يتم وضع علامة على SIM (بعد كـ "متقاعد"، متاحة لإعادة الاستخدام (التجديد)

5. بعد إلغاء التوفير

يقوم النظام بتنظيف:

- لم يعد العميل يرى الخدمة في بوابة الخدمة الذاتية
- تبقى الخدمة في CRM للتقارير التاريخية
- يتم الحفاظ على المعاملات والفواتير للمحاسبة
- يمكن تجديد 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` - المنتج معطل، تستمر الخدمات الحالية

تعطيل المنتجات:

- ضع علامة على المنتج كـ `enabled: false` لمنع الطلبات الجديدة
- تبقى الخدمات الحالية نشطة
- يمكن للعملاء تجديد/تعديل الخدمات الحالية
- مفيد لإيقاف المنتجات القديمة

إدارة المخزون

حالات المخزون:

- `New` - مخزون جديد، جاهز للتخصيص
- `In Stock` - متاح للتوفير
- `Assigned` - مرتبط بخدمة العميل
- `Decommissioned` - يمكن تجديده وإعادة استخدامه
- `Damaged` - يحتاج إلى إصلاح أو التخلص منه

إعادة استخدام المخزون:

بعد إلغاء التوفير:

- بطاقات SIM: تجديد ووضع علامة "في المخزون"
- أرقام الهواتف: الإفراج بعد فترة النقل (30 يومًا)
- المعدات: اختبار، تجديد، وضع علامة "مستخدمة"

قياسات التوفير

راقب:

- 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
- **مزامنة جهات الاتصال** - جهات اتصال العملاء تتم مزامنتها تلقائيًا مع Mailjet للحملات
- **قوالب البريد الإلكتروني** - أكثر من 10 نوع من قوالب البريد الإلكتروني المهيأة مسبقًا مع قوالب Mailjet القابلة للتخصيص
- **حملات التسويق** - حملات بريد إلكتروني مقسمة بناءً على بيانات العملاء
- **تسليم موثوق** - بنية تحتية احترافية للبريد الإلكتروني مع تتبع وتحليلات

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 **مفتاح API السري** و **المفتاح**
4. لصقها في `crm_config.yaml`

تكوين قالب البريد الإلكتروني

يستخدم OmniCRM **10** يتم **أنواع مختلفة من قوالب البريد الإلكتروني** للاتصالات الآلية. تكوين كل 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 ترسل:

- يقوم عميل جديد بالتسجيل عبر بوابة الرعاية الذاتية
- يقوم الموظف بإنشاء حساب عميل جديد
- يقوم العميل بتفعيل الخدمة للمرة الأولى

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 الرعاية الذاتية
- `{{ var:support_url }}` - رابط إلى صفحة الدعم

رسالة فاتورة العميل

ترسل عندما يتم إنشاء فاتورة وتكون جاهزة للدفع.

```
api_crmCommunicationCustomerInvoice:
  from_email: "billing@yourcompany.com"
  from_name: "Your Company Billing"
  template_id: 6759851
  subject: "Your Invoice - "
```

متى ترسل:

- يتم إنشاء فاتورة تلقائيًا لفترة الفوترة
- فاتورة يدوية تم إنشاؤها 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/تحميل PDF الفاتورة
- `{{ 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 ترسل:

- يوم (قابل للتكوين) X الفاتورة متأخرة بمقدار
- تذكير يدوي تم تفعيله بواسطة الموظف
- سير عمل تذكير تلقائي (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 ترسل:

- يقوم المسؤول بإنشاء مستخدم موظف جديد
- تم النقر على زر "إرسال رسالة ترحيب" في إدارة المستخدم

المتغيرات المتاحة في القالب:

- `{{ var:user_name }}` - الاسم الكامل لمستخدم الموظف
- `{{ var:email }}` - بريد مستخدم الموظف
- `{{ 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 كل��ة مروره.

```
api_crmCommunicationUserPasswordReset:
  from_email: "noreply@yourcompany.com"
  from_name: "Your Company Security"
  template_id: 5977511
  subject: "Password Reset Request"
```

متى ترسل:

- 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 أمنية:

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 ترسل:

- يكمل المستخدم بنجاح 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 ترسل:

- يغير المستخدم كلمة المرور من الملف الشخصي/الإعدادات
- يعيد المسؤول 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 ترسل:

- تم إنشاء حساب جديد (عميل أو موظف)
- يغير المستخدم عنوان بريده الإلكتروني
- يتطلب التحقق من البريد الإلكتروني لأسباب أمنية

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/مخصص خدمة العميل.

```
api_crmCommunicationsBalanceExpired:
  from_email: "support@yourcompany.com"
  from_name: "Your Company Support"
  template_id: 5977515
  subject: "Your Service Balance Has Expired"
```

متى ترسل:

- انتهاء صلاحية الرصيد المدفوع مسبقًا
- تجاوز تاريخ تجديد المخصص 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 }}` - "نقدي"، "صوت"، "بيانات"، إلخ.
- `{{ var:renewal_url }}` - رابط للتجديد/إعادة التعبئة

تنبيه انخفاض الرصيد

ترسل عندما ينخفض رصيد العميل دون العتبة المحددة.

```
api_crmCommunicationsBalanceLow:
  from_email: "support@yourcompany.com"
  from_name: "Your Company Support"
  template_id: 5977516
  subject: "Low Balance Alert"
```

متى ترسل:

- ينخفض الرصيد دون العتبة (على سبيل المثال، 20% متبقية)
- محدد في خطة الخدمة أو OCS
- مراقبة في الوقت الحقيقي تحفز التنبيه

المتغيرات المتاحة في القالب:

- `{{ var:customer_name }}`
- `{{ var:service_name }}`
- `{{ var:current_balance }}`
- `{{ var:threshold }}`
- `{{ var:balance_type }}`
- `{{ var:topup_url }}` - رابط لإضافة رصيد

إنشاء قوالب البريد الإلكتروني في 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 Mailjet بالسحب والإفلات أو محرر HTML 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 المتغيرات 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 القالب

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 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 لتفعيل التغييرات:

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

مزامنة جهات الاتصال

يتم مزامنة جميع جهات اتصال العملاء في OmniCRM تلقائيًا مع Mailjet.

ما الذي يتم مزامنته:

- اسم جهة الاتصال
- عنوان البريد الإلكتروني
- نوع جهة الاتصال (فواتير، تقنية، إلخ.)
- موقع العميل

- حقول مخصصة

تكرار المزامنة:

تتم مزامنة جهات الاتصال في الوقت الحقيقي عندما:

- يتم إنشاء عميل جديد
- تمت إضافة/تحديث جهة الاتصال
- تم تعديل تفاصيل العميل

استكشاف الأخطاء وإصلاحها

البريد الإلكتروني لا يرسل

- **السبب:** بيانات اعتماد API غير صالحة، حساب Mailjet معلق، أو معرف القالب خاطئ
- **الإصلاح:**
  - تحقق من `api_key` و `api_secret` في `crm_config.yaml`
  - تحقق من حالة حساب Mailjet والفوترة
  - تحقق من وجود معرف القالب في Mailjet
  - تحقق من سجلات API بحثًا عن الأخطاء

المتغيرات القالب لا تستبدل

- **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 API تمرر بيانات المتغير في استدعاء OmniCRM
  - 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

جهات الاتصال لا تتم مزامنتها مع Mailjet

- **السبب:** تجاوز حد معدل API أو عدم تشغيل خدمة المزامنة
- **الإصلاح:**
    - تحقق من حدود معدل API في Mailjet (200 استدعاء/دقيقة)
    - تحقق من أن خدمة OmniCRM-API قيد التشغيل
    - راجع سجلات المزامنة بحثًا عن الأخطاء
    - قم بتفعيل المزامنة يدويًا للاختبار

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 الرعاية الذاتية
- المراقبة والقياسات - قياسات واجهة برمجة تطبيقات Mailjet، تتبع تسليم البريد الإلكتروني، ومراقبة الأداء

قراءة إضافية

- وثائق Mailjet: <https://dev.mailjet.com/>
- مرجع واجهة برمجة تطبيقات Mailjet: <https://dev.mailjet.com/email/reference/>

سجل التغييرات

هذا يحتوي على آخر 50 تغيير تم إجراؤه على مجموعة برامج OmniCRM أو اعتماداته.

ملاحظة: هذا لا يتتبع التغييرات في إعدادات العملاء الفرديين.

المراقبة والقياسات

يوفر واجهة برمجة التطبيقات OmniCRM قياسات شاملة تعتمد على Prometheus لمراقبة أداء التطبيق وعمليات الأعمال والتكاملات الخارجية. يتم عرض جميع القياسات عند نقطة النهاية `/crm/metrics` بتنسيق عرض Prometheus.

نظرة عامة

يتتبع نظام القياسات:

- **عمليات التوفير** - تنفيذ الوظائف، المدة، معدلات النجاح/الفشل
- **أداء قاعدة البيانات** - أوقات الاستعلام، صحة مجموعة الاتصالات
- **التكاملات الخارجية** - مكالمات واجهة برمجة التطبيقات OCS/CGRateS، Stripe، Mailjet
- **وظائف الخلفية** - تنفيذ المهام غير المتزامنة والأداء
- **طلبات HTTP** - استخدام نقاط نهاية واجهة برمجة التطبيقات وأوقات الاستجابة (تم إنشاؤها تلقائيًا)

نقطة نهاية القياسات

URL: `http://your-omnicrm-api:5000/crm/metrics`

التنسيق: تنسيق عرض Prometheus

المصادقة: نقطة نهاية القياسات متاحة للجمهور لجمع البيانات بواسطة Prometheus. في الإنتاج، يُوصى بتقييد الوصول باستخدام قواعد جدار الحماية أو مصادقة الوكيل العكسي.

فئات القياسات

1. قياسات التوفير

تتبع قياسات التوفير تنفيذ كتب اللعب Ansible التي توفر الخدمات، وتدير المخزون، وتقوم بتكوين الأنظمة الخارجية. راجع نظام التوفير و كتب اللعب 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 عندما تكتمل وظائف التوفير بحالتها النهائية.

omnicrm_provision_job_duration_seconds

النوع: هيستوجرام

التسميات:

- `playbook` - اسم كتاب اللعب Ansible المنفذ
- `status` - حالة إكمال الوظيفة: `success`, `failed`

الأقسام: `[1, 5, 10, 30, 60, 120, 180, 300, 600]` ثوانٍ (1s إلى 10min)

الوصف: الوقت المستغرق لإكمال وظائف التوفير. يسجل مدة تنفيذ كتاب اللعب بالكامل من البداية إلى النهاية.

omnicrm_provision_jobs_active

النوع: مقياس

الوصف: عدد وظائف التوفير التي تعمل حاليًا. يتم زيادته عند بدء وظيفة، ويتم إنقاصه عند اكتمالها.

omnicrm_provision_tasks_total

النوع: عداد

التسميات:

- `playbook` - اسم كتاب اللعب Ansible
- `status` - نتيجة المهمة: `ok`, `failed`, `ignored`

المنفذة ضمن كتب اللعب. يتم زيادته لكل مهمة فردية Ansible **الوصف:** العدد الإجمالي لمهام (نجاح أو فشل) تكتمل.

omnicrm_provision_errors_total

النوع: عداد

التسميات:

- `error_type` - نوع الخطأ: `fatal`, `task_failed`, `timeout`
- `playbook` - اسم كتاب اللعب Ansible

الوصف: العدد الإجمالي لأخطاء التوفير حسب النوع. يتم زيادته عندما تفشل مهام التوفير أو تحدث أخطاء قاتلة أثناء التنفيذ.

---

2. قياسات قاعدة البيانات

OmniCRM تراقب قياسات قاعدة البيانات أداء الاستعلام وصحة مجموعة الاتصالات. يستخدم مع أدوات قياس تلقائية. راجع معمارية النظام للحصول على تفاصيل نموذج SQLAlchemy البيانات.

omnicrm_db_query_duration_seconds

النوع: هيستوجرام

التسميات:

- `operation` - نوع عملية SQL: `SELECT`, `INSERT`, `UPDATE`, `DELETE`
- `table` - اسم جدول قاعدة البيانات

الأقسام: `[0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0, 5.0, 10.0]` ثوانٍ

الوصف: وقت تنفيذ استعلام قاعدة البيانات. يتم تتبعه تلقائيًا عبر مستمعي أحداث SQLAlchemy على كل استعلام.

omnicrm_db_pool_size

النوع: مقياس

الوصف: الحجم الحالي لمجموعة اتصالات قاعدة البيانات. العدد الإجمالي للاتصالات في المجموعة (المستعارة والمتاحة).

omnicrm_db_pool_overflow

النوع: مقياس

الوصف: تجاوز الحجم الحالي لمجموعة اتصالات قاعدة البيانات. عدد الاتصالات التي تم إنشاؤها تتجاوز حد حجم المجموعة العادي.

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 ولكنها غير مستخدمة حاليًا في الكود الحالي (محجوزة للاستخدام المستقبلي)

---

3. قياسات OCS/CGRateS

تتبع قياسات 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.

الأقسام: `[0.1, 0.25, 0.5, 1.0, 2.0, 5.0, 10.0, 30.0]` ثوانٍ

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`

الوصف: العدد الإجمالي لاستعلامات الرصيد إلى OCS. يُستخدم لتتبع استخدام عمليات استعلام 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. قياسات دفع Stripe

تتبع قياسات Stripe للحصول على طرق الدفع و دليل نظام الدفع. راجع عمليات معالجة الدفع. 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 دفع.

omnicrm_stripe_api_duration_seconds

النوع: هيستوجرام

التسميات:

- `operation` - نوع عملية Stripe

الأقسام: `[0.1, 0.5, 1.0, 2.0, 5.0, 10.0, 30.0]` ثوانٍ

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 بالسينت. مفيد لتتبع حجم المعاملات والإيرادات.

omnicrm_stripe_payment_failures_total

النوع: ع��اد

التسميات:

- `reason` - سبب الفشل: `card_declined`, `insufficient_funds`, `expired_card`, إلخ.

الوصف: العدد الإجمالي لفشل مدفوعات Stripe مصنفة حسب رمز الرفض.

---

5. قياسات بريد Mailjet

تتبع قياسات Mailjet تكامل راجع .Mailjet عمليات تسليم البريد الإلكتروني 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

الأقسام: `[0.1, 0.5, 1.0, 2.0, 5.0, 10.0, 30.0]` ثوانٍ

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 الرسائل المرسلة.

---

6. قياسات وظائف الخلفية

تتبع قياسات وظائف الخلفية العمليات غير المتزامنة مثل سلاسل كتب اللعب 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 إنقاصه عند اكتمال الوظيفة.

omnicrm_background_job_duration_seconds

النوع: هيستوجرام

التسميات:

- `job_type` - فئة الوظيفة
- `status` - نتيجة الوظيفة: `success`, `failed`

الأقسام: `[1, 5, 10, 30, 60, 120, 180, 300, 600, 1800, 3600]` ثوانٍ (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. قياسات HTTP لـ Flask (تم إنشاؤها تلقائيًا)

تُنشأ هذه القياسات تلقائيًا بواسطة مكتبة `prometheus-flask-exporter` وتتبع 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 حالة HTTP

الوصف: مدة طلب HTTP لجميع نقاط نهاية واجهة برمجة التطبيقات. تم قياسها تلقائيًا.

flask_http_request_total

النوع: عداد

التسميات:

- `method` - طريقة HTTP
- `endpoint` - اسم مسار Flask
- `status` - رمز حالة 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. تشير إلى الأخطاء أو الأخطاء غير المتوقعة.

---

8. قياسات أخطاء واجهة برمجة التطبيقات (محجوزة)

تم تعريف هذه القياسات ولكنها غير مُعَدّة حاليًا. تم حجزها للاستخدام المستقبلي.

omnicrm_api_errors_total

النوع: عداد

التسميات:

- `endpoint` - نقطة نهاية واجهة برمجة التطبيقات
- `error_type` - فئة الخطأ
- `status_code` - رمز حالة HTTP

الحالة: محددة ولكنها غير مستخدمة بنشاط

omnicrm_api_auth_failures_total

النوع: عداد

التسميات:

- `auth_method` - طريقة المصادقة: `jwt`, `api_key`, `ip_whitelist`
- `reason` - سبب الفشل

الحالة: محددة ولكنها غير مستخدمة بنشاط

---

9. معلومات قياس التطبيق

omnicrm_api

النوع: معلومات

التسميات:

- `version` - سلسلة إصدار واجهة برم��ة التطبيقات
- `environment` - اسم البيئة: `production`, `staging`, `development`

الوصف: قياس معلومات واجهة برمجة التطبيقات OmniCRM. يتم تعيينه مرة واحدة عند بدء 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'
```

تنبيهات نموذجية

معدل فشل التوفير مرتفع

```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 وظائ 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 البيانات البطيئة

```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 البيانات البطيئة"
    description: "هو 99 وقت استعلام النسبة المئوية {{ $value }}s"
```

متوقفة 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 الـتط��يقات"
    description: "{{ $value | humanizePercentage }} من مكالـمات
تفشل OCS واجهة برمجة الـتطبـيقات"
```

مشكلات دفع 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 الـدفع في الـثـانـية"
```

أفضل الممارسات

استراتيجية المراقبة

1. **إعداد تنبيهات أساسية** - تكوين تنبيهات للقياسات الحرجة:
   - معدل فشل التوفير > 10%
   - استنفاد مجموعة اتصالات قاعدة البيانات
   - فشل واجهة برمجة التطبيقات OCS/CGRateS

- أخطاء معالجة دفع Stripe

2. تتبع **قياسات الأعمال** - مراقبة مؤشرات الأداء الرئيسية التشغيلية:
   - الإيرادات المعالجة عبر Stripe
   - معدل التوفير
   - 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 القياسات للتنبؤ:
   - حجم مجموعة اتصالات قاعدة البيانات
   - توسيع عمال وظائف الخلفية
   - سعة خادم واجهة برمجة التطبيقات

الاحتفاظ بالقياسات

توصيات احتفاظ Prometheus:

- **15 يومًا** - قياسات عالية الدقة (فاصل جمع 15 ثانية)
- **90 يومًا** - قياسات منخفضة العينات (تجميع 5 دقائق)
- **سنة واحدة** - قياسات مجمعة على المدى الطويل (تجميع ساعة واحدة)

استخدم Thanos أو Cortex أو VictoriaMetrics للتخزين على المدى الطويل والاستعلام العالمي.

استكشاف الأخطاء وإصلاحها

القياسات غير الظاهرة

فقط للمصادر الداخلية (غير العامة). يمكنك `/metrics` بشكل افتراضي، يتم عرض نقطة النهاية الخاص بك إذا لزم الأمر nginx تغيير ذلك في تكوين.

تحقق من نقطة نهاية القياسات:

```
curl http://localhost:5000/crm/metrics
```

تحقق مما إذا كان Prometheus يمكنه جمع البيانات:

```
# تحقق من صفحة أهداف Prometheus
http://prometheus:9090/targets
```

القياسات المحددة المفقودة

بعض القياسات تُنشأ فقط بعد استخدامها لأول مرة:

- تظهر قياسات التوفير بعد أول وظيفة توفير
- تظهر قياسات Stripe بعد أول دفعة
- تظهر قياسات OCS بعد أول عملية فوترة

مشكلات الكاردينالية العالية

إذا كان Prometheus بطيئًا أو يستهلك ذاكرة مفرطة، تحقق من التسميات ذات الكاردينالية العالية:

```
# عد السلاسل الزمنية الفريدة لكل قياس
count by (__name__) ({__name__=~".+"})
```

قد تشير القياسات التي تحتوي على أكثر من 10,000 سلسلة إلى مشكلة في الكاردينالية.

ملخص القياسات

إجمالي القياسات: 34 (تم إنشاؤها تلقائيًا Flask 31 مخصص + 3 قياسات)

حسب النوع:

- **عدادات:** 20

- **مقاييس:** 5
- **هيستوجرامات:** 8
- **معلومات:** 1

حسب الفئة:

- التوفير: 5 قياسا
- قاعدة البيانات: 5 قياسات
- OCS/CGRateS: 5 قياسات
- Stripe: 5 قياسات
- Mailjet: 5 قياسات
- وظائف الخلفية: 4 قياسات
- HTTP/Flask: 3 قياسات
- معلومات التطبيق: 1 قياس
- محجوز (للاستخدام المستقبلي): 2 قياسات

الوثائق ذات الصلة

- معمارية النظام - تصميم النظام العام وعلاقات المكونات
- نظام التوفير - سير عمل التوفير وتنفيذ الوظائف
- كتب اللعب Ansible - هيكل كتاب اللعب والتنفيذ
- وثائق واجهة برمجة التطبيقات - نقاط نهاية واجهة برمجة التطبيقات والمصادقة
- نظرة عامة على الفوترة - نظام الفوترة والشحن
- دليل نظام الدفع Stripe ومعالجة الدفع - 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). تمكّن طرق الدفع من الفوترة التلقائية للخدمات، والمدفوعات لمرة واحدة، والرسوم المتكررة دون تخزين بيانات البطاقة الحساسة في OmniCRM.

انظر أيضًا: `دليل نظام الدفع <payment_system_guide>`, `نظرة عامة على الفوترة <billing_overview>`, `معالجة الدفع <payments_process>`, `الفواتير <payments_invoices>`.

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** - دعم طرق الدفع من Stripe وPayPal
- **بطاقات متعددة** - ي��كن 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 العناية الذاتية <self_care_portal>`
- **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)

لا يتم تخزينها في OmniCRM:

فقط بتخزين OmniCRM تفاصيل البطاقة يتم ترميزها بواسطة بائعي الدفع وتخزينها بأمان. يقوم:

- 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 قبل البائع

الوصول إلى طرق الدفع

من صفحة العميل:

1. انتقل إلى **العملاء** ← **[اختر عميلًا]**
2. انقر على علامة التبويب **الفوترة**
3. انتقل إلى قسم **طرق الدفع**

أو مباشرةً:

من لوحة معلومات البطاقات التي على وشك الانتهاء:

عرض جميع العملاء الذين لديهم بطاقات على وشك الانتهاء:

يعرض هذا قائمة شاملة للبطاقات التي ستنتهي صلاحيتها خلال 60 يومًا القادمة.

قائمة طرق الدفع

تعرض جدول طرق الدفع جميع البطاقات المخزنة لعميل:

وصف الأعمدة:

- **اللقب** - اسم ودود للبطاقة (مثل "بطاقة شخصية"، "فيزا العمل")
- **المصدر** - علامة البطاقة وآخر 4 أرقام
- **انتهاء الصلاحية** (بتنسيق MM/YYYY) - شهر/سنة انتهاء الصلاحية
- **تاريخ الإضافة** - تاريخ إضافة البطاقة إلى الحساب
- **افتراضي** - علامة تحقق تشير إلى طريقة الدفع الافتراضية للرسوم التلقائية

الإجراءات لكل بطاقة:

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

- **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

يظهر 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 الفوترة

الحقول الاختيارية:

- **لقب البطاقة** - تسمية ودية للتمييز بين البطاقات

الأمان:

- يتم إدخال تفاصيل البطاقة مباشرة في إطار آمن مستضاف من قبل البائع (Stripe Elements / PayPal SDK)
- لا يرى OmniCRM أبدًا أو يخزن أرقام البطاقة الكاملة
- يتم التعامل مع الامتثال لمعيار PCI DSS بواسطة بائع الدفع
- تمنع التحقق في الوقت الحقيقي أرقام البطاقة غير الصالحة

الخطوة 2: الإرسال والترميز

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. **الترميز:**
   - يتم إرسال تفاصيل البطاقة مباشرة إلى بائع الدفع (ليس إلى OmniCRM)
   - يقوم البائع بإنشاء رمز آمن (مثل `pm_1A2B3C4D` لـ Stripe)
   - يتم إرجاع الرمز إلى OmniCRM
3. **معالجة الخادم:**
   - يقوم OmniCRM بحفظ الرمز في سجل العميل مع معرف البائع
   - يخزن آخر 4 أرقام، العلامة، تاريخ الانتهاء، واسم البائع للعرض
   - لا تلمس OmniCRM أبدًا رقم البطاقة الكامل خوادم

الخطوة 3: التأكيد

تظهر رسالة النجاح:

> تم إضافة بطاقة Visa إلى حسابك بـ 1234 الخاصة بك التي تنتهي.

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
- إعادة الشحن والتعبئة
- المعاملات لمرة واحدة (ما لم يتم تحديد خلاف ذلك)

لتغيير الافتراضي:

1. ابحث عن البطاقة التي تريد تعيينها كافتراضية في جدول طرق الدفع
2. انقر على **قائمة الإجراءات (⋮)** بجوار البطاقة
3. اختر **"تعيين كافتراضي"**
4. تظهر رسالة تأكيد

   بطاقة Visa التي تنتهي بـ 5678 هي الآن طريقة الدفع الافتراضية الخاصة بك.

تتحرك علامة التحقق إلى البطاقة المحددة حديثًا.

مؤشر بصري:

تظهر البطاقات الافتراضية:

في عمود الافتراضي، عادةً مع شارة علامة تحقق خضراء.

حذف طريقة دفع

إزالة البطاقات التي انتهت صلاحيتها، أو فقدت، أو لم تعد مطلوبة.

الخطوة 1: بدء الحذف

1. ابحث عن البطاقة المراد حذفها في جدول طرق الدفع
2. انقر على **قائمة الإجراءات (⋮)**
3. اختر **"حذف"**

الخطوة 2: تأكيد الحذف

تظهر نافذة تأكيد:

> هل أنت متأكد أنك تريد حذف هذه الطريقة للدفع؟
>
> البطاقة: Visa 12/2026 :تاريخ الانتهاء: 1234 التي تنتهي بـ
>
> ⚠ تحذير: إذا كانت هذه هي طريقة الدفع الوحيدة لديك، ستحتاج إلى إضافة واحدة جديدة للاستمرار في استخدام الخدمات التي تتطلب الفوترة التلقائية.

> [حذف طريقة الدفع] [إلغاء]

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 الافتراضي إذا كانت هناك بطاقات أخرى موجودة** - قم بتعيين بطاقة مختلفة كافتراضية أولاً
- **تحذير عند حذف آخر بطاقة** - قد يتم تعليق الخدمات التي تتطلب الدفع
- **لا يمكن التراجع** - الحذف دائم؛ يجب على العميل إعادة إضافة البطاقة 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:

> العميل بطاقة انتهاء الصلاحية 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 منتهية الصلاحية تحديث

الفلاتر:

- **نطاق انتهاء الصلاحية** - الأيام 30/60/90 القادمة أو منتهية الصلاحية بالفعل
- **نوع العميل** - فردي مقابل أعمال
- **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 انتهاء صلاحية البطاقة

يمكن للعملاء النقر على رابط في البريد الإلكتروني لتحديث طريقة الدفع الخاصة بهم عبر بوابة العناية الذاتية.

متغيرات قالب البريد الإلكتروني:

تتلقى قوالب Mailjet:

- اسم العميل
- علامة البطاقة وآخر 4 أرقام
- تاريخ انتهاء الصلاحية
- رابط إلى صفحة طرق الدفع في العناية 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 العناية 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 الم��تهية
6. يؤكد مع العميل

تحذير

لا تطلب أبدًا من العملاء إرسال تفاصيل البطاقة عبر البريد الإلكتروني أو الرسائل النصية. استخدم دائمًا:

- بوابة العناية الذاتية الآمنة للخدمة الذاتية
- الهاتف مع الموظف الذي يدخل التفاصيل مباشرة في النظام
- شخصيًا في موقع البيع بالتجزئة

ماذا يحدث عند انتهاء صلاحية البطاقات

عندما تصل بطاقة الدفع إلى تاريخ انتهاء صلاحيتها ولا يتم تحديثها:

التأثيرات الفورية:

1. **تفشل المدفوعات التلقائية**
    - يرفض بائع الدفع المعاملات مع البطاقات المنتهية
    - تفشل تجديدات الخدمة الشهرية في المعالجة
    - تفشل إعادة الشحن 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 العناية الذاتية
3. **تغييرات حالة الخدمة**
    - **الخدمات المدفوعة لاحقًا** - قد تستمر مؤقتًا مع رصيد مستحق
    - **الخدمات المدفوعة مسبقًا** - تعليق الخدمة عند نفاد الرصيد
    - **الخدمات التي تتجدد 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
- 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 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 للنسخ الاحتياطي
- 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. إضافة طريقة دفع جديدة صالحة
2. تعيينها كافتراضية
3. دفع الرصيد المستحق (إن وجد)
4. الاتصال بالدعم لإعادة تفعيل الخدمة
5. استعادة الخدمة في غضون دقائق إلى ساعات

أمان طريقة الدفع

الترميز

يستخدم OmniCRM ترميز البائع لضمان الأمان:

1. **يدخل العميل البطاقة** → يتم إرسالها مباشرة إلى خوادم بائع الدفع
2. **تحقق البائع ويقوم بالترميز** → 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 يخزنه 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 يخزنه OmniCRM:

- رقم البطاقة الكامل
- رمز CVV/CVC
- بيانات الشريط المغناطيسي
- أرقام PIN

الامتثال لمعيار PCI

من خلال استخدام نماذج الدفع المستضافة من قبل البائع:

- أبدًا OmniCRM **مخفض** - بيانات البطاقة لا تلمس خوادم **نطاق PCI**

- **حقول مستضافة من قبل البائع** - يتم إدخال البطاقة في إطار آمن للبائع
- **لا تخزين للبطاقات** - يتم استخدام الرموز بدلاً من بيانات البطاقة الخام
- **نقل آمن** - جميع الاتصالات عبر HTTPS/TLS

للحصول على تفاصيل أمان بائع الدفع `<payment_system_guide>` انظر `دليل نظام الدفع`.

سير العمل الشائعة

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. يُطلب منه إضافة طريقة الدفع أثناء الخروج
4. يدخل تفاصيل البطاقة في نافذة Stripe
5. يتم ترميز 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 (60 يومًا قبل انتهاء الصلاحية)
2. يقوم بتسجيل الدخول إلى بوابة العناية الذاتية
3. ينتقل إلى **الفوترة → طرق الدفع**
4. يستعرض البطاقة الحالية التي ستنتهي صلاحيتها 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: الموظف 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، و��م استعادة الخدمة"

سير العمل 4: عميل أعمال لديه بطاقات متعددة

السيناريو: ترغب الشركة في بطاقات مختلفة لأغراض مختلفة

1. يضيف عميل الأعمال بطاقة رئيسية (Visa 1111 تنتهي بـ)
2. يعينها كافتراضية لرسوم الخدمة الشهرية
3. يضيف بطاقة احتياطية (Mastercard 2222 تنتهي بـ) لإعادة الشحن
4. يضيف بطاقة شراء (Amex 3333 تنتهي بـ) لمشتريات المعدات
5. عند إجراء إعادة الشحن، يختار Mastercard يدويًا أثناء الخروج
6. لا تزال Visa الافتراضية مستخدمة للفوترة التلقائية الشهرية

سير العمل 5: إدارة البطاقات التي على وشك الانتهاء (الإدارة)

السيناريو: إدارة انتهاء الصلاحية بشكل استباقي

1. ينتقل المسؤول إلى **الفوترة → البطاقات التي على وشك الانتهاء**
2. الفلاتر: "الأيام الـ 30 القادمة"

3. يرى 15 عميلًا لديهم بطاقات على وشك الانتهاء
4. يحدد الكل ← **"إرسال رسائل تذكير عبر البريد الإلكتروني"**
5. يرسل Mailjet رسائل بريد إلكتروني مخصصة لكل عميل
6. يقوم العملاء بتحديث بطاقاتهم عبر العناية الذاتية
7. يراجع المسؤول 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 طريقة 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 الموجودة في الملف

خطأ في إضافة طريقة الدفع" (خطأ عام)"

- **السبب:** خطأ في واجهة برمجة التطبيقات لبائع 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 حواجز الإعلانات)

لا يمكن حذف طريقة الدفع (زر معطل)

- **السبب:** محاولة حذف البطاقة الافتراضية، أو أنها البطاقة الوحيدة
- **الإصلاح:**
  - قم بتعيين بطاقة مختلفة كافتراضية أولاً
  - إذا كانت البطاقة الوحيدة، أضف بطاقة جديدة قبل الحذف

"تظهر البطاقة كمنتهية ولكنها ليست في قائمة "البطاقات التي على وشك الانتهاء

- **السبب:** انتهت صلاحية البطاقة مؤخرًا، لم يتم تحديث الذاكرة المؤقتة
- **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 لا تحمل

- **السبب:** عدم تحميل واجهة برمجة التطبيقات لبائع الدفع، مشكلة في مفتاح API، أو امتداد المتصفح يمنع التحميل
- **الإصلاح:**
  - تحقق من وحدة التحكم في المتصفح للحصول على أخطاء
  - قم بتعطيل حواجز الإعلانات وحماية التتبع
  - تحقق من تكوين بائع الدفع في إعدادات النظام
  - تأكد من تحميل برنامج SDK (تحقق من علامة 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 الصلاحي
  - 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 الملف للنسخ الاحتياطي
- قم بتحديث البطاقات التي على وشك الانتهاء قبل 30 يومًا أو أكثر من انتهاء الصلاحية
- احذف البطاقات القديمة/المنتهية لتجنب الارتباك
- استخدم ألقاب وصفية ("فيزا شخصية"، "أميكس العمل")
- 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£)

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 فشل الدفع لتحديد البطاقات المنتهية
- درب الموظفين على إجراءات التعامل الآمن مع البطاقات

الوثائق ذات الصلة

- `دليل نظام الدفع <payment_system_guide>` - مرجع كامل لواجهة برمجة التطبيقات للدفع والهندسة المعمارية
- `payments_process` - معالجة المدفوعات باستخدام طرق الدفع المخزنة
- `payments_invoices` - الدفع التلقائي للفواتير باستخدام البطاقة الافتراضية

- `features_topup_recharge` - نظام إعادة الشح�� باستخدام طرق الدفع
- `basics_payment` - مفاهيم الدفع والفوترة العامة
- `customer_care` - بوابة العناية الذاتية للعملاء لإدارة بطاقاتهم الخاصة

دليل واجهة برمجة تطبيقات نظام الدفع OmniCRM

نظرة عامة

يوفر نظام الدفع OmniCRM بنية تحتية شاملة لمعالجة المدفوعات **غير مرتبطة بالبائع**. يدعم ولكن الهيكلية المودولارية تسمح بالتكامل مع أي مزود دفع، PayPalو Stripe اليوم كل من (Square، Adyen، Braintree، إلخ) دون الحاجة لتغيير كود التطبيق.

يغطي هذا المستند جميع واجهات برمجة التطبيقات الخاصة بالمدفوعات وسير العمل المتاحة في النظام.

> **تكامل Playbook Ansible**: لتنفيذ واجهات برمجة التطبيقات هذه في Playbooks التزويد، راجع **Playbooks التحصيل والمدفوعات من**

جدول المحتويات

1. الهيكلية المودولارية
2. المفاهيم الأساسية
3. المستندات المالية
4. واجهات برمجة تطبيقات طرق الدفع
5. واجهات برمجة تطبيقات تدفق الدفع
6. واجهات برمجة تطبيقات المحفظة
7. ملخص مرجع واجهة برمجة التطبيقات
8. حالات الاستخدام الشائعة

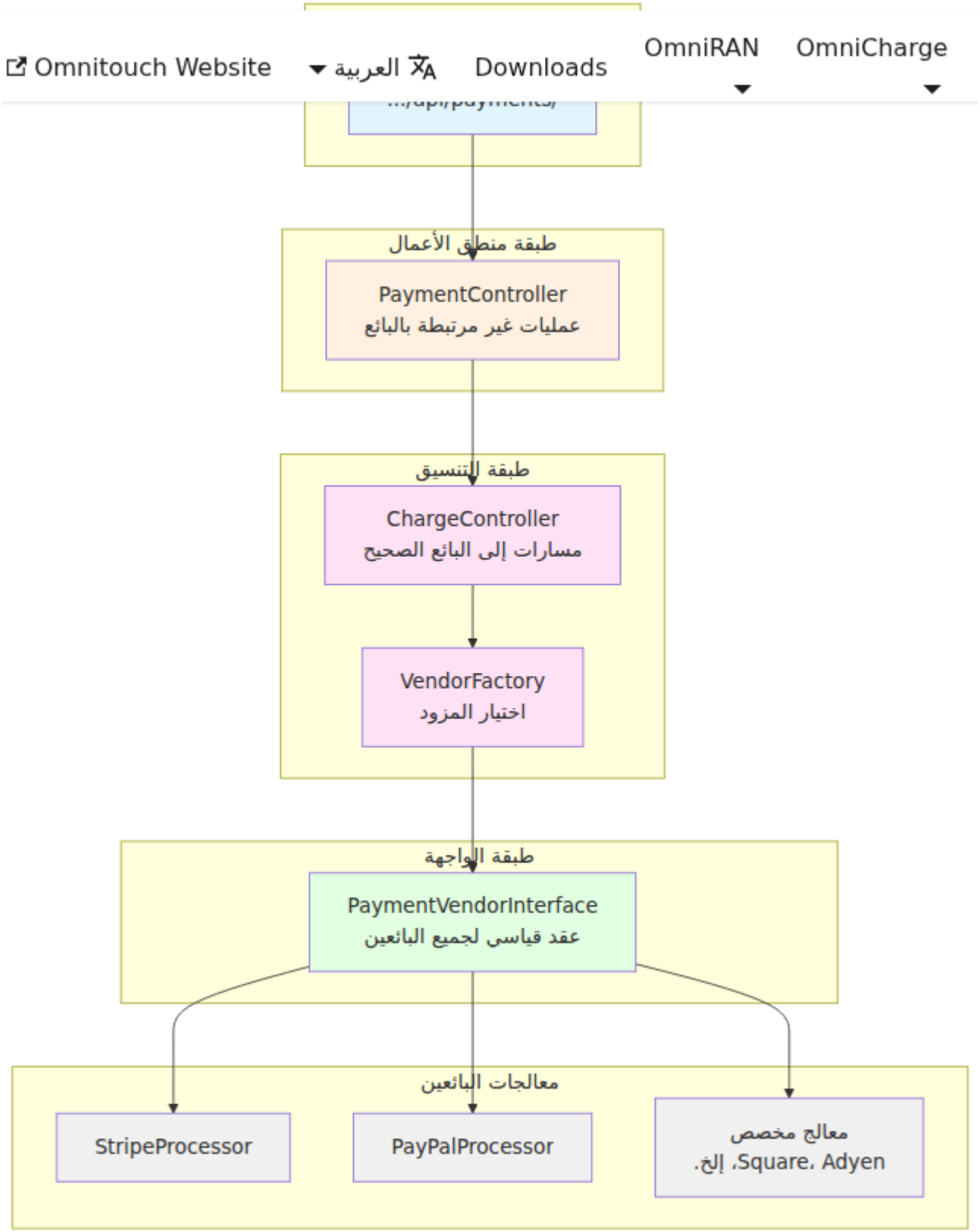
الهيكلية المودولارية

لماذا غير مرتبطة بالبائع؟

يستخدم النظام **طبقات التجريد** لفصل منطق الأعمال عن تفاصيل مزود الدفع. وهذا يعني:

- إضافة مزودي دفع جدد دون الحاجة لتعديل كود التطبيق
- تبديل البائعين دون الحاجة لترحيل قاعدة البيانات
- دعم عدة بائعين في نفس الوقت
- واجهة برمجة تطبيقات متسقة بغض النظر عن مزود الخلفية

طبقات الهيكلية



إضافة بائع دفع جديد

لإضافة مزود جديد (على سبيل المثال، Square، Adyen، Braintree)، اتصل بفريق OmniCRM الفني الخاص بك.

للحصول على تفاصيل التكوين حول البائعين الحاليين (Stripe، PayPal)، راجع **تكوين البائع**.

نظرة عامة على العملية:

الخطوات:

1. **تنفيذ فئة المعالج** مع `PaymentVendorInterface` القياسية (تفويض، التقاط، شحن، استرداد، إفراج)
2. **تسجيل المعالج** في VendorFactory مع اسم البائع
3. **اختبار التكامل** مع بيئة اختبار البائع
4. **نشر** - تدعم جميع واجهات برمجة التطبيقات الحالية البائع الجديد تلقائيًا

النتيجة: بمجرد النشر، يعمل البائع 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 الخاصة بالدفع البائع الجديد تلقائيًا دون الحاجة لتغييرات في كود التطبيق.

وقت التكامل: عادةً ما يستغرق من 1 إلى 2 يومًا لمعالج بائع جدي.

تكوين البائع

تكوين Stripe

تعتبر Stripe 3 هي البائع الرئيسي للدفع، حيث توفر معالجة بطاقات مع دعمD 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:

- استخدم **مفاتيح الاختبار** (pk_test_/sk_test_) للتطوير
- استخدم **المفاتيح الحية** (pk_live_/sk_live_) للإنتاج فقط
- **لا تقم أبدًا** بإيداع مفاتيح واجهة برمجة التطبيقات في التحكم في الإصدارات

2. تكوين الخلفية

أضف إلى `crm_config.yaml`:

```
payment_vendors:
  stripe:
    api_key: "sk_live_YOUR_SECRET_KEY_HERE"
    publishable_key: "pk_live_YOUR_PUBLISHABLE_KEY_HERE"
```

أو عبر متغيرات البيئة:

```
export STRIPE_SECRET_KEY="sk_live_YOUR_SECRET_KEY_HERE"
export STRIPE_PUBLISHABLE_KEY="pk_live_YOUR_PUBLISHABLE_KEY_HERE"
```

3. تكوين الواجهة الأمامية

أضف إلى `.env`:

```
REACT_APP_STRIPE_PUBLISHABLE_KEY=pk_live_YOUR_PUBLISHABLE_KEY_HERE
```

ملاحظة أمنية: يجب أن يذهب المفتاح القابل للنشر فقط في الواجهة الأمامية. يجب **ألا** يتم الكشف عن المفتاح السري للمتصفح.

تكوين PayPal

يوفر PayPal تخزين بطاقات ومدفوعات حساب PayPal.

التكوين

أضف إلى `crm_config.yaml`:

```yaml
payment_vendors:
  paypal:
    client_id: "AXx_YOUR_CLIENT_ID_HERE"
    client_secret: "ELx_YOUR_CLIENT_SECRET_HERE"
    mode: "live"  # أو "sandbox" للاختبار
```

الامتثال لمعايير PCI

كيف يحافظ OmniCRM على الامتثال لمعايير PCI:

- يتم إدخال بيانات البطاقة مباشرة في iframes مستضافة من قبل البائع (Stripe Elements، PayPal Card Fields)
- OmniCRM **لا ترى أو تخزن** أرقام البطاقة الكاملة
- يتم تخزين طرق الدفع المرمزة فقط في قاعدة البيانات
- تم تقليل نطاق الامتثال لمعايير PCI لعملك

مقاييس معالجة الدفع

يوفر OmniCRM مقاييس شاملة لمراقبة عمل��ات معالجة الدفع. للحصول على تفاصيل كاملة حول مقاييس الدفع الخاصة بـ 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 بائع دفع دون الحاجة لترحيل:

حفظ بطاقة - **مثال** Square:

```
{
  "vendor": "square",
  "vendor_payment_method_id": "sq_card_abc123",
  "payment_type": "card"
}
```

لا تغييرات في الكود. لا ترحيل. يعمل ببساطة.

المفاهيم الأساسية

نماذج البيانات

PaymentMethod

تخزين غير مرتبط بالبائع لطرق دفع العملاء.

```
{
  "payment_method_id": 789,
  "customer_id": 123,
  "vendor": "stripe",                    // 'stripe', 'paypal', or
any added vendor
  "vendor_payment_method_id": "pm_xxx", // Vendor's internal ID
  "payment_type": "card",                // 'card', 'paypal',
'ach', etc.
  "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 (تجميد الأموال).

```
{
  "authorization_id": 301,
  "customer_id": 123,
  "payment_method_id": 789,
  "vendor": "stripe",                    // Which vendor
authorized
  "vendor_authorization_id": "auth_xxx", // Vendor's authorization
ID
  "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",     // Vendor's transaction
ID
  "amount": 200.00,
  "currency": "USD",
  "status": "succeeded",                 // 'succeeded', 'failed',
'refunded'
  "captured_at": "2025-12-27T10:30:00Z",
  "vendor_response": {},                 // Full vendor response
  "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",   // Links to related
object
  "reference_id": 103,
  "meta": {},
  "created_at": "2025-12-27T10:35:00Z"
}
```

المستندات المالية

فواتير العملاء لإنشاء قوالب الفواتير والتخصيص، راجع.

الفواتير

التعريف: الفاتورة هي مستند يحتوي على قائمة من **معاملات الخصم** (الرسوم). عند استدعاء واجهة برمجة التطبيقات لإنشاء فاتورة، يجب تقديم مصفوفة من معرفات معاملات الخصم. يتم "ربط" تلك المعاملات بالفاتورة عن طريق تعيين حقل `invoice_id` الخاص بها.

المعاملات للحصول على تفاصيل إدارة المعاملات، راجع.

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`، بالتتابع ضمن السنة التقويمية، وإعادة تعيينها في 1 يناير من كل عام.

المدفوعات على الفواتير

كيف تعمل المدفوعات: بمجرد إنشاء فاتورة من معاملات الخصم، يمكن تطبيق **الدفع** عن طريق إنشاء **معاملة ائتمانية** (تكلفة التجزئة السلبية) 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 الخصومات لحساب رصيد الفاتورة

حالة الفاتورة:

- **مدفوعة**: تساوي المدفوعات الإجمالية (الائتمانات) أو تتجاوز الخصومات 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 خصم (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) رسوم الخدمة :دفع للفاتورة",
  "payment_type": "stripe_capture",
  "payment_reference": "ch_xxxxx"
}

// النتيجة الصافية: 0$ رصيد ← تم وض علامة على الفاتورة كمدفوعة
```

بيانات الفاتورة التكميلية: بالإضافة إلى المعاملات الائتمانية، تخزن الفاتورة أيضًا حقول ملخص (`payment_reference`، `payment_type`، `payment_time`) لعمليات البحث السريعة. ومع ذلك:

- **الطريقة الأساسية**: المعاملات الائتمانية مرتبطة عبر `invoice_id` (مواصفات هايدن)
- **البيانات الوصفية الثانوية**: تخزن حقول الفاتورة ملخصًا لآخر/أهم دفعة
- **لعدة مدفوعات**: استعلام عن المعاملات الائتمانية للحصول على السجل الكامل

البيانات المالية

التعريف: تُظهر البيانات المالية جميع **الخصومات والائتمانات** من معاملات العميل خلال فترة محددة. هذا هو النوع الوحيد من المستندات حيث يتم عرض كل من الخصومات والائتمانات كعناصر قائمة معًا، مثل بيان البنك.

ملاحظات الائتمان

الغرض: تطبق فقط على الفواتير التي تم تطبيق المدفوعات عليها بالفعل. يجب أن يتم ذلك كجزء من عملية إلغاء الفاتورة.

العملية:

1. يتم إلغاء الفاتورة (مع معاملات الخصم المرتبطة بها)
2. يتم ربط أي مدفوعات تم تطبيقها بالفعل بملاحظة الائتمان التي تم إنشاؤها
3. يدخل رصيد العميل في الائتمان بمقدار المبلغ المعادل
4. يمكن بعد ذلك أن يكون رصيد ملاحظة الائتمان:
    - مطبق على فاتورة أخرى كدفع، أو
    - مسترد للعميل

منطق الاسترداد: إذا تم اختيار الاسترداد، يتم استدعاؤه بناءً على المعاملات الائتمانية المرتبطة (على سبيل المثال، استرداد Stripe إذا كان نوع المعاملة الائتمانية هو Stripe وكيف تم دفعها في الأصل).

واجهات برمجة تطبيقات طرق الدفع

عنوان URL الأساسي تستخدم جميع نقاط النهاية: `/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', or any added vendor
  "payment_token": "pm_xxxxx",           // One-time token from frontend SDK
  "is_default": false,                   // Set as default payment method?
  "card_nickname": "My Visa Card"        // Optional friendly name
}
```

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 ��رق 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 OK):

```
{
  "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 OK):

```
{
  "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 كطريقة افتراضية.

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 OK):

```
{
  "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 OK):

```
{
  "success": true,
  "message": "تم حذف طريقة الدفع بنجاح"
}
```

واجهات برمجة تطبيقات تدفق الدفع

يدعم النظام عدة تدفقات دفع اعتمادًا على حالة الاستخدام الخاصة بك.

1. الدفع المباشر (رسوم بسيطة)

حالة الاستخدام: دفع بسيط بخطوة 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
في المستقبل؟
  "metadata": {
    "order_id": "12345"
  }
}
```

الاستجابة (200 OK):

```
{
  "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 للدفع المتبقي.

نقطة النهاية: `POST /api/payments/invoice`

التدفق:

1. تحقق من رصيد المحفظة
2. استخدم أموال المحفظة أولاً
3. شحن البطاقة للقصور (إذا كان هناك أي)
4. ائتمان المحفظة بمبلغ البطاقة
5. خصم المحفظة للمبلغ الكامل للفاتورة

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 الفاتورة"
    }
  }'
```

الاستجابة (200 OK):

```
{
  "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 المبلغ
    "card_portion_used": 50.00,          // تم شحن البطاقة للمبلغ المتبقي
    "charged_amount": 50.00,
    "wallet_credited": 50.00,
    "wallet_debited": 200.00,
    "final_balance": 0.00,
    "payment_method_used": true
  }
}
```

3. حجز التفويض (تجميد الأموال)

حالة الاستخدام: حجز الأموال للتقاط لاحقًا (على سبيل المثال، حجوزات الفنادق، الإيجارات).

> **تنفيذ Playbook**: للحصول على أمثلة Playbook Ansible لتدفقات الدفع ذات المرحلتين، راجع نمط الالتزام ذو المرحلتين

الخطوة 1: إنشاء حجز تفويض

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. يحدث خصم المحفظة في 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 OK):

```
{
  "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 المبلغ
    "card_amount": 50.00,                // تم تفويض البطاقة لهذا المبلغ
    "message": "تم تفويض البطاقة بمبلغ 50 دولار (تعبئة المحفظة).
سيتم خصم المحفظة بمبلغ 200 دولار في وقت 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 للمبلغ الكامل للخدمة
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 - يتم التقاط 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 OK):

```
{
  "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": {                    // البطاقة قامت بتعبئة المحفظة
      "ledger_id": 401,
      "amount": 50.00
    },
    "wallet_debit": {                     // تم شحن الخدمة من المحفظة
      "ledger_id": 402,
      "amount": 200.00
    },
    "transaction": {                      // تم إنشاؤه إذا كانت الفاتورة=true
      "transaction_id": 7001
    },
    "invoice": {                          // تم إنشاؤه إذا كانت الفاتورة=true
      "invoice_id": 12345,
      "invoice_number": "INV-2025-000001"
    }
  }
}
```

الخطوة 3: إفراج عن 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 OK):

```
{
  "success": true,
  "message": "تم الإفراج عن 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 خصمها حتى وقت الالتقاط.

4. دفع التعبئة (مرحلتين مع توفير)

حالة الاستخدام: معالجة الدفع لتعبئة الخدمة التي تتطلب توفيرًا (على سبيل المثال، تفعيل نقطة اتصال/دونجل). إذا فشل التوفير، يتم الإفراج عن 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"
    }
  }'
```

الاستجابة (200 OK):

```
{
  "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/نقطة اتصال مجهولة
  }
}
```

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 مجهو��**:

```
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"
    }
  }'
```

الاستجابة (200 OK):

```
{
  "success": true,
  "message": "تمت معالجة دفع الإيجار",
  "data": {
    "transaction_id": "ch_xxxxx",
    "amount": 75.00,
    "payment_method_saved": false        // مجهول: لم يتم حفظ الطريقة
  }
}
```

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 الطلب:

```
{
  "transaction_id": "ch_xxxxx",          // معرف معاملة البائع
  "vendor": "stripe",                    // 'stripe', 'paypal',
إلخ.
  "amount": 50.00,                       // اختياري - استرداد كامل إذا
تم حذفه
  "reason": "customer_request"           // سبب الاسترداد الاختياري
}
```

الاستجابة (200 OK):

```
{
  "success": true,
  "message": "تمت معالجة الاسترداد",
  "data": {
    "refund_id": "re_xxxxx",
    "amount": 50.00,
    "status": "succeeded"
  }
}
```

واجهات برمجة تطبيقات المحفظة

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 OK):

```
{
  "customer_id": 123,
  "balance": 150.50,
  "currency": "USD"
}
```

الحصول على معلومات المحفظة

احصل على معلومات كاملة عن المحفظة والائتمان بما في ذلك إعدادات إعادة الشحن التلقائي.

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 OK):

```
{
  "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 دفع.

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 OK):

```
{
  "success": true,
  "payment": {
    "transaction_id": "ch_xxxxx",
    "amount": 100.00
  },
  "wallet": {
    "ledger_id": 503,
    "balance_after": 250.50
  },
  "message": "تم تعبئة المحفظة بمبلغ 100.00 دولار"
}
```

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 OK):

```
{
  "customer_id": 123,
  "count": 2,
  "transactions": [
    {
      "ledger_id": 501,
      "transaction_type": "credit",
      "amount": 100.00,
      "balance_before": 150.50,
      "balance_after": 250.50,
      "currency": "USD",
      "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,
      "currency": "USD",
      "description": "رسوم الخدمة",
      "reference_type": "service_charge",
      "reference_id": 789,
      "created_at": "2025-12-27T11:00:00Z"
    }
  ]
}
```

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

الوصف	نقطة النهاية	الطريقة
دفع مباشر (خطوة واحدة)	`/api/payments/charge`	POST
دفع فاتورة (المحفظة أولاً)	`/api/payments/invoice`	POST
تعبئة مع توفير	`/api/payments/topup`	POST
إنشاء حجز تفويض	`/api/payments/authorize/hold`	POST
التقاط التفويض	`/api/payments/capture/{id}`	POST
الإفراج عن التفويض	`/api/payments/release/{id}`	POST
دفع إيجار/طرف ثالث	`/api/payments/rental`	POST
استرداد الدفع	`/api/payments/refund`	POST

نقاط نهاية المحفظة

الوصف	نقطة النهاية	الطريقة
الحصول على رصيد المحفظة	`/api/wallet/balance?customer_id={id}`	GET
الحصول على معلومات المحفظة + الإعدادات	`/api/wallet/info?customer_id={id}`	GET
تعبئة المحفظة عبر طريقة الدفع	`/api/wallet/topup`	POST
الحصول على تاريخ المعاملات	`/api/wallet/transactions?customer_id={id}`	GET

نقاط نهاية محددة لـ PayPal

الطريقة	نقطة النهاية	الوصف
POST	`/api/payments/paypal/vault/setup-token`	إنشاء رمز إعداد لمجموعة PayPal Card Fields SDK
POST	`/api/payments/paypal/vault/finalize`	إنهاء خزينة 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 الفاتورة (المحفظة أولاً، البطاقة 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 التعبئة 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 → الإفراج عن التفويض (لا شحن)

حالة الاستخدام 3: دفع مجهول لنقطة اتصال

السيناريو: مستخدم مجهول يدفع للوصول إلى نقطة اتصال WiFi. لا سجل عميل، لا 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": "مقهى وسط المدينة"
    }
  }'
```

النتيجة:

- ✅ تمت معالجة الدفع
- ✅ تم تفعيل نقطة الاتصال
- ❌ لم يتم حفظ طريقة الدفع
- ❌ لم يتم إنشاء سجل عميل

حالة الاستخدام 4: حجز الفندق (حجز التفويض + التقاط)

السيناريو: حجز الأموال لتفويض حجز الفندق، التقاط عند تسجيل الوصول، الإفراج إذا تم الإلغاء.

```
# الخطوة 1: تسجي�ي 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 - 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": "إقامة الفندق - 3 ليالٍ",
      "description": "غرفة 302، 27-30 ديسمبر",
      "send_email": true
    }
  }'

# الخطوة 2b: إلغاء العميل - الإفراج عن 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": "بطاقة Visa للعمل"
  }'
```

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": "تمت تعبئة المحفظة بمبلغ 100.00 دولار"
}
```

معالجة الأخطاء

استجابات الأخطاء الشائعة

(400 طلب غ��ر 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 استرداد كامل 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:

- الاسترداد إلى طريقة الدفع الأصلية Stripe/PayPal تعالج
- خلال 5-10 أيام عمل PayPal يرى العميل الائتمان على بطاقته/حساب
- لا تذهب الأموال إلى المحفظة
- يتم الحفاظ على سجل تدقيق كامل في جدول `PaymentCapture`

متى تستخدم:

- ألغى العميل الطلب
- الخدمة لم تُسلم
- خطأ في الفوترة
- يطلب العميل صراحة الاسترداد إلى البطاقة

2. ائتمان إلى المحفظة

حالة الاستخدام: استرداد جزئي، ائتمان الخدمة، أو الاحتفاظ بالأموال للشراء في المستقبل.

ملاحظة: يتم التعامل مع الائتمانات في المحفظة عادةً داخليًا بواسطة النظام خلال سيناريوهات الخطأ. للحصول على ائتمانات المحفظة الي��وية، 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`

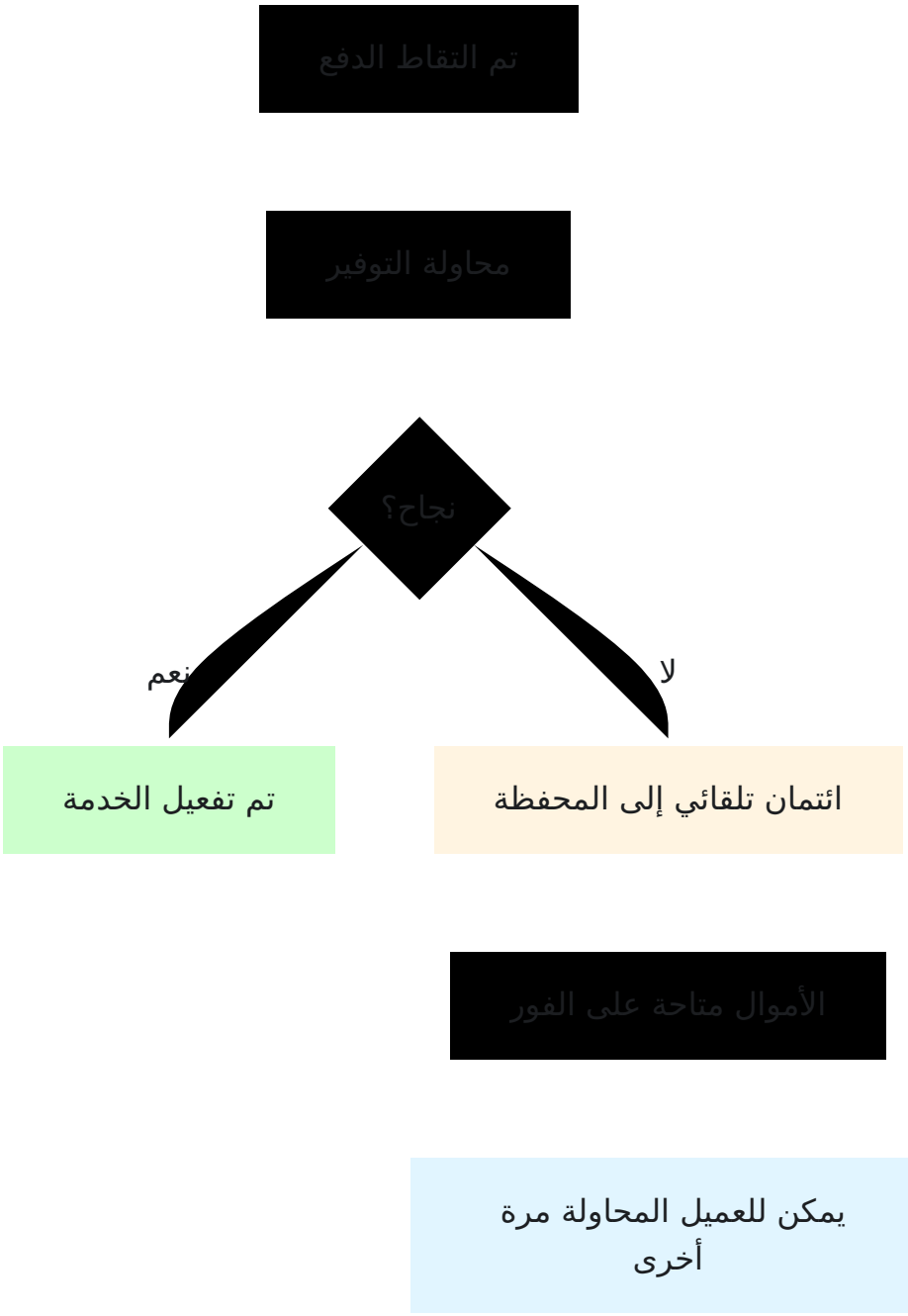
متى تستخدم:

- ائتمانات الخدمة (على سبيل المثال، تعويض عن التوقف)
- استردادات جزئية حيث سيعيد العميل الشراء
- تعويض عن الخطأ (فشل التوفير)
- ائتمانات ترويجية

استراتيجية استرداد هجينة

أفضل ممارسة: في سيناريوهات الخطأ خلال 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 المحاولة مرة أخرى على الفور
    - لا رسوم استرداد
    - تجربة مستخدم أفضل

توجيه المحفظة أولاً: شحنات بطاقات محسّنة

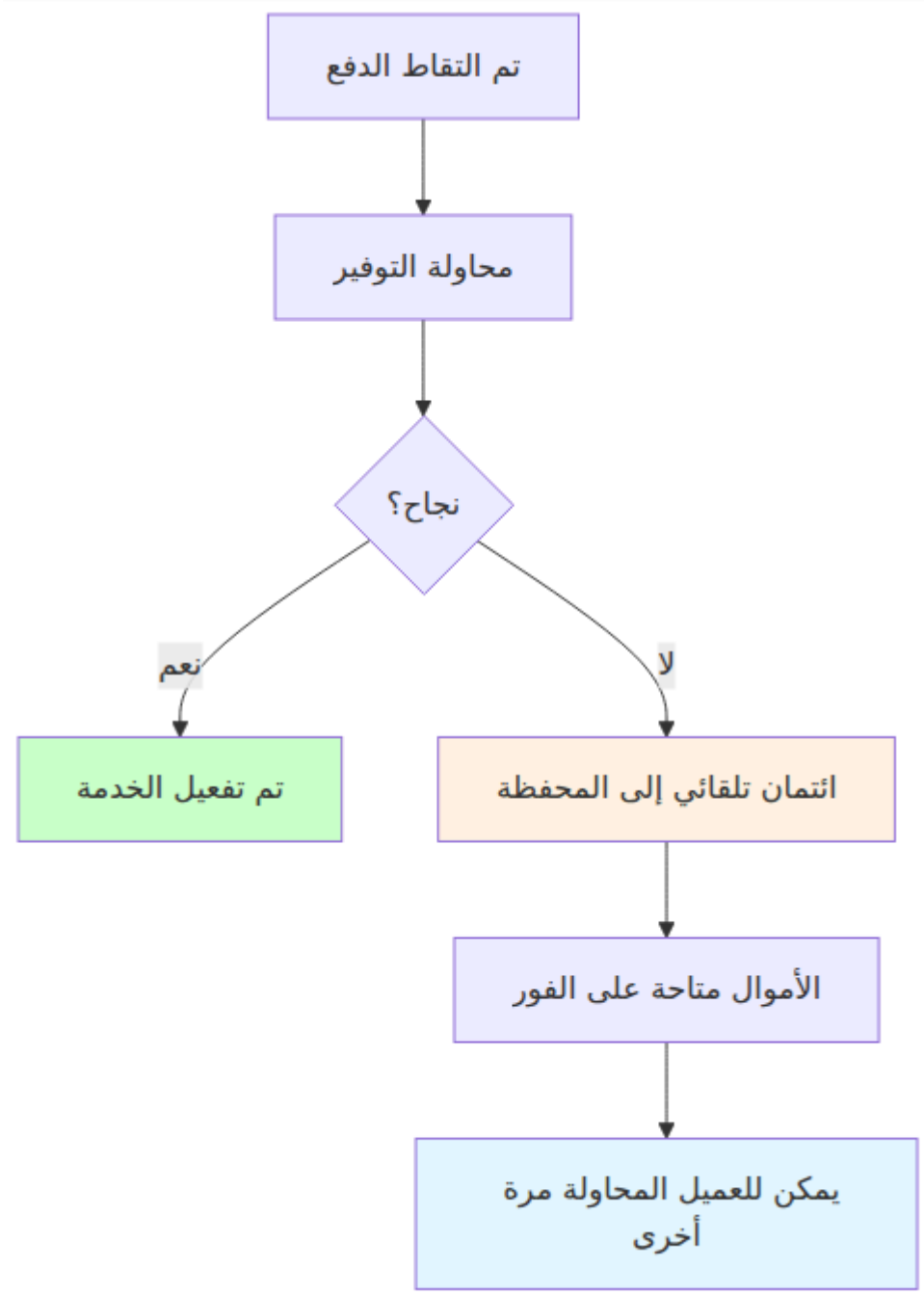
يقوم النظام **بشحن بطاقتك فقط للقصور**، وليس المبلغ الكامل، عندما يكون لديك رصيد في المحفظة.

مثال 1: رصيد محفظة 1 دولار + شراء 10 دولارات

السيناريو: لديك 1 دولار في محفظتك وتريد 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
    }
  }'
```

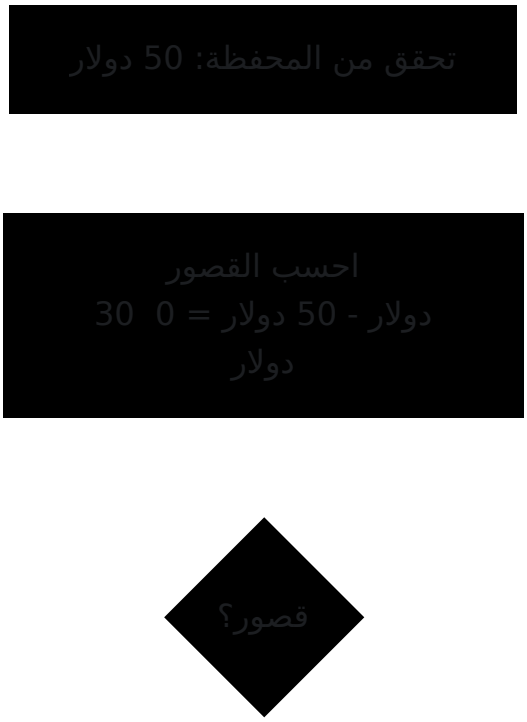
استجابة واجهة برمجة التطبيقات:

```
{
  "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,     // استخدم 1 دولار الموجود
    "card_portion_used": 9.00,       // تم شحن 9 دولارات فقط للقصور
    "charged_amount": 9.00,          // ← شحن البطاقة
    "wallet_credited": 9.00,         // ← تعبئة المحفظة
    "wallet_debited": 10.00,         // ← رسوم الخدمة
    "final_balance": 0.00
  }
}
```

مثال 2: رصيد محفظة 50 دولار + شراء 30 دولار

السيناريو: لديك 50 دولار في المحفظة وتكلفة الشراء 30 دولار.

تدفق المحفظة أولاً:



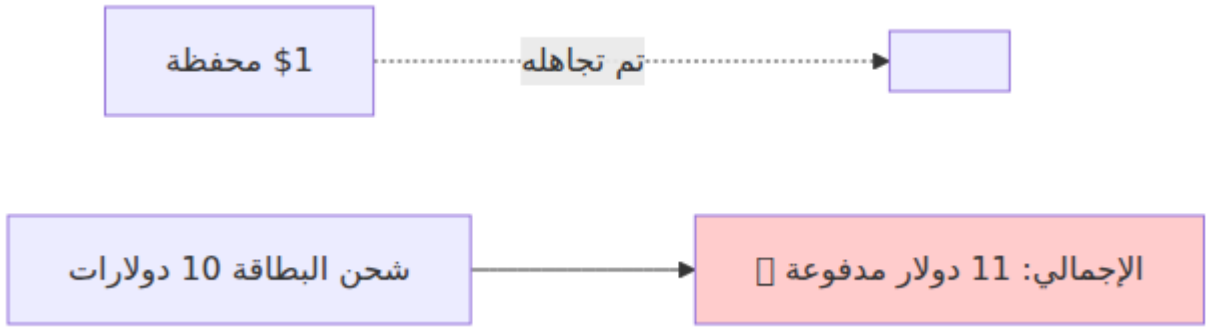
استجابة واجهة برمجة التطبيقات:

```
{
  "success": true,
  "data": {
    "service_amount": 30.00,
    "initial_balance": 50.00,
    "charged_amount": 0,              // ← لم يتم شحن البطاقة
    "wallet_debited": 30.00,
    "final_balance": 20.00,
    "payment_method_used": false      // ← لم يتم استخدام البطاقة
  }
}
```

مثال 3: حجز التفويض مع رصيد المحفظة

السيناريو: حجز تفويض بمبلغ 500 دولار 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 المبلغ
  "card_amount": 350.00,              // تم تفويض البطاقة لهذا المبلغ
  "message": "تم تفويض البطاقة بمبلغ 350 دولار (تعبئة المحفظة).
سيتم خصم المحفظة بمبلغ 500 دولار في وقت الالتقاط."
}
```

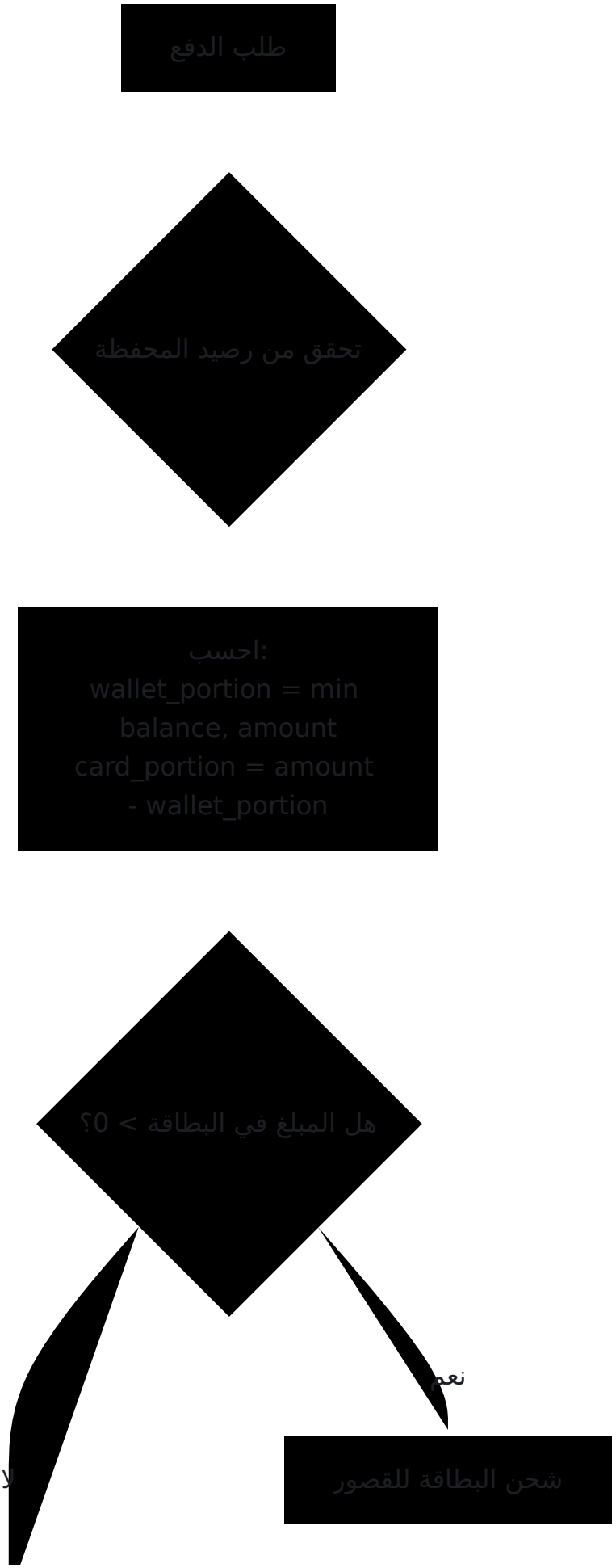
الخطوة 2: الالتقاط (`POST /api/payments/capture/301`):

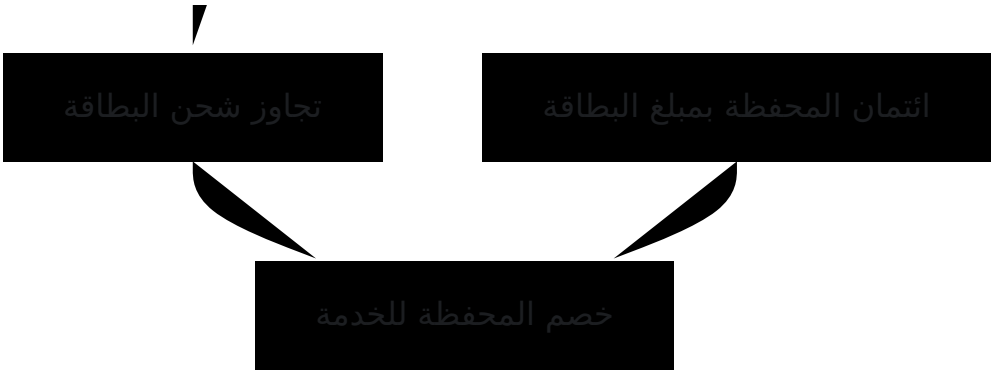
لماذا هذا مهم:

- يتم حجز 350 دولار فقط على بطاقتك، وليس 500 دولار
- يقلل من تأثير الائتمان المتاح للعميل
- مبالغ التفويض أكثر دقة
- تجربة مستخدم أفضل

تفاصيل التنفيذ

منطق توجيه المحفظة أولاً:


طلب الدفع
تحقق من رصيد المحفظة
احسب:
wallet_portion = min
balance, amount
card_portion = amount
- wallet_portion
هل المبلغ في البطاقة > 0؟
نعم
شحن البطاقة للقصور



اكتمال

كيف يعمل:

1. يتحقق النظام من الرصيد الحالي في المحفظة
2. يحسب القصور: `wallet_balance - المبلغ`
3. إذا كان القصور > 0، يشحن البطاقة فقط للقصور
4. يئتم المحفظة بمبلغ الدفع من البطاقة
5. يخصم المحفظة للمبلغ الكامل للخدمة
6. النتيجة: يتم شحن العميل فقط لما لم تتمكن المحفظة من تغطيته

تجاوز وضع التوجيه

يمكنك تجاوز سلوك المحفظة أولاً إذا لزم الأمر:

وضع التجاوز - شحن البطاقة دائمًا حتى لو كان هناك أموال في المحفظة:

```
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: يريد العميل استخدام البطاقة للحصول على مكافآت/نقاط حتى مع وجود رصيد في المحفظة.

أفضل الممارسات

1. **استخدم دائمًا توجيه المحفظة أولاً** لمدفوعات العملاء لتمكين الو��ائف المدفوعة مسبقًا
2. **استخدم المدفوعات ذات المرحلتين** (نقطة نهاية التعبئة) عند توفير الخدمات لتجنب الشحن إذا فشل التوفير
3. **قم بتعيين البيانات الوصفية** لجميع المدفوعات للحفاظ على سجل التدقيق
4. **استخدم المدفوعات المجهولة** فقط للمستخدمين المجهولين حقًا (نقاط الاتصال، الإيجارات)
5. **تعامل مع الأخطاء بشكل جيد** وقدم ملاحظات واضحة للمستخدم
6. **اختبر تدفقات الدفع** في كل من سيناريوهات النجاح والفشل
7. **راقب انتهاء صلاحية التفويضات** (عادةً 7 أيام للبطاقات)
8. **نفذ إعادة الشحن التلقائي** للخدمات المتكررة لتحسين تجربة العميل

أفضل الممارسات الخاصة بـ Playbook

عند تنفيذ تدفقات الدفع في Playbooks Ansible:

1. **استخدم دائمًا نمط الكتلة/الإنقاذ** - لف لف توفير في try/catch لاستعادة تلقائية
2. **قم بتخزين authorization_id** - احفظه لعمليات الالتقاط/الإفراج
3. **تحقق من استجابات واجهة برمجة التطبيقات** - تأكد من النجاح قبل المتابعة
4. **قم بتدوير قيم العملة** - استخدم دائمًا مكانين عشريين
5. **تحقق من طرق الدفع** - تحقق من أن العميل لديه طريقة دفع افتراضية قبل التفويض

راجع Playbook أفضل الممارسات في للحصول على تفاصيل كاملة وأمثلة.

المصادقة

تتطلب جميع نقاط نهاية واجهة برمجة التطبيقات المصادقة عبر مفتاح واجهة برمجة التطبيقات في رأس `Authorization`:

```
Authorization: Bearer YOUR_API_KEY
```

اتصل بالمسؤول عن النظام الخاص بك للحصول على توفير مفتاح واجهة برمجة التطبيقات.

دعم البائعين

المدعومون حاليًا

- **Stripe** - دعم كامل (بطاقات، ACH)
- **PayPal** - دعم كامل (حسابات PayPal، بطاقات عبر مجموعة Card Fields SDK)

إضافة بائعين جدد

تجعل الهيكلية المودولارية من السهل إضافة بائعين جدد. راجع قسم الهيكلية المودولارية للحصول على التفاصيل.

الوثائق ذات الصلة

أدلة التنفيذ

- **Playbooks التحصيل والمدفوعات من** - تنفيذ Playbook Ansible
  - نمط الالتزام ذو المرحلتين
  - أمثلة التفويض/الالتقاط
  - إنشاء المعاملات المباشرة
  - حسابات النسبة المئوية

أدلة خاصة بالميزات

- **فواتير العملاء** - إنشاء الفواتير، القوالب، والتخصيص
- **طرق الدفع** PayPal - إدارة طرق دفع العملاء، البطاقات، وحسابات
- **المعاملات** - إدارة المعاملات، الرسوم اليدوية، والائتمانات
- **معالجة الدفع** - تدفقات الدفع والمعالجة
- **أساسيات الفوترة** - أساسيات فوترة العملاء

التنقل السريع

- **هل تضيف بائعين؟** → راجع الهيكلية المودولارية
- **تكوين Stripe/PayPal** تكوين البائ → راجع **؟**
- **تخصيص الفواتير؟** → راجع فواتير العملاء
- **توفير الخدمات؟** Playbook → راجع تكامل

فواتير العملاء

> لواجهات برمجة تطبيقات دفع الفواتير وسير العمل، راجع **دليل نظام الدفع API**: **مرجع API**

`Transactions </payments_transaction>` يتم تجميعها معًا لتشكيل فاتورة، والتي تُرسل إلى العميل للدفع.

تحتوي الفواتير على تاريخ بدء وتاريخ انتهاء، وهو الفترة التي تغطيها الفاتورة، وتاريخ استحقاق، وهو التاريخ الذي يتوجب فيه دفع الفاتورة.

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.

`Mailjet <integrations_mailjet>` تكون فوات��ر العملاء مُصممة بالكامل باستخدام ويمكن تخصيصها لتشمل شعار الشركة، والعنوان، وتفاصيل الدفع، ويمكن إرسالها إلى العميل عبر البريد الإلكتروني، أو تنزيلها كملف PDF.

تخصيص قوالب الفواتير

تستخدم OmniCRM قوالب HTML مع Jinja2، لتوليد الفواتير. يمكنك تخصيص تصميم الفاتورة، والعلامة التجارية، والألوان، والتخطيط بالكامل.

موقع قالب الفاتورة

يتم تخزين قوالب الفواتير في `OmniCRM-API/invoice_templates/`

القوالب الافتراضية:

- `norfone_invoice_template.html` - نموذج 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: 'norfone_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 }}` - الدولة

عناصر خط المعاملات:

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 المعاملات:

- `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 لنسبة 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 norfone_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، الدولة</p>
    <p>الهاتف: +1-555-123-4567 | البريد الإلكتروني:
billing@yourcompany.com</p>
    <p>رقم التعريف الضريبي: 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 دفع المبلغ 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. إنشاء فاتورة اختبار
3. تنزيل PDF للتحقق من التنسيق
4. إرسال الفاتورة إلى نفسك لاختبار تسليم البريد الإلكتروني

التخصيص المتقدم

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 GST: ${{ "%.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 يتم تعيينه افتراضيًا إلى 0% أو إذا لم يتم تحديده للمنتج.

رمز QR للدفع:

إنشاء 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

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

CSS المدعوم:

- معظم خصائص CSS 2.1
- CSS3 محدود (بعض التحولات، flexbox)

- خطوط الويب عبر `@font-face`

غير المدعوم:

- 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 مقروء (ليس صغيرًا جدًا)
    - تتماشى الجداول بشكل صحيح
    - جميع الأقسام موجودة
2. **دقة البيانات:**
    - تفاصيل العميل صحيحة
    - مجموعات المعاملات صحيحة

    - تواريخ التنسيق بشكل صحيح
    - جميع المتغيرات تستبدل بشكل صحيح
3. **جودة PDF:**
    - حجم الملف معقول (<5MB)
    - الصور حادة وواضحة
    - لا يوجد نص مقطوع أو تدفق زائد
    - 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 بالضبط (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، `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 HTML أو تعطل WeasyPrint
- **الإصلاح:** تحقق من سجلات HTML، تحقق من WeasyPrint، قم بتبسيط التخطيطات المعقدة

تخزين PDF للفواتير

نظام تخزين OmniCRM الزائد، تتضمن PDF لتحسين الأداء وتقليل إنشاء PDF للفواتير. عندما يتم 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 السلامة)
        - طابع زمني للإنشاء
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 طابع `last_accessed` لتتبع استخدام التخزين
3. **إبطال التخزين** - يتم إبطال PDFs المخزنة عندما:
    - يتم تعديل الفاتورة (إضافة/إزالة المعاملات، تغيير التفاصيل)
    - يتم تحديث قالب الفاتورة
    - يتم تشغيل مسح التخزين يدويًا

الفوائد:

- **الأداء** - تسليم PDF فوري للطلبات المتكررة (بدون تأخير في إعادة التوليد)
- **الاتساق** - نفس PDF لجميع تنزيلات الفاتورة (ما لم يتم تعديل الفاتورة)
- **تحميل الخادم** - تقليل استخدام وحدة المعالجة المركزية من إنشاء PDF
- **تجربة المستخدم** - يظهر مؤشر التحميل أثناء الإنشاء 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 ��لإدخالات التي لم يتم الوصول إليها في 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 التحميل أثناء إنشاء PDF، ثم يتم تخزينه
- الطلبات اللاحقة: تنزيل فوري من التخزين
- تأثير التخزين/عدم وجوده 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 ا��نظام:

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 (جميع الأوقات) 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: عرضها كلاحقة "k" (1.5$ مثلk)
- القيم التي تزيد عن 1,000,000: عرضها كلاحقة "M" (2.3$ مثلM)
- القيم التي تزيد عن 1,000,000,000: عرضها كلاحقة "B" (1.1$ مثلB)

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 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)

رموز 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

لإنشاء فاتورة جديدة انقر على **"إنشاء فاتورة برو forma"**.

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 نطاق التاريخ مع القدرة على تضمين/استبعاد معاملات معينة

اختيار المعاملات:

- ✓ **(علامة زرقاء)** - انقر لاستبعاد معاملة من الفاتورة
- × **(علامة حمراء)** - انقر لتضمين معاملة تم استبعادها مسبقًا
- **تحديد الكل** - تضمين جميع المعاملات المعروضة
- **مسح الكل** - استبعاد جميع المعاملات

- تظهر المعاملات المستبعدة بلون رمادي مع نص مشطوب
- يتم تحديث الإجماليات في الوقت الحقيقي أثناء تحديد/إلغاء تحديد المعاملات

ماذا يحدث:

1. يجد النظام جميع المعاملات غير المفوترة للعميل ضمن نطاق التاريخ
2. يعرض معاينة المعاملة مع القدرة على تضمين/استبعاد المعاملات الفردية
3. يظهر حساب في الوقت الحقيقي للمجموع الفرعي، والضريبة، والإجمالي بناءً على المعاملات المحددة
4. يتم إضافة المعاملات المحددة (المتضمنة) فقط إلى الفاتورة
5. يتم إنشاء PDF الفاتورة وتخزينه
6. يتم وضع علامة على المعاملات المحددة كفواتير (حقل مُعبأ `invoice_id`)
7. تبقى المعاملات المستبعدة غير مفوترة ومتاحة للفواتير المستقبلية
8. تظهر الفاتورة في القائمة مع حالة "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 (مثل، استبعاد المعاملات غير المتنقلة 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، والإجماليات، والإجراءات المتاحة.

نافذة تفاصيل الفاتورة:

- **معلومات الفاتورة** - تظهر معرف الفاتورة، والعنوان، والتواريخ، وحالة الدفع، وحالة الإلغاء
- **قائمة المعاملات** - تعرض جميع المعاملات المضمنة في الفاتورة مع:
  - تاريخ المعاملة
  - العنوان والوصف
  - تكلفة التجزئة
  - مبلغ الضريبة والنسبة (مُنسقة 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 (متاحة دائمًا)
  - **إرسال بريد إلكتروني** - إرسال الفاتورة إلى العميل (الفواتير غير الملغاة)
  - **دفع الفاتورة** - معالجة الدفع (فواتير غير مدفوعة، غير ملغاة فقط)
  - **استرداد** - استرداد دفعة Stripe (فواتير Stripe المدفوعة فقط)
  - **حذف** - إلغاء الفاتورة (فواتير غير مدفوعة، غير ملغاة فقط)

تنزيل PDFs للفواتير

في نافذة تفاصيل الفاتورة لتنزيل **"PDF** انقر على **رمز التنزيل (⬇)** في الجدول أو زر **"تنزيل** فاتورة كملف PDF.

عملية التنزيل:

1. انقر على رمز التنزيل بجوار الفاتورة
2. يظهر مؤشر التحميل أثناء الإنشاء (في المرة الأولى فقط)
3. يطلب المتصفح حفظ الملف: `Invoice_01234.pdf`
4. أو يتم حفظه في مجلد التنزيلات PDF يفتح

سلوك التخزين PDF:

- يتم إنشاء PDF - **أول تنزيل** 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 تكون API معطلة
- **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 **رمز الدفع (💳)** لدفع فاتورة عبر الإنترنت.

عملية الدفع:

1. انقر على رمز 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 المس��خدمين)
    - **نقدًا** - دفع نقدي يدوي (موظف فقط)
    - **استرداد** - تطبيق استرداد كدفع (موظف فقط)
    - **معاملة POS** - جهاز نقطة البيع (موظف فقط)
    - **تحويل بنكي** - تحويل بنكي يدوي (موظف 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 الدفع
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

الرعاية الذاتية مقابل دفع الموظف:

:doc:`بوابة الرعاية الذاتية <self_care_portal>` (العملاء):

- تتوفر فقط طريقة دفع Stripe
- يجب أن يكون لديك طريقة دفع محفوظة
- يظهر تحذير إذا لم توجد طرق دفع
- يتم توفير رابط لإضافة طريقة دفع

بوابة الموظف (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 يطالبهم بإضافة طريقة دفع قبل أن يتمكنوا من دفع الفواتير.

إرسال الفواتير عبر البريد الإلكتروني

انقر على **رمز البريد الإلكتروني (✉)** لإرسال الفاتورة إلى العميل.

ماذا يحدث:

1. انقر على رمز البريد الإلكتروني بجوار الفاتورة

2. يسترجع النظام PDF (أو ينشئها إذا لم تكن مخزنة) الفاتورة من التخزين
3. يرسل البريد الإلكتروني عبر `Mailjet <integrations_mailjet>` باست��دام قالب `api_crmCommunicationCustomerInvoice`
4. يتضمن البريد الإلكتروني:
    - PDF الفاتورة كمرفق
    - اسم العميل
    - رقم الفاتورة وتاريخ الاستحقاق
    - إجمالي المبلغ 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 }}` - إجمالي المبلغ 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 (حاول تنزيله أولاً)

إلغاء الفواتير

لإلغاء فاتورة **رمز الحذف ()** على انقر.

المتطلبات:

- يجب أن تكون الفاتورة **غير مدفوعة**
- لا يمكن إلغاء الفواتير المدفوعة (يجب استردادها بدلاً من ذلك)

كيفية الإلغاء:

1. ابحث عن الفاتورة غير المدفوعة في القائمة
2. انقر على رمز الحذف ()
3. أكد في النافذة:

ماذا 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 ��اتورة 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 تكون هذه الفاتورة موجودة أبدًا" (خطأ في الفوترة، تكرار)
- **الاسترداد** = "عكس فاتورة مدفوعة صحيحة" (إرجاع المال للعميل)

استرداد الفواتير

انقر على **رمز 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 فقط لمستخدمي الموظفين (ليس للرعاية الذاتية)

كيفية الاسترداد:

1. ابحث عن فاتورة Stripe المدفوعة
2. انقر على رمز الاسترداد (□)
3. تفتح نافذة تأكيد ال��سترداد:

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

ماذا يحدث بعد الاسترداد:

- تبقى الفاتورة في النظام (غير ملغاة)
- تظهر الحالة "مستردة"
- تبقى المعاملات مرتبطة بالفاتورة
- يتلقى العميل استردادًا 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) أو عبر التحويل البنكي عكس يدوي)
- لا يمكن الاسترداد الجزئي (مبلغ الفاتورة بالكامل فقط)
- لا يمكن الاسترداد مرتين

البحث وتصنيف الفواتير

البحث

استخدم ش��يط البحث للعثور على الفواتير. يبحث عبر:

- معرف الفاتورة
- عنوان الفاتورة
- اسم العميل (عرض على مستوى النظام فقط)

الفلاتر

قم بتطبيق 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 تاريخ الإنشاء
- **المبلغ** - الفرز حسب تكلفة التجزئة (الأعلى/الأدنى)
- **الحالة** - الفرز حسب حالة الدفع (المدفوعة أولاً أو غير المدفوعة أولاً)

انقر مرة أ��رى لعكس اتجاه الفرز (تصاعدي ↔ تنازلي).

الصفحات

انتقل عبر 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. يصل نهاية الشهر (مثل 31 يناير)
2. انتقل إلى **الفوترة** → **الفواتير**
3. انقر على **"+ إنشاء فاتورة برو forma"**
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. يظهر قسم **معاينة المعاملة** مع جميع المعاملات غير المفوترة من يناير
7. مراجعة المعاينة:
    - يتم تضمين جميع المعاملات بشكل افتراضي
    - تحقق من الإجما��يات: المجموع الفرعي، الضريبة، وإجمالي الفاتورة
    - تحقق من صحة جميع الرسوم
8. انقر على **"إنشاء الفاتورة"** (يظهر عدد المعاملات، مثل "إنشاء 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 رمز البريد الإلكتروني في الجدول
12. يتلقى العميل بريد الفاتورة مع PDF ورابط الدفع

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 برو forma"**
    - العنوان: "خدمات الهاتف المحمول - يناير 2025"
    - تاريخ البدء/الانتهاء: من 1 إلى 31 يناير
    - تاريخ الاستحقاق: 15 فبراير
    - في معاينة المعاملة، **استبعد** جميع المعاملات غير المتنقلة:
        - انقر على زر **X** بجوار معاملات الإنترنت
        - انقر على زر **X** بجوار الرسوم المتنوعة
        - تبقى فقط معاملات خدمة الهاتف المحمول محددة
    - تحقق من أن الإجماليات تعكس فقط خدمات الهاتف المحمول
    - انقر على **"إنشاء الفاتورة"** (يظهر عدد معاملات الهاتف المحمول)
4. **إنشاء الفاتورة الثانية (خدمات الإنترنت):**
    - انقر على **"إنشاء فاتورة برو forma"** مرة أخرى

- العنوان: "خدمات الإنترنت - يناير 2025"
- تاريخ البدء/الانتهاء: من 1 إلى 31 يناير (نفس الفترة)
- في معاينة المعاملة:
  - تم فوتره المعاملات المتنقلة بالفعل (لا تظهر)
  - استبعد الرسوم المتنوعة باستخدام زر **X**
  - تبقى فقط معاملات خدمة الإنترنت
- انقر على **"إنشاء الفاتورة"**

5. **إنشاء الفاتورة الثالثة (الرسوم الإضافية):**
   - مرة أخرى انقر على **"إنشاء فاتورة ب��و forma"**
   - العنوان: "الرسوم الإضافية - يناير 2025"
   - تظهر فقط الرسوم المتنوعة غير المفوترة في المعاينة
   - انقر على **"تحديد الكل"** لتضمين الجميع
   - انقر على **"إنشاء الفاتورة"**
6. إرسال جميع الفواتير الثلاثة إلى العميل

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 برو forma"**
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:
   - ابحث عن المعاملة المتنازع عليها (مثل "رسوم تجاوز البيانات")
   - انقر على زر **X** لاستبعادها
   - ابحث عن المعاملة المعلقة (مثل "رسوم التركيب")
   - انقر على زر **X** لاستبعادها
   - يتم تحديث عدد المعاملات: "تم تحديد 18 معاملة"
   - يتم إعادة حساب الإجماليات تلقائيًا
8. مراجعة الإجماليات المحدثة (تستبعد المبالغ المتنازع عليها)
9. انقر على **"إنشاء الفاتورة (18)"**

10. تم إنشاء الفاتورة مع المعاملات المعتمدة فقط
11. تبقى المعاملات المتنازع عليها/المعلقة غير مفوترة للدورة الفوترة التالية

سير العمل 4: مراجعة سريعة للفواتير وضبطها

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 الإجماليات في الوقت الحقيقي
5. تحقق من أن الإجمالي الجديد يتطابق مع المبلغ المتوقع
6. انقر على **"إنشاء الفاتورة"** مع المعاملات المصححة
7. ارجع وألغِ/احذف ا��معاملات المستبعدة إذا لزم الأمر
8. إرسال الفاتورة إلى العميل بثقة

سير العمل 5: فاتورة تركيب لمرة واحدة

1. أكمل الفني الميداني التركيب
2. يضيف الموظف معاملة التركيب يدويًا
3. انتقل إلى علامة تبويب **الفوترة** للعميل
4. انقر على **"+ إنشاء فاتورة برو forma"**
5. تعيين التواريخ:
    - تاريخ البدء: اليوم
    - تاريخ الانتهاء: اليوم
    - تاريخ الاستحقاق: اليوم + 7 أيام
    - العنوان: "خدمات التركيب"
6. تظهر معاينة المعاملة فقط معاملات اليوم
7. تحقق من ظهور رسوم التركيب
8. استبعد أي رسوم متكررة باستخدام زر **X** (إذا كانت موجودة)
9. انقر على **"إنشاء الفاتورة"**
10. إرسالها 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. يتنقل الموظف إلى قائمة فواتير العميل
3. **انقر على صف الفاتورة** لفتح نافذة تفاصيل الفاتورة
4. مراجعة معلومات الفاتورة:
    - معرف الفاتورة، التواريخ، الحالة
    - جميع المعاملات المضمنة مع الوصف
    - تفاصيل الضريبة لكل معاملة
    - المجموع الفرعي، الضريبة، والمبالغ الإجمالية
5. الإجابة على 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. تأكيد الإلغاء
8. تصبح المعاملات ��ير 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. إرسال 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، وتُعَلَّم الفاتورة كـ "مدفوعة"
11. كرر ذلك لبقية الفواتير
12. جميع الفواتير الآن مُعَلَّمة كـ "دفوعة��"

سير العمل 9: التعامل مع طلب استرداد

1. يطلب العميل استردادًا بسبب الدفع 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. تأكيد الاسترداد
8. يقوم النظام بمعالجة استرداد Stripe
9. تتغير حالة الفاتورة إلى "مستردة"
10. يتلقى العميل الاسترداد في غضون 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. ضبط نطاق التاريخ أو إزالة الفلتر.

الفاتورة PDF فشل إنشاء

- **السبب:** خطأ في بناء القالب، تعطل WeasyPrint، أو بيانات العميل مفقودة
- **الإصلاح:** تحقق من قالب الفاتورة HTML بحثًا عن الأخطاء، تحقق من تعبئة حقول API. عنوان العميل، مراجعة سجلات

Stripe فشل الدفع مع خطأ

- **السبب:** تم رفض البطاقة، أو عدم كفاية الأموال، أو انتهاء صلاحية البطاقة، أو مشكلة في API Stripe
- **الإصلاح:** جرب طريقة دفع مختلفة، تحقق من صلاحية 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 Stripe لفواتير أو أنشئ معاملة ائتمانية يدويًا.

البريد الإلكتروني للفواتير لا يُرسل

- **السبب:** بيانات اعتماد API Mailjet غير صحيحة، أو العميل ليس لديه جهة اتصال فوترة، أو القالب 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 الفاتورة غير مدفوعة
- **الإصلاح:** يظهر زر الاسترداد فقط لمدفوعات Stripe. بالنسبة لطرق الدفع الأخرى، أنشئ معاملة ائتمانية يدوية.

قديم بتصميم القالب PDF يظهر

- **السبب:** تم تخزين PDF قبل تحديث القالب

- امسح تخزين PDF للفواتير: `DELETE FROM Invoice_PDF_Cache WHERE invoice_id = X;` **الإصلاح:**

لا يمكن للعميل دفع الفاتورة (لا توجد طرق دفع)

- **السبب:** لا توجد طرق دفع محفوظة في بوابة الرعاية الذاتية
- **الإصلاح:** يجب على العميل إضافة بطاقة ائتمان في صفحة **طرق الدفع** قبل دفع الفواتير.

تم إنشاء فواتير متعددة لنفس الفترة

- **السبب:** قام الموظف بإنش��ء فاتورة مرتين، أو تتداخل نطاقات التواريخ
- **الإصلاح:** قم بإلغاء الفاتورة المكررة. ضبط نطاقات التواريخ لمنع التداخل. استخدم معاينة المعاملة لضمان مجموعات معاملات فريدة.

لا تظهر معاملات في المعاينة

- **السبب:** جميع المعاملات في نطاق التاريخ تم فوترتها بالفعل أو لا توجد معاملات
- **الإصلاح:** تحقق من أن نطاق التاريخ صحيح. تحقق من قائمة المعاملات لتأكيد وجود المعاملات. قم بتصفية الفواتير لرؤية الفاتورة التي 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. 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 لكل معاملة والضريبة. 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 (ليست مستبعدة). تحقق من أن تاريخ البدء يأتي قبل تاريخ الانتهاء. تحقق من تعيين تاريخ الاستحقاق.

نافذة تفاصيل الفاتورة لا تفتح

- **السبب:** خطأ في JavaScript أو الصفحة لم يتم تحميلها بالكامل
- **الإصلاح:** قم بتحديث الصفحة. تحقق من وحدة التحكم في المتصفح بحثًا عن الأخطاء. جرب متصفحًا مختلفًا. تحقق من الاتصال بالإنترنت.

الضريبة لا تظهر في تفاصيل الفاتورة

- **السبب:** المعاملة تحتوي على 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 المعاملات التي تظهر في الفواتير
- `payments_process` - معالجة مدفوعات الفواتير
- `basics_payment` - إدارة طرق الدفع
- `payment_system_guide` - مرجع API الدفع وتكوين البائع

معالجة المدفوعات

ستتم معالجة معظم المدفوعات تلقائيًا بواسطة النظام، ولكن هناك أوقات قد تحتاج فيها إلى معالجة دفعة يدويًا.

لدفع فاتورة، حدد الفاتورة غير المدفوعة، وانقر على زر "دفع".

سيؤدي ذلك إلى فتح نموذج دفع، حيث يمكنك إدخال طريقة الدفع، والنقر على "إرسال" لمعالجة الدفع.

سيتلقى العميل إيصالًا تلقائيًا للدفع، وسيتم وضع علامة على الفاتورة كمدفوعة.

بالنسبة للتحويلات البنكية، يمكنك إدخال مرجع الدفع وتاريخ 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 ائتمانًا، يتم إنشاء معاملة لمبلغ الائتمان، أو إذا تم فرض رسوم تركيب، يتم إنشاء معاملة لرسوم التركيب.

تُجمع المعاملات معًا لتشكيل `Invoices <payments_invoices>`، 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 عميل]**
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 تُدرج بعد في فاتورة
- **إجمالي المعاملات هذا الشهر** - مجموع المعاملات التي تم إنشاؤها هذا الشهر الميلادي
- **إجمالي المعاملات الشهر الماضي** - مجموع المعاملات التي تم إنشاؤها الشهر الماضي

تنسيق القيم:

- القيم التي تزيد عن 1,000: عرضها مع لاحقة "k" (على سبيل المثال، 1.5$k)
- القيم التي تزيد عن 1,000,000: عرضها مع لاحقة "M" (على سبيل المثال، 2.3$M)

- القيم التي تزيد عن 1,000,000,000: عرضها مع لاحقة "B" (على سبيل المثال، 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 للائتمانات)
- **الفاتورة** - معرف الفاتورة إذا تم فوتره المعاملة (رابط قابل للنقر)
- **الحالة** - علامة تحقق إذا تم فوترتها، شرطة إذا لم يتم فوترتها بعد

الإجراءات لكل صف:

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

- **عرض التفاصيل** - يفتح نافذة تفاصيل المعاملة
- **تحميل فاتورة PDF** - تحميل 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 لأجهزة المودم، 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
- **ائتمانات الحساب** - ائتمانات حسن النية، تعويض
- **الاستردادات** - الأموال المعادة 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

ماذا يحدث:

1. يتم إنشاء المعاملة في قاعدة البيانات
2. تظهر في قائمة معاملات العميل

3. تُدرج في عدد "المعاملات غير المفوترة"
4. متاحة للإدراج في إنشاء الفاتورة التالية
5. يتم إنشاء إدخال في سجل النشاط

البحث وتصفية المعاملات

البحث

استخدم شريط البحث للعثور على المعاملات. يبحث عبر:

- معرف المعاملة
- العنوان
- الوصف
- معرف الفاتورة

الفلاتر

قم بتطبيق 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 ال��طأ (تحديدها كملغاة).

المتطلبات:

- يجب ألا تكون المعاملة مفوترة
- بمجرد أن يتم 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

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 تتضمن 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
  - **التجاوز اليدوي** - يمكن للموظفين تجاوز نسبة الضريبة عند إنشاء معاملة
  - **مبلغ الضريبة**: يتم حسابه كالتالي - `retail_cost × (tax_percentage / 100)`
  - **تنسيق العرض** - يظهر كـ: `10.00$ (%10)` في قوائم المعاملات
- **معاملات الائتمان (المدفوعات/الاستردادات)** - لا تُطبق ضريبة على الائتمانات
  - يتم إخفاء حقل نسبة الضريبة للمعاملات الائتمانية

- يتم تعيين الضريبة تلقائيًا إلى 0% لجميع المدفوعات والاستردادات
- تقلل الائتمانات من رصيد العميل المستحق 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%)

ضريبة صفرية (NIL/معفاة):

- يمكن أن تكون المنتجات معفاة من الضريبة عن طريق تعيين نسبة الضريبة إلى 0
- الافتراضي للضريبة هو 0% إذا لم يتم تحديدها
- تظهر المعاملات المعفاة من الضريبة "-" في عمود الضريبة

عرض تفاصيل المعاملة

انقر على معاملة لعرض التفاصيل الكاملة:

المعاملات المفوترة مقابل غير المفوترة

المعاملات غير المفوترة:

- لم تُدرج بعد في أي فاتورة
- متاحة لإنشاء الفاتورة التالية
- يمكن إلغاؤها
- تُحتسب في إجمالي "المعاملات غير المفوترة"
- الحالة تظهر شرطة (-)

المعاملات المفوترة:

- مدرجة 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. تُدرج المعاملات في الفاتورة الجديدة
4. يتم ملء حقل `invoice_id` للمعاملة
5. يتم الآن وضع علامة على المعاملة كـ "مفوتر"

انظر `payments_invoices` للحصول على تفاصيل 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 الائتمان **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 المفوترة
14. سيتم تضمينها في الفاتورة التالية

سير العمل 3: إلغاء معاملة مكررة

1. يلاحظ الموظف وجود معاملة مكررة
2. تحقق من أن المعاملة لم تُفوتر 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 المفوترين 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 إلى إنشاء 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 المعاملة

- **السبب:** تم فوترت المعاملة بالفعل
- **الحل:** المعاملة جزء من تاريخ الفواتير. إذا كانت هناك حاجة إلى استرداد، قم بإنشاء معاملة ائتمان بدلاً من ذلك.

ظهور معاملات مكررة

- **السبب:** تم فرض رسوم على الخدمة عدة مرات أو خطأ في التوفير
- **الحل:** قم بإلغاء المعاملة (المعاملات) المكررة إذا لم يتم فوترتها. إذا تم فوترتها، قم بإصدار ائتمان.

لا تظهر المعاملة في القائمة

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

إجمالي غير المفوتر لا يتطابق مع المتوقع

- **السبب:** بعض المعاملات تم فوترتها بالفعل، أو تم استبعاد المعاملات الملغاة
- **الحل:** قم بتطبيق الفلتر "الفاتورة: غير مفوترة" لرؤية فقط غير المفوترين. تحقق من المعاملات الملغاة بشكل منفصل.

لا يمكن إضافة معاملة (حقل العميل معطل)

- **السبب:** عرض صفحة المعاملات ��لخاصة بالعميل
- **الحل:** العميل محدد 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 الشحن ومعالجة المدفوعات داخل Playbooks الخاصة بـ Ansible لعمليات توفير OmniCRM.

> **كامل**: للحصول على تفاصيل كاملة حول واجهات برمجة التطبيقات للدفع، **مرجع API**
> وتوجيه المحفظة، والمبالغ المستردة، والهندسة المعمارية المستقلة عن البائع، راجع **دليل API نظام الدفع**

نظرة عامة

يمكن أن تتعامل Playbooks الخاصة بـ OmniCRM Ansible مع معالجة المدفوعات بعدة طرق مختلفة، لكن جميعها تقوم فقط باستدعاء واجهات برمجة التطبيقات على CRM للشحن.

1. **تدفق دفع الالتزام ذو المرحلتين** - للخدمات المدفوعة التي تتطلب تفويض دفع فوري واحتجاز (أي الخدمات المدفوعة مسبقًا)
2. **إنشاء معاملة مباشرة** - لإضافة رسوم/ائتمانات بدون معالجة دفع فورية (مثل رسوم Postpad) (أي خدمات الإعداد، الائتمانات اليدوية) والتي يمكن معالجتها لاحقًا

تضمن هذه الأساليب معاملات ذرية حيث يتم شحن العملاء فقط إذا نجحت عملية 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. إنها لا تحدد ما يجب عليك شحنه - Playbook لديها السيطرة الكاملة على التسعير النهائي.

Playbook الخاصة بـ Ansible هي في الأساس **محرك برمجة** يمكنك تخصيصه لتلبية أي متطلبات منطق الأعمال. يمكنك:

- **استخدام الأسعار المخزنة كما هي** - ببساطة الإشارة إلى `api_response_product.json.retail_cost` في تفويضك

- **تجاوز الأسعار تمامًا** - شحن مبلغ مختلف بغض النظر عما هو موجود في تعريف المنتج
- **تطبيق خصومات ديناميكية** - حساب النسب المئوية بناءً على مستوى العميل، العروض الترويجية، أو الولاء
- **تنفيذ تسعير متدرج** - شحن معدلات مختلفة بناءً على الكمية، مستويات الاستخدام، أو شروط العقد
- **تسعير مجمع** - دمج عدة منتجات مع خصومات مجمعة مخصصة
- **تسعير يعتمد على الوقت** - ضبط الأسعار بناءً على الوقت من اليوم، الموسم، أو فترات العروض الترويجية
- **تسعير خاص بالعميل** - البحث عن الأسعار المتفاوض عليها من عقود العملاء

مثال: تجاوز سعر المنتج

```yaml
# هو 99.00 دولار، لكننا نريد شحن 79.00 دولار كعرض سعر المنتج retail_cost 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 شريك الجملة

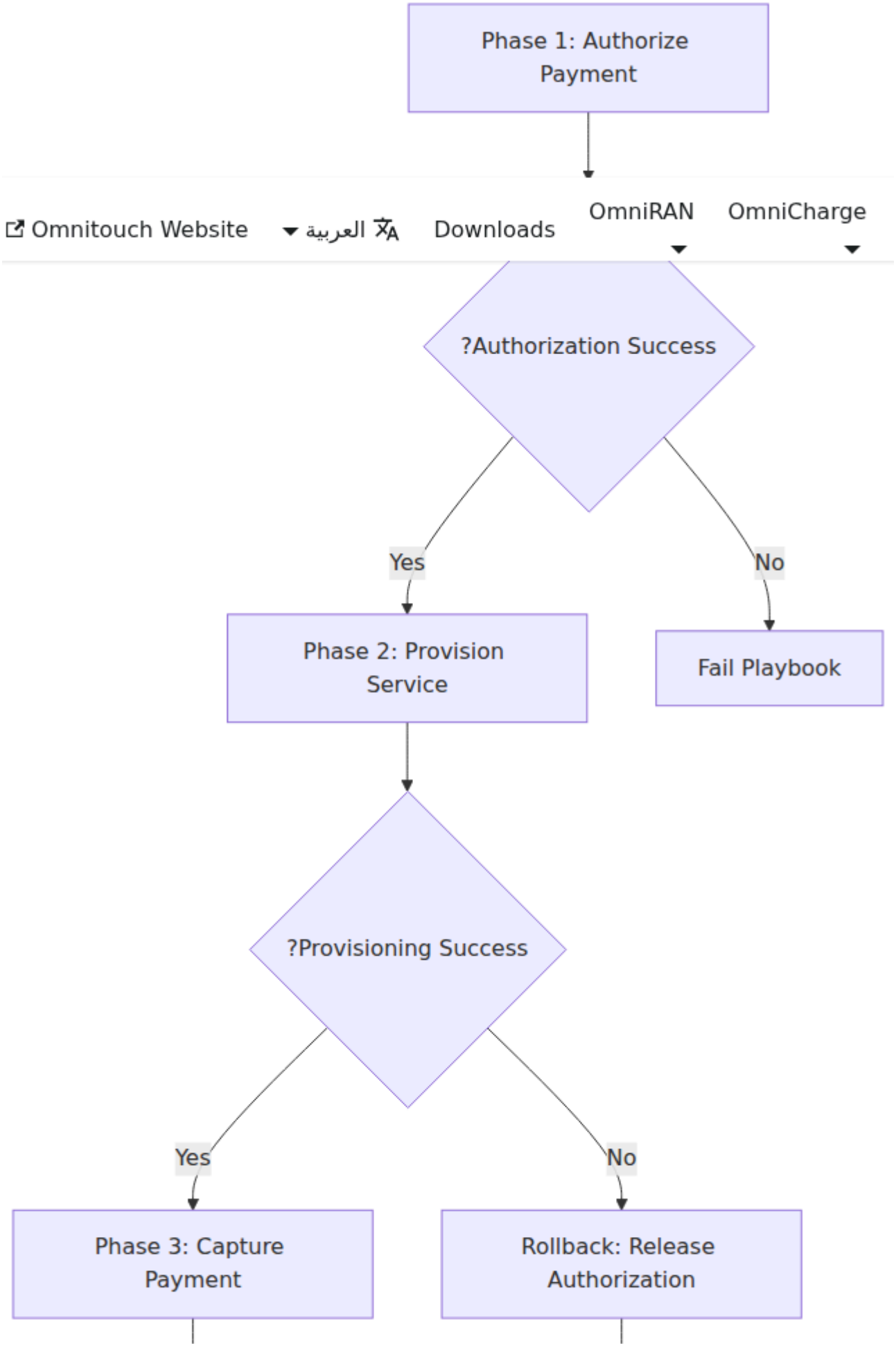
```
# يدفع الشركاء بالجملة wholesale_cost بدلاً من retail_cost
- name: Determine price based on customer type
  set_fact:
    charge_amount: >-
      {% if customer_info.json.is_wholesale_partner %}
      {{ api_response_product.json.wholesale_cost | float }}
      {% else %}
      {{ api_response_product.json.retail_cost | float }}
      {% endif %}
```

السيطرة البرمجية Playbook النقطة الرئيسية هي أنك **لست مقيدًا بأي نموذج تسعير**. يمنحك هي مجرد CRM الكاملة لتنفيذ أي قواعد أع��ال تتطلبها منظمتك. أسعار المنتجات/الخدمات في قيم افتراضية ملائمة يمكن أن تستخدمها Playbooks أو تتجاهلها 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 اكتملت عملية التوفير بنجاح.

نظرة عامة على التدفق


Phase 1: Authorize Payment
Omnitouch Website
العربية
Downloads
OmniRAN
OmniCharge
?Authorization Success
Yes
No
Phase 2: Provision Service
Fail Playbook
?Provisioning Success
Yes
No
Phase 3: Capture Payment
Rollback: Release Authorization

نمط التنفيذ

يتبع النمط هذه المراحل:

المرحلة 1: التفويض - احتجاز الأموال على طريقة الدفع **المرحلة 2: التوفير** - تنفيذ تحديثات رصيد/إجراءات OCS - تحرير **التراجع** - إنهاء الدفع عند نجاح التوفير **المرحلة 3: الالتقاط** 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: المرحلة 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 التوفير الناجح
        - 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:
        # تراجع المعاملة: إلغاء 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 إلغاء تفويض الدفع {{ authorization_id }}. فشل توفير OCS.
لم يتم شحن العميل."
```

نقاط نهاية API للدفع

 توجيه المحفظة أولاً: يقوم نظام الدفع تلقائيًا بتحسين رسوم البطاقة 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 نهاية احتجازًا على طريقة دفع العميل للمبلغ ا��محدد. يتم حجز الأموال ولكن لم يتم التقاطها بعد.

سلوك المحفظة أولاً: يقوم التفويض تلقائيًا بحساب العجز بعد التحقق 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 والوصف",
    "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 (يكون 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 يبدأ 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 مكالمات الالتقاط أو الإلغاء
- لاحظ أن `card_amount` يظهر فقط أن العجز 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 النهاية بإنهاء تفويض الدفع وتحصيل المبلغ من 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
  }
}
```

ما يحدث:

1. **تم التقاط البطاقة** للمبلغ العاجل (إذا تم تفويض البطاقة)
2. **تم ائتمان المحفظة** بمبلغ البطاق�ة الملتقطة
3. **تم خصم المحفظة** للمبلغ الكامل للخدمة
4. إذا كانت `invoice: true` في بيانات التعريف الخاصة بالتفويض:
   - **تم إنشاء معاملة الخصم** (مبلغ إيجابي = شحن)
   - **تم إنشاء فاتورة** مرتبطة بمعاملة الخصم
   - **تم إنشاء معاملة ائتمان** (مبلغ سالب = الدفع المستلم)
   - تم وضع علامة على الفاتورة كمدفوعة (الخصومات والائتمانات تتساوى مع الصفر)
   - 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)
- الرصيد النهائي للمحفظة: 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 النهاية بإلغاء 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"
}
```

ما يحدث:

1. **تم إطلاق تفويض البطاقة** (إذا تم تفويض البطاقة)
2. يتم 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 لأن المحفظة لا يتم خصمها حتى وقت الالتقاط.

إنشاء معاملة مباشرة

بالنسبة للرسوم ��لتي لا تتطلب معالجة الدفع (رسوم الإعداد، الائتمانات اليدوية، التعديلات)، يمكنك إنشاء معاملات مباشرة عبر واجهة برمجة التطبيقات.

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 ال��لب:

- **customer_id** (مطلوب) - معرف العميل
- **service_id** (اختياري) - معرف الخدمة لربط المعاملة بها
- **title** (مطلوب) - اسم قصير للمعاملة
- **description** (اختياري) - وصف مفصل
- **invoice_id** (اختياري) - معرف الفاتورة إذا تم إصدارها مسبقًا (عادةً null)
- **wholesale_cost** (اختياري) - تكلفتك
- **retail_cost** (مطلوب) - التكلفة الموجهة للعميل
- **site_id** (اختياري) - معرف الموقع لربط المعاملة به
- **tax_percentage** (اختياري) - نسبة الضريبة

حالات الاستخدام:

- رسوم الإعداد أثناء توفير الخدمة
- رسوم التركيب
- الائتمانات أو التعديلات اليدوية
- رسوم المعدات
- الرسوم الإدارية

ملاحظة: إنشاء المعاملة مباشرة لا يعالج الدفع - إنه فقط ينشئ سجل فواتير. ستظهر المعاملة كغير مفوترة ويمكن تضمينها في الفواتير المستقبلية.

حساب الرسوم النسبية

يسمح الشحن النسبي لك بشحن العملاء بشكل نسبي بناءً على فترات الفواتير الجزئية. هذا شائع عندما:

- يقوم العميل بالتسجيل في منتصف الشهر
- يتم ترقية/خفض الخدمة في منتصف الدورة
- يتم حساب الفواتير بناءً على الأيام المستخدمة

صيغة حساب النسبة

```
pro_rata_charge = (monthly_cost × days_remaining) / days_in_month
```

مثال التنفيذ

إليك كيفية حساب رسوم نسبية في Playbook:

```
# حساب الرسوم النسبية لشهر جزئي
# إذا قام العميل بالتسجيل في اليوم الخامس عشر والفوترة في اليوم الأول،
# يتم الشحن بشكل نسبي 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 معين 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 (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 الجديدة: 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 بلف 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:
    # تحرير 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

تحقق 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 لاستخدامه في الالتقاط/الإلغاء:

```
- 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 منزلتين عشريتين:

```
- 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: تحرير 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 (بدون معالجة الدفع)
4. تظهر رسوم الإعداد في الفاتورة التالية

راجع `play_simple_service.yaml` للحصول على مثال كامل في الأسطر 202-232.

النمط 3: تعبئة/إضافة مجانية

بالنسبة للتعبئة المجانية التي لا تتطلب الدفع:

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

لرسوم تلقائية متكررة:

1. إنشاء إجراء مع `*http_post` إلى نقطة نهاية التوفير
2. إنشاء ActionPlan مع توقيت `*monthly`
3. تعيين ActionPlan إلى الحساب
4. ستقوم CGRateS تلقائيًا باستدعاء نقطة النهاية شهريًا
5. تتعامل Playbook نقطة النهاية مع معالجة الدفع

استكشاف الأخطاء وإصلاحها

فشل التفويض

الأعراض: ترجع نقطة نهاية التفويض خطأ

الأسباب الشائعة:

- طريقة 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

الأعراض: تم توفير الخدمة ولكن فشل التقاط الدفع

المشكلة: هذه حالة فشل حرجة - الخدمة نشطة ولكن لم يتم شحن العميل

الحل:

- قد يكون التفويض قد انتهت صلاحيته (عادةً 7 أيام)
- تحقق من أن التفويض لا يزال صالحًا قبل محاولة الالتقاط
- تنفيذ المراقبة لفشل الالتقاط
- قد تكون هناك حاجة لتدخل يدوي

لم يتم إنشاء معاملة بعد الالتقاط

الأعراض: تم التقاط الدفع ولكن لا توجد معاملة في الفواتير

السبب: لم يتم تعيين `invoice: true` في بيانات التعريف الخاصة بالتفويض

الحل: إما:

- تعيين `invoice: true` في بيانات التعريف الخاصة بالتفويض، أو

- إنشاء المعاملة يدويًا بعد الالتقاط الناجح

حساب النسبة غير صحيح

الأعراض: الرسوم النسبية لا تتطابق مع القيم المتوقعة

المشكلات الشائعة:

- أخطاء في العد (شاملة/غير شاملة تواريخ البدء/النهاية)
- شهر خاطئ مستخدم لحساب الأيام
- أخطاء تقريب

الحل:

- استخدم نطاقات تواريخ شاملة (شاملة كل من اليوم الأول والأخير)
- قم دائمًا بالتقريب إلى منزلتين عشريتين
- اختبر الحسابات بقيم معروفة
- وثق أي تواريخ مشمولة في الحسابات

المبالغ المستردة والتعامل مع الأخطاء

خيارات المبالغ المستردة

يدعم نظام الدفع نوعين من المبالغ المستردة:

1. **استرداد إلى مصدر الدفع**/PayPal - يتم 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 للشراء في المستقبل (يتم التعامل معه تلقائيًا في سيناريوهات الخطأ)

بالنسبة لفشل التوفير، يقوم النظام تلقائيًا بإئتمان المحفظة بدلاً من استرداد البطاقة إلى:

- تجنب رسوم الاسترداد
- توفير توافر 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 للدفع (Square، Adyen، Braintree، إلخ) دون تغيير Playbooks.

انظر أيضًا:

- الهندسة المعمارية المعيارية - كيف تعمل تجريد البائع

- إعداد تكوين البائع - Stripe/PayPal ومفاتيح API

الوثائق ذات الصلة

أدلة محددة للـ Playbook

- `concepts_ansible.md` - أنماط Playbook العامة والبنية
- `concepts_provisioning.md` - نظرة عامة على نظام التوفير

وثائق نظام الدفع

- **دليل نظام الدفع 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 المعيارية
- `payments_transaction.md` - إدارة المعاملات والرسوم اليدوية
- `payments_process.md` - معالجة المدفوعات والفواتير
- `basics_payment.md` - طرق الدفع وفواتير العملاء

التحكم في الوصول القائم على القواعد

الأدوار والأذونات والمستخدمون في OmniCRM

يستخدم 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 RBAC من:

1. **حماية البيانات** — يرى المستخدمون فقط ما يُسمح لهم برؤيته وفعله.
2. **التوافق التشغيلي** — تعكس الأدوار وظائف العمل (الإدارة، الدعم، المالية، إدارة العملاء).
3. **إدارة بسيطة** — منح الوصول عن طريق تعيين الأدوار؛ تجنب الإدارة الدقيقة لكل مستخدم.
4. **عزل المستأجرين** — تحد من أذونات "عرض خاصتك ..." الرؤية إلى بيانات 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 *الإدارة*، *الدعم*، *المالية*).
- **التعيين** — يتلقى المستخدمون دورًا واحدًا أو أكثر؛ تتجمع الأذونات.

تحدد التفويض (الأدوار/ .(مدرج في القائمة البيضاء IP أو، مفتاح API، JWT) تثبت المصادقة *من أنت* (الأذونات) *ما يمكنك فعله*.

إدارة المستخدمين

يسمح نظام إدارة المستخدمين في OmniCRM للمسؤولين بإنشاء وإدارة مستخدمي الموظفين (المسؤولين، وكلاء ��دمة العملاء)، وعرض وتعديل أدوار المستخدمين، وإعادة تعيين كلمات المرور، وإدارة المصادقة الثنائية، والتحكم في وصول المستخدمين.

أنواع المستخدمين

مستخدمو العملاء - يتم إنشاؤهم عبر التسجيل الذاتي أو من قبل 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 الإدارة، الدعم، المالية، إلخ. يمكن لهؤلاء المستخدمين الوصول إلى واجهة CRM لإدارة العملاء، وتوفير الخدمات، والتعامل مع الفواتير، إلخ.

المستخدمون الإداريون - المستخدمون الذين لديهم إذن `admin`. لديهم وصول كامل إلى النظام بما في ذلك إدارة المستخدمين، وإدارة الأدوار، وجميع نقاط ��لنهاية 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 واحدًا أو أكثر للتعيين (مطلوب)
    - **جهة اتصال العميل** - ربط المستخدم بسجل جهة اتصال العميل (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)

- لجميع الموظفين الإداريين والدعم FAتفعيل 2
- ربط مستخدمي العملاء بسجل جهة الاتصال الخاصة بهم لضمان نطاق البيانا�ت الصحيح

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`

ترجع 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
- الحالة النشطة/المحذوفة
- حالة تفعيل 2FA
- تاريخ آخر تسجيل دخول

الحصول على مستخدم محدد:

`GET /auth/users/{user_id}`

ترجع تفاصيل المستخدم الكاملة بما في ذلك:

- المعلومات الشخصية
- الأدوار المعينة والأذونات الفعالة
- حالة 2FA
- آخر تسجيل دخول ومعلومات الجلسة
- جهة الاتصال المرتبطة بالعميل (إذا كان ذلك مناسبًا)

إنشاء وإدارة الأدوار

الأدوار هي مجموعات من الأذونات التي يمكن تعيينها للمستخدمين. بدلاً من تعيين الأذونات بشكل فردي لكل مستخدم، يمكنك إنشاء أدوار ��جمع الأذونات ذات الصلة وتعيين تلك الأدوار للمستخدمين.

إنشاء دور جديد

عبر واجهة الويب:

1. انتقل إلى **المستخدمين والأدوار** → علامة التبويب **الأدوار**
2. انقر على **"إنشاء دور"**
3. أدخل تفاصيل الدور:
   - **الاسم** - اسم قصير وصفي (مثل "Tier2_Support")
   - **الوصف** - شرح الغرض من الدور والمسؤوليات
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": "Level 2 support team with elevated provisioning
access"
}
```

الاستجابة:

```
{
  "id": 45,
  "name": "Tier2_Support",
  "description": "Level 2 support team with elevated provisioning
access",
  "permissions": [],
  "users": []
}
```

إضافة أذونات إلى دور

بمجرد إنشاء دور، قم بتعيين الأذونات لتحديد ما يمكن أن يفعله المستخدمون الذين لديهم ذا الدور.

عبر واجهة الويب:

1. انتقل إلى **المستخدمين والأدوار** ← علامة التبويب **الأدوار**
2. انقر على اسم الدور لعرض التفاصيل
3. في قسم **الأذونات**، انقر على **"إضافة إذن"**
4. اختر واحدًا أو أكثر من الأذونات 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 إضافة أذونات متعددة:

```
{
  "permission_ids": [123, 124, 125]
}
```

مثال: إنشاء دور "أخصائي توفير"

يمكن لهذا الدور عرض العملاء، وإدارة الخدمات، والتوفير:

1. إنشاء الدور:

```
POST /auth/roles
{
  "name": "Provisioning_Specialist",
  "description": "Can provision services and manage
customer services"
}
```

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. تأكيد 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.

تعديل تفاصيل الدور

تحديث اسم الدور أو الوصف:

عبر واجهة الويب:

1. انتقل إلى **المستخدمين والأدوار** ← علامة التبويب **الأدوار**
2. انقر على الدور للتعديل
3. تعديل اسم الدور أو الوصف
4. انقر على **"حفظ"**

عبر API:

نقطة النه��ية: `PUT /auth/roles/{role_id}`

```
{
  "name": "Senior_Support",
  "description": "Senior support team with full customer access"
}
```

حذف دور

تحذير: يؤدي حذف دور إلى إزالته من جميع المستخدمين المعينين. تأكد من أن المستخدمين لديهم 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 **المستخدمين والأدوار** ← علامة التبويب **المستخدمون**
2. انقر على المستخدم للتعديل
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 للمستخدم 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`

ترجع 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

يوفر OmniCRM طرقًا متعددة لإدارة كلمات المرور حسب السياق.

إعادة تعيين كلمة المرور من قبل المستخدم

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 إلكترون��ا لإعادة تعيين كلمة المرور
3. **تحقق من البريد الإلكتروني** - يحتوي البريد الإلكتروني على رابط إعادة تعيين آمن مع رمز (صالح لمدة ساعة واحدة)
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 حرف 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 وإرسال بريد إلكتروني.

2. **إعادة التعيين باستخدام الرمز:**

   نقطة النهاية: `POST /auth/reset_password`

   ```
   {
     "token": "abc123...",
     "new_password": "NewSecureP@ssw0rd!"
   }
   ```

إعادة تعيين كلمة المرور من قبل 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 **المستخدمين 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 true، يجب على المستخدم تغيير كلمة المرور عند تسجيل الدخول التالي

تغيير المستخدم لكلمة المرور الخاصة به

يمكن للمستخدمين المصرح لهم 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 تجزئة 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 تطبيق متطلبات التعقيد

إدارة المصادقة الثنائية (2FA)

لتعزيز الأمان. يمكن للمسؤولين TOTP الم��ادقة الثنائية المعتمدة على OmniCRM يدعم تفعيل أو تعطيل أو إعادة تعيين 2FA للمستخدمين.

تفعيل 2FA لمستخدم

عبر واجهة الويب:

1. انتقل إلى **المستخدمين والأدوار** ← **المستخدمون**

2. انقر على المستخدم لعرض التفاصيل
3. **تفعيل 2FA"** في قسم **الأمان**، انقر على **"**
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. يقوم المستخدم بحفظ الرموز الاحتياطية في مكان آمن
8. تم تفعيل 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"
}
```

ترجع `{"verified": true}` عند النجاح.

تدفق تسجيل الدخول 2FA

بمجرد تفعيل 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:

- 10 2 رموز تم إنشاؤها أثناء إعداد FA
- للاس��خدام لمرة واحدة فقط (تستهلك بعد الاستخدام)
- تستخدم إذا كان تطبيق المصادقة غير متاح
- يمكن إعادة إنشائها بواسطة المستخدم أو المسؤول

2 التحقق من رمز IFA

نقطة النهاية: `POST /2fa/verify/user/{user_id}`

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

يقبل كلا من:

- **رمز TOTP** (6 أرقام من تطبيق المصادقة)
- **الرمز الاحتياطي** (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",
    ...
  ]
}
```

2 ��إعادة تعيي FA من قبل المسؤول

إذا فقد المستخدم الوصول إلى تطبيق المصادقة الخاص به وجميع الرموز الاحتياطية، يمكن للمسؤول تعطيل وإعادة تفعيل 2FA.

عبر واجهة الويب:

1. انتقل إلى **المستخدمين والأدوار** → **المستخدمون**
2. انقر على المستخدم
3. انقر على زر "**إعادة تعيين 2FA**"
4. تأكيد إعادة التعيين
5. يتم تعطيل 2FA فقط؛ يمكن للمستخدم تسجيل الدخول باستخدام كلمة المرور

6. إرشاد المستخدم لإعداد 2FA مرة أخرى مع سر جديد

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 تفعيل 2FA للحصول على سر جديد ورموز احتياطية.

إعادة تعيين 2FA (جهاز جديد) من قبل المستخدم

إذا حصل المس��خدم على جهاز جديد ولكن لا يزال لديه وصول إلى الرموز 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

- **يتطلب 2FA لجميع موظفي الإدارة والدعم**
- **تخزين الرموز الاحتياطية بأمان** (مدير كلمات المرور أو ملاحظة آمنة)
- **إعادة إنشاء الرموز الاحتياطية** بعد استخدام عدة رموز

- **استخدام تطبيقات المصادقة الموثوقة** (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 **المستخدمين والأدوار** → **المستخدمون**
2. انقر على المستخدم للتعديل
3. تعديل أي حقول قابلة للتعديل:
   - الاسم الأول، الاسم الأوسط، اسم العائلة
   - عنوان البريد الإلكتروني (يتطلب التحقق)
   - رقم الهاتف
   - الأدوار
   - ربط جهة الاتصال بالعميل
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

يستخدم OmniCRM **الحذف الناعم** للمستخدمين - يتم وضع علامة عليهم كمحذوفين ولكن لا يتم إزالتهم من قاعدة البيانات. وهذا يحافظ على سجلات التدقيق والبيانات التاريخية.

حذف مستخدم

عبر واجهة الويب:

1. انتقل إلى **المستخدمين والأدوار** ← **المستخدمون**
2. ابحث عن المستخدم المراد حذفه
3. انقر على زر **"حذف"**
4. تأكيد 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
- تظل البيانات المرتبطة (جهات اتصال العملاء، الأنشطة) محفوظة

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
}
```

ويسمح للمستخدم بتسجيل الدخول مرة أخرى `deleted` هذا يمسح علامة.

ملاحظة: تظل كلمة مرور المستخدم دون تغيير، لذا يمكنه استخدام كلمة المرور السابقة.

حذف مستخدم بشكل دائم

تحذير: هذا غير قابل للتراجع ويزيل جميع بيانات المستخدم من قاعدة البيانات.

لا يتم الكشف عنه عبر واجهة المستخدم. متاح فقط عبر الوصول المباشر إلى قاعدة البيانات (لأغراض الامتثال مثل، طلبات حذف البيانات وفقًا لـ GDPR).

أفضل الممارسات لحذف المستخدمين

- **الحذف الناعم بشكل افتراضي** - يحافظ على سجلات التدقيق
- **توثيق أسباب الحذف** - إضافة ملاحظة في سجل النشاط قبل الح��ف
- **نقل الملكية** - إعادة تعيين تذاكر المستخدم المفتوحة، والمهام قبل الحذف
- **مراجعة الوصول** - التأكد من عدم اعتماد العمليات الحرجة على المستخدم
- **أرشفة البيانات** - تصدير تاريخ عمل المستخدم إذا لزم الأمر
- **إبلاغ الفرق المعنية** - إبلاغ المديرين/الزملاء بالحذف

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 خاصتك ..."** التي تقيد الرؤية إلى بيانات 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 (يتم تدقيقه؛ 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:** *عرض عميل خاصتك*
- **موقع العميل**
  - view_customer_site, create_customer_site, update_customer_site, delete_customer_site
  - **نطاق خاص:** *عرض موقع العميل الخاص بك*
- **جهة اتصال العميل**
  - view_customer_contact, create_customer_contact, update_customer_contact, delete_customer_contact
  - **نطاق خاص:** *عرض جهة اتصال العميل الخاصة بك*
- **سمة العميل** (انظر `Customer Attributes <administration_attributes>`)
  - view_customer_attribute, create_customer_attribute, update_customer_attribute, delete_customer_attribute
  - **نطاق خاص:** *عرض سمة العميل الخاصة بك*
- **علامة العميل** (انظر `Customer Tags <administration_tags>`)
  - view_customer_tag, create_customer_tag, update_customer_tag, delete_customer_tag
  - **نطاق خاص:** *عرض علامة العميل الخاصة بك*
- **خدمة العميل**
  - view_customer_service, create_customer_service, update_customer_service, delete_customer_service
  - **نطاق خاص:** *عرض خدمة العميل الخاصة بك*
- **نشاط العميل**
  - view_customer_activity, create_customer_activity, update_customer_activity, delete_customer_activity
  - **نطاق خاص:** *عرض نشاط العميل الخاص بك*

الفوترة

- **بطاقة Stripe**
  - `view_customer_stripe_card`, `create_customer_stripe_card`, `update_customer_stripe_card`, `delete_customer_stripe_card`
  - **نطاق خاص:** *عرض بطاقة Stripe الخاصة بالعميل*
- **المعاملات**
  - `view_customer_transaction`, `create_customer_transaction`, `update_customer_transaction`, `delete_customer_transaction`
  - **نطاق خاص:** *عرض معاملة العميل الخاصة بك*
- **الفواتير**
  - `view_customer_invoice`, `create_customer_invoice`, `update_customer_invoice`, `delete_customer_invoice`
  - **نطاق خاص:** *عرض فاتورة العميل الخاصة بك*

الاتصالات

- view_communication, create_communication, update_communication, delete_communication
- **نطاق خاص:** *عرض الاتصالات الخاصة بك*

المخزون والقوالب

- **المخزون**
  - `view_inventory`, `create_inventory`, `update_inventory`, `delete_inventory`
  - **نطاق خاص:** *عرض المخزون الخاص بك*
- **قالب المخزون**
  - `view_inventory_template`, `create_inventory_template`, `update_inventory_template`, `delete_inventory_template`

- **نطاق خاص:** *عرض قالب المخزون الخاص بك*

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:** *عرض التوفير الخاص بك*
- **حدث التوفير**
  - `view_provision_event`, `create_provision_event`, `update_provision_event`, `delete_provision_event`

الوصول "عرض خاصتك"

تحدد أذونات "عرض خاصتك ..." نطاق القراءة (وأحيانًا التعديل، حيث تم تنفيذ ذلك) للبيانات المرتبطة بـ **عميلك/مستأجرك الخاص**. على سبيل المثال، يمكن لدور *مدير العملاء* إدارة جهات الاتصال والمواقع والفواتير والخدمات الخاصة بـ *مستأجره*، ولكن لا يمكنهم رؤية مستأجرين آخرين.

تصميمات الأدوار النموذجية

الدور	الأذونات النموذجية	ملاحظات
مسؤول النظام	`admin`، اختياريًا `can_impersonate`؛ واسع حسب الحاجة CRUD بالإضافة إلى	السيطرة الكاملة على المستخدمين/الأدوار/الأذونات
الدعم	`view_customer`، `view_customer_service`، `view_communication`، `view_provision`؛ التحديثات الاختيارية	إضافة `can_impersonate` إذا كان مسموحًا
المالية	`view_customer_invoice`، `view_customer_transaction`، `view_product`؛ `create_customer_invoice` الاختياري	قراءة كثيفة؛ كتابة محدودة
مدير العميل (الم��تأجر)	عرض خاصتك ..." عبر العملاء، المواقع، جهات الاتصال، الخدمات، المخزون، الفواتير، المعاملات، الاتصالات، التوفير	إدارة محددة للمستأجر
المدقق ذو القراءة فقط	`view_*` واسع فقط	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": "Read-only access to example objects"
}
```

قائمة الأدوار

نقطة النهاية: `GET /auth/roles`

إنشاء دور

نقطة النهاية: `POST /auth/roles`

جسم الطلب:

```
{
"name": "Support",
"description": "Tier-1 support team"
}
```

إضافة إذن إلى دور

نقط�� 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 الهوية (مراقب)

- **المطلوب:** `can_impersonate` أو `admin`

بدء انتحال الهوية

نقطة النهاية: `POST /auth/impersonate`

جسم الطلب:

```
{ "user_id": 42 }
```

إيقاف انتحال الهوية

نقطة النهاية: `POST /auth/stop_impersonation`

أفضل الممارس��ت

- **أقل امتياز أولاً.** ابدأ بأدوار الحد الأدنى؛ أضف الأذونات حسب الحاجة.
- **فضل "عرض خاصتك ...".** 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`.

ولوحات المعلومات Grafana تحليلات

يتكامل OmniCRM مع Grafana لتوفير قدرات تحليلية وتصويرية قوية. يمكن للمستخدمين الذين لديهم إذن `grafana_access` الوصول إلى Grafana لإنشاء لوحات معلومات وتقارير وتصويرات مخصصة.

الوصول إلى Grafana

يمكنك الوصول إلى Grafana، إذا تم منح حساب المستخدم الخاص بك إذن `grafana_access`، عن طريق الانتقال إلى `/grafana` باستخدام بيانات اعتماد في متصفحك. سيتم مصادقتك تلقائيًا - لا حاجة لتسجيل دخول منفصل إلى OmniCRM الخاصة بك Grafana.

من واجهة Grafana، يمكنك:

- إنشاء لوحات معلوما�ت مخصصة مع الرسوم البيانية 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 كـ `/grafana/d/abc123/dashboard-name`. المعرف هو الجزء بعد `/d/` (في هذا المثال، `abc123`).

3. **تكوين لوحة المعلومات في بيئتك** - اتصل بمسؤول ال��ظام 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 لـ Grafana
- **تسمية لوحة المعلومات** - الاسم الودود المعروض في الشريط الجانبي لـ OmniCRM

يمكن تكوين لوحات معلومات متعددة وستظهر كعناصر قائمة بالترتيب الذي تم تعريفها به. انظر وثائق تكوين الإدارة للحصول على تفاصيل حول تكوين متغيرات البيئة.

من يمكنه الوصول إلى Grafana؟

يمكنهم الوصول إلى Grafana وعرض `grafana_access` فقط المستخدمون الذين لديهم إذن لوحات المعلومات المضمنة. يتم منح هذا الإذن عادةً لـ:

- المسؤولين الذين يحتاجون إلى رؤية كاملة لمقاييس النظام
- محللي الأعمال الذين ينشئون تقارير وتصويرات
- فرق المالية التي تتعقب الإيرادات ومقاييس الفوترة
- 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 خاصتك ..." ما هو خاصتي؟ تقوم بتقييم الربط بين المستخدم/جهة الاتصال الخاصة بك وعميلك (المستأجر).

هل يمكنني إنشاء لوحات معلومات Grafana الخاصة بي؟ نعم، إذا كان لديك 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
- `<customer_care> تقمص الموظف` → لأغراض استكشاف الأخطاء (من العملاء → "تسجيل الدخول كعميل")

لتفاصيل تقمص الهوية، `تدفقات المصادقة` `<customer_care>` انظر أيضًا: `رعاية العملاء` `<authentication_flows>` 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 الانتهاء
3. **تتبع الاستخدام** - مراقبة البيانات، الصوت، الرسائل القصيرة، والأرصدة المالية
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 مع:

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 الانتهاء
- **الفواتير** - الفوا��ير، المعاملات، طرق الدفع
- **الشحن** - شراء بيانات، صوت، رسائل قصيرة، أو رصيد مالي

تفاصيل الحساب

يعرض قسم الحساب معلومات العميل ويسمح بتحديثات خدمة ذاتية محدودة.

الحقول القابلة للتعديل:

يمكن للعملاء تحديث:

- عنوان البريد الإلكتروني (يتطلب التحقق)
- رقم الهاتف
- كلمة المرور
- تفضيلات الإشعارات

معلومات للعرض فقط:

- معرف العميل
- تاريخ إنشاء الحساب
- نوع العميل (فرد/شركة)
- 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 كلمة 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 "موبايل - +44 7700 900123")
- **المنتج** - اسم الخطة أو المنتج
- **الحالة** - نشط، معلق، منتهي، ملغى
- **تاريخ الإنشاء** - تاريخ تفعيل الخدمة
- **تاريخ الانتهاء** - متى تنتهي الخدمة (إذا كان ذلك مناسبًا)
- **التكلفة الشهرية** - رسوم متكررة
- **التجديد التلقائي** - مؤشر مفعل/معطل

إجراءات الخدمة:

- **عرض الاستخدام** - رؤية استهلاك الرصيد (بيانات، صوت، رسائل قصيرة، مالية)
- **حن** - إضافة رصيد أو بيانات��
- **إضافات** - شراء ميزات إضافية
- **تعديل** - تغيير معلمات الخدمة (إذا كان مسموحًا)
- **عرض التفاصيل** - رؤية تكوين الخدمة بالكامل

مؤشرات حالة الخدمة:

- ✅ **نشط** - الخدمة تعمل
- ⚠️ **ستنتهي قريبًا** - تجدد أو تنتهي في أقل من 7 أيام
- ⏸️ **معلق** - الخدمة معطلة مؤقتًا (مشكلة في الدفع، تعليق يدوي)
- ● **منتهي** - الخدمة لم تعد نشطة

تتبع الاستخدام

يمكن للعملاء مراقبة استخدامهم عبر جميع أنواع الأرصدة في الوقت الفعلي.

استخدام البيانات

عرض استهلاك البيانات مع تفصيلات دقيقة حسب السعة وتاريخ الانتهاء.

عرض استخدام البيانات:

- **إجمالي الرصيد** - جميع سعات البيانات مجتمعة
- **المستخدم في هذه الفترة** - الاستهلاك منذ آخر تجديد
- **شريط التقدم** - تمثيل بصري للاستهلاك
- **م��لومات الانتهاء** - متى تنتهي كل سعة
- **تفصيل السعات** - عدة سعات بيانات مع ترتيب الأولوية

أولوية السعات:

عندما توجد عدة سعات بيانات (مثل: خطة أساسية + شحنات)، يتم استهلاكها حسب ترتيب الوزن:

- **الوزن 10** - يتم استهلاكه أولاً (عادةً بيانات ترويجية/مكافآت)
- **الوزن 20** - يتم استهلاكه ثانيًا (عادةً بيانات شحن)
- **الوزن 30** - يتم استهلاكه أخيرًا (عادةً بيانات الخطة الأساسية)

استخدام الصوت

تتبع دقائق المكالمات المتبقية عبر جميع سعات الصوت.

عرض استخدام الصوت:

- **الدقائق المتبقية** - إجمالي رصيد الصوت
- **الدقائق المستخدمة** - الاستهلاك في هذه الفترة
- **سجل المكالمات** - المكالمات الأخيرة (إذا كانت مفعلة)
- **تواريخ الانتهاء** - متى تنتهي سعات 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 بأسعار خاصة

استخدام الرسائل القصيرة

مراقبة حدود الرسائل القصيرة والاستهلاك.

عرض استخدام الرسائل القصيرة:

- **الرسائل المتبقية** - رصيد الرسائل القصيرة
- **المستخدمة هذا الشهر** - الرسائل المرسلة
- **مشمولة** - ما إذا كانت الرسائل متعددة الوسائط تحتسب ضمن الرصيد **MMS**
- **الرسائل القصيرة الدولية** - تتبع منفصل (إذا كان ذلك مناسبًا)

الرصيد المالي

عرض رصيد الائتمان المدفوع مسبقًا لخدمات الدفع حسب الاستخدام.

عرض الرصيد المالي:

- **الرصيد الحالي** - الائتمان المتاح
- **آخر شحن** - مقدار الشحن الأخير وتاريخه
- **تاريخ الانتهاء** - متى ينتهي الرصيد (إذا كان ذلك مناسبًا)
- **إعادة الشحن التلقائي** - حالة مفعل/معطل

إدارة الفواتير

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** - الموعد النهائي للدفع
- **المبلغ** - إجمالي مبلغ الفاتورة
- **الحالة** - مدفوعة، غير مدفوعة، متأخرة
- **الإجراءات** - تنزيل PDF، الدفع عبر الإنترنت

دفع فاتورة:

1. انتقل إلى **الفواتير** ← **الفواتير**
2. ابحث عن الفاتورة غير المدفوعة في القائمة
3. انقر على زر **"ادفع الآن"**
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 PDF الكاملة مع تفاصيل الفاتورة
3. محفوظة للسجلات الضريبية والوثائق

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 تشمل 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 طريقة الدفع الأساسية
- **إزالة بطاقة** - حذف بطاقات منتهية أو غير مستخدمة
- **تفاصيل البطاقة** - عرض آخر 4 أرقام، تاريخ الانتهاء، علامة البطاقة

انظر `طرق الدفع` `<payment_methods>` لوثائق إدارة الدفع التفصيلية.

الشحن / إعادة الشحن

شراء بيانات إضافية، صوت، رسائل قصيرة، أو رصيد مالي 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:

- **شحنات البيانات** - خيارات 1GB، 5GB، 10GB، 20GB، 50GB
- **شحنات الصوت** - دقائق مكالمات إضافية
- **حزم الرسائل القصيرة** - حزم الرسائل
- **رصيد مالي** - رصيد مدفوع مسبقًا (£5، £10، £20، £50، £100)

لتفاصيل سير العمل الخاصة `<features_topup_recharge>` انظر `الشحن وإعادة الشحن` بالشحن.

إضافات الخدمة

شراء ميزات إضافية وتحسينات للخدمات الحالية.

الإضافات المتاحة:

- **التجوال الدولي** - تفعيل الخدمة في الخارج
- **عنوان IP ثابت** - IP ثابت للإنترنت المنزلي
- **محتوى مميز** - قنوات IPTV، خدمات البث
- **ترقيات الأجهزة** - إيجار 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

لتفاصيل إدارة الإضافات. `<csa_modify>` انظر `تعديل الخدمات - الإضافات`

الإشعارات والتنبيهات

يتلقى العملاء إشعارات آلية للأحداث المهمة:

إشعارات البريد الإلكتروني:

- تم إنشاء الفاتورة وجاهزة للدفع
- تأكيد استلام الدفع
- تحذيرات انتهاء الخدمة (7 أيام، 3 أيام، 1 يوم)
- تنبيهات انخفاض الرصيد (بيانا��، صوت، مالي)
- تفعيل/تعطيل الخدمة
- طلبات إعادة تعيين كلمة المرور
- تنبيهات الأمان (تسجيل الدخول من جهاز جديد)

تنبيهات داخل البوابة:

- فواتير غير مدفوعة
- خدمات على وشك الانتهاء
- تحذيرات انخفاض البيانات (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 عند بقاء أقل من 20% من البيانات)
- فئات الإشعارات (الفوترة، الاستخدام، الخدمة)

قيود الخدمة الذاتية

بعض العمليات تتطلب مساعدة من الموظفين:

يتطلب خدمة العملاء:

- تغيير نوع العميل (فرد ↔ شركة)
- نقل الخدمات بين العملاء
- إلغاء الخدمات (قد تكون خدمة ذاتية إذا كانت مفعلة)
- الطعن ف�� الفواتير
- طلب استرداد الأموال
- تغيير جهة الاتصال الأساسية
- مشاكل توفير معقدة

قيود الأمان:

- لا يمكن عرض أو تعديل حسابات المستخدمين الآخرين
- محدود للبيانات الخاصة بالعميل (عزل المستأجر)
- لا يمكن الوصول إلى الوظائف الإدارية
- لا يمكن إلغاء الفواتير أو المعاملات
- لا يمكن تعديل تكوين الخدمة (فقط الإضافات/الشحنات)

وصول 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 كعميل"**
5. تفتح علامة تبويب جديدة 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** - تأكيد أن تكوينات الخدمة تظهر بشكل صحيح
- **التدريب** - عرض ميزات البوابة
- **الدعم** - إرشاد العميل عبر البوابة أثناء عرض شاشته

الأمان والتدقيق:

- يتطلب إذن `can_impersonate` أو `admin`
- يتم تسجيل جميع الإجراءات في سجل التدقيق
- يرى العميل التقمص في سجل النشاط
- انتهاء جلسة بعد inactivity
- لا يمكن تغيير كلمة مرور العميل أثناء التقمص

انظر `<customer_care>` لوثائق التقمص الكاملة `رعاية العملاء - تقمص المستخدم`.

استجابة الهاتف المحمول

بوابة الرعاية الذاتية متجاوبة تمامًا ومهيأة للأجهزة المحمولة.

ميزات الهاتف المحمول:

- تنقل محسّن للمس
- تخطيطات مبسطة للشاشات الصغيرة
- إيماءات السحب للتنقل عبر الكاروسيل
- نما��ج ومدخلات صديقة للهاتف المحمول
- مسح رمز QR لتوفير eSIM
- الاتصال الهاتفي بنقرة واحدة
- تكامل GPS لإكمال العناوين

تطبيق ويب تقدمي (PWA):

- تثبيت كتطبيق على الشاشة الرئيسية
- عرض البيانات الأخيرة بدون 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 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 لمدة 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 (!@#$%^&*)

إغلاق 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 للتجديدات التلقائية
5. مراقبة الاستخدام بانتظام لتجنب رسوم الاستخدام الزائد
6. حفظ رموز النسخ الاحتياطي في مكان آمن (إذا كانت المصادقة الثنائية مفعلّة)
7. تسجيل الخروج بعد استخدام أجهزة الكمبيوتر المشتركة/العامة

إدارة الاستخدام:

1. تفعيل تنبيهات انخفاض الرصيد (10-20% المتبقية)
2. الشحن 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

وصول API

يمكن للعملاء الذين لديهم متطلبات تقنية استخدام API مباشرة.

تو ليد مفتاح API:

متاح للعملاء التجاريين أو عند الطلب:

1. **وصول** ← **الحساب** انتقل إلى **API**
2. **تو ليد مفتاح "** انقر على **API"**
3. تعيين الأذونات (للقراءة فقط أو للقراءة والكتابة)
4. تعيين تاريخ الانتهاء
5. حفظ مفتاح API (يظهر مرة واحدة فقط) بشكل آمن

حالات استخدام API:

- مراقبة الاستخدام الآلي
- التكامل مع أنظمة الفوترة الداخلية
- الشحن البرمجي
- توفير الخدمة عبر السكربتات
- تصدير البيانات للتحليلات

انظر `وثائق API <concepts_api>` لتفاصيل النقاط النهائية والأمثلة.

الأسئلة المتكررة

س: لماذا لا أستطيع رؤية جميع خدماتي؟

ج: تأكد من أنك مسجل الدخول باستخدام الحساب الصحيح. إذا كان لديك حسابات عملاء متعددة، فإن لكل منها خدمات منفصلة. اتصل بالدعم لدمج الحسابات إذا لزم الأمر.

س: فشلت عملية الدفع لكن تم خصم المبلغ. ماذا أفعل؟

ج: تحقق من كشف حسابك البنكي للرسوم المعلقة. إذا ظهر الخصم ولكن لا تزال الفاتورة 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 من خدمة العملاء لأسباب أمنية وامتثال.

س: لماذا ينخفض رصيد البيانات الخاص بي بشكل أسرع مما هو متوقع؟

ج: تحقق من تحديثات التطبيقات في الخلفية، جودة بث الفيديو، والنسخ الاحتياطي التلقائي على السحابة. راجع تفصيل الاستخدام في البوابة للحصول على استهلاك مفصل.

س: فقدت جهاز المصادقة الثنائية الخاص بي. كيف يمكنني استعادة الوصول؟

ج: استخدم رموز النسخ الاحتياطي لتسجيل الدخول، ثم قم بإيقاف وتفعيل المصادقة الثنائية مرة أخرى. 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 يكون دفع ��لفاتورة كزائر متاحًا 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. للحصول على تنزيلات جماعية، اتصل بالدعم أو استخدم API إذا كان متاحًا.

الوثائق ذات الصلة

- `<customer_care> رعاية العملاء - تقمص المستخدم` - وصول موظفي الدعم لاستكشاف الأخطاء
- `<payment_methods> طرق الدفع` - إدارة البطاقات والدفع
- `<features_topup_recharge> الشحن وإعادة الشحن` - شراء رصيد إضافي
- `<csa_service_usage> استخدام الخدمة` - فهم تتبع الرصيد
- `<billing_overview> نظرة عامة على الفوترة` - مفاهيم الفوترة والفواتير

- تسجيل الدخول والأمان - `<authentication_flows> تدفقات المصادقة`
- إعداد المصادقة الثنائية - `<2fa> المصادقة الثنائية`
- إضافة ميزات وإضافات - `<csa_modify> تعديلات الخدمة`

دليل عمليات OmniCRM

OmniCRM هو الحل الشامل من Omnitouch لإدارة أنظمة الأعمال (BSS) وإدارة العمليات (OSS) لمقدمي خدمات الهاتف المحمول والثابت. منصة كاملة تدير 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. خطوات إنشاء عميل خطوة بخطوة - **إنشاء عميل**
3. زود أول خدمة لك - **إضافة خدمة**
4. التعامل مع مدفوعات العملاء - **معالجة الدفع**
5. إضافة رصيد إلى حسابات العملاء - **إعادة شحن الخدمات**

المهام اليومية:

- إدارة خدمات العملاء - **إدارة الخدمات**
- التحقق من الأرصدة والاستخدام - **عرض الاستخدام**
- تغيير إعدادات الخدمة - **تعديل الخدمات**
- إنشاء وإرسال الفواتير - **توليد الفواتير**

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. **إعداد المستخدمين** - إنشاء حسابات وتعيين أدوار

المواضيع المتقدمة:

- **إدارة المخزون** - إدارة الأصول
- **التخصيص** - تخصيص حسب احتياجاتك
- **تكامل API** - ربط الأنظمة الخارج��ة
- **إعداد الأمان** - تكوين المصادقة الثنائية والأذونات

للعملاء

استخدام بوابة الخدمة الذاتية:

- **الوصول إلى حسابك** - تسجيل الدخول والتنقل
- **عرض الخدمات** - رؤية خدماتك النشطة
- **التحقق 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
معالجة الدفع	معالجة الدفع
إدارة الفواتير	توليد 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
البحث العالمي	البحث عن أي شيء
هيكل النظام	فهم النظام
وثائق API	استخدام API

مكتبة الوثائق

القراءة الأساسية

- **هيكل النظام** - نظرة تقنية شاملة مع الرسوم البيانية
- **دليل دورة حياة المنتج** - إدارة المنتج 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

عمليات الخدمة

- إدارة الخدمة - نظرة عامة
- إضافة خدمات - سير عمل التزويد
- تعيين الخطط - تعيين الخطط
- تعديل الخدمات - إجراء التغييرات
- استخدام الخدمة والأرصدة - مراقبة الاستخدام
- إعادة الشحن - إضافة أرصدة

الفوترة والمالية

- نظرة عامة على الفوترة - دليل الفوترة الشامل
- طرق الدفع - إدارة طرق الدفع
- المعاملات - الرسوم والائتمانات
- معالجة المدفوعات - سير عمل المدفوعات
- إدارة الفواتير - توليد وإدارة الفواتير

الإدارة

- تكوين - تكوين النظام OmniCRM
- تخصيص - التخصيص ليتناسب مع عملك
- تتبع الأصول - إدارة المخزون
- إدارة الوصول إلى API - مفاتيح API

الأمان والوصول

- كيفية عمل المصادقة - تدفقات المصادقة
- إعداد المصادقة الثنائية - المصادقة الثنائية
- المستخدمون، الأدوار، والأذونات - التحكم في الوصول المعتمد على الدور

التكاملات

- معالجة المدفوعات - تكامل Stripe
- خدمة البريد الإلكتروني - تكامل Mailjet
- مرجع API REST - وثائق API

واجهة العملاء

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

المرجع

- المصطلحات والتعريفات - المعجم
- تاريخ الإصدارات - سجل التغييرات

ما الذي يجعل OmniCRM مميزًا؟

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

أتمتة تزويد الخدمة

لتزويد الخدمات في ثوانٍ. سواء كنت **أتمتة Ansible** انسَ التكوين اليدوي - يستخدم OmniCRM أو تكوين معدات العملاء، أو إعداد حزمة معقدة، يتولى النظام ذلك تلقائيًا، تقوم بتفعيل بطاقة SIM.

ما الذي يفعله:

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

تعرف على المزيد حول التزويد Ansible | شاهد دليل كتب تشغيل

الفوترة الذكية والمدفوعات

Stripe تقييمًا وتحميلًا في الوقت الفعلي، بينما يتولى **تكامل CGRateS** يوفر التكامل المدمج مع معالجة المدفوعات بسلاسة.

ما الذي يفعله:

- تتبع الاستخدام والتقييم في الوقت الفعلي
- توليد الفواتير تلقائيًا
- معالجة بطاقات الائتمان عبر Stripe
- نماذج الفوترة المدفوعة مسبقًا وما بعد الدفع
- تسعير مرن ورموز ترويجية
- حساب الضرائب والتقارير

استكشف ميزات الفوترة | معالجة المدفوعات | إدارة الفواتير

إدارة العملاء الشاملة

قم بإدارة العملاء، وجهات الاتصال، والمواقع المتعددة، والخدمات من خلال نموذج علاقة شامل.

ما ا��ذي يفعله:

- 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 خرائط Google لتحديد المواقع الجغرافية

دليل إدارة العملاء | **إنشاء عميل** | **علامات العملاء**

إدارة المخزون

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

ما الذي يفعله:

- تتبع بطاقات SIM، والملحقات، والمعدات
- إدارة أرقام الهواتف وكتل عناوين IP
- تعيين تلقائي أثناء التزويد
- قدرات استيراد وتصدير جماعي
- تخزين تكوين المعدات
- قوالب مخزون مخصصة

وثائق نظام المخزون

بوابة الخدمة الذاتية للعملاء

مكّن عملاءك من خلال **بوابة الخد��ة الذاتية** حيث يمكنهم إدارة خدماتهم بأنفسهم.

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

تم بناءه مع الأمان في جوهره، ويتميز بمصادقة وتفويض شاملة.

ميزات الأمان:

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

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

تكاملات قوية

يتكامل OmniCRM بسلاسة مع الخدمات الرائدة في الصناعة:

- **CGRateS** - محرك الفوترة والتقييم في الوقت الفعلي للتحميلات من الدرجة التليفونية
- **Stripe** - معالجة المدفوعات الآمنة وتخزين بطاقات الائتمان
- **Mailjet** - تسليم البريد الإلكتروني الاحترافي مع القوالب
- **خرائط Google** - التحقق من العناوين وتحديد المواقع الجغرافية لمواقع دقيقة
- **RESTful API** - دمج مع أنظمتك الحالية

وثائق API | **تكامل Stripe** | **تكامل Mailjet**

كتالوج المنتجات المرن

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

أنواع المنتجات المدعومة:

- **مس��قل** - 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 أعمال.

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

نظام البث الطارئ

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

ما الذي يفعله:

- إرسال تنبيهات الطوارئ إلى المناطق الجغرافية
- دعم الرسائل متعددة اللغات
- استهداف بواسطة مشغل الشبكة المحمول
- الامتثال لمعايير تنبيهات الحكومة

وثائق البث الخلوي

بحث وتنقل قوي

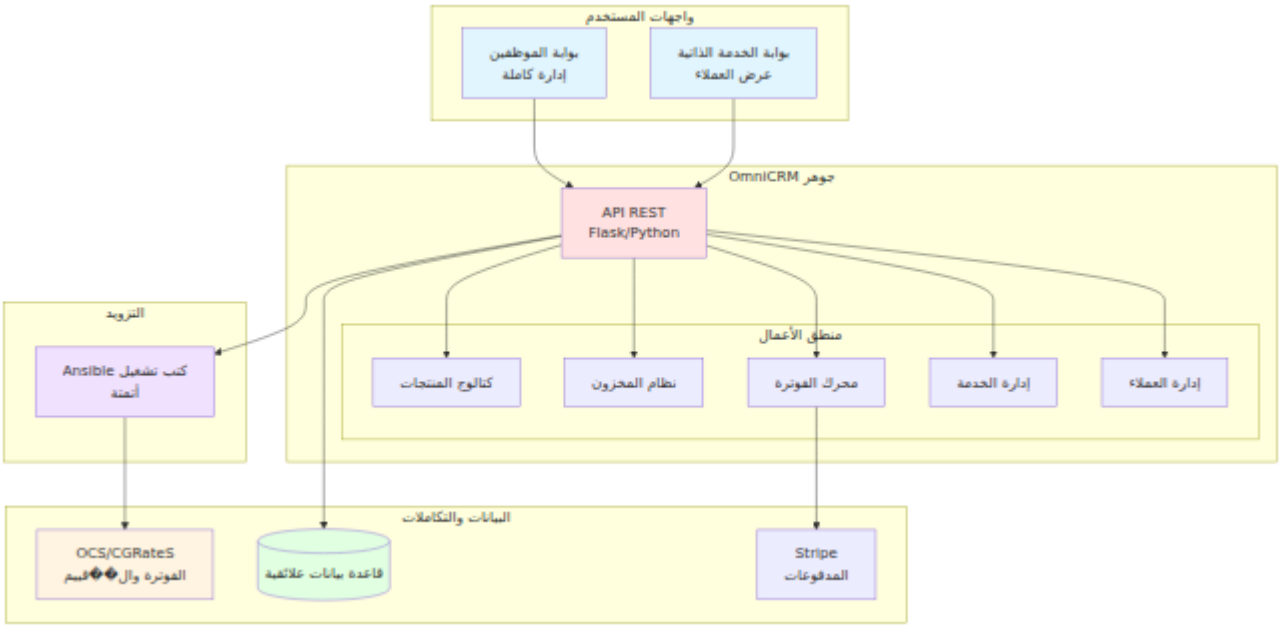
ابحث عن أي شيء على الفور باستخدام **البحث العالمي** عبر العملاء، والخدمات، والفواتير، وأكثر.

ميزات البحث:

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

دليل التنقل | **البحث العالمي**

هيكل النظام



عرض وثائق الهيكل الكامل

المفاهيم الأساسية

فهم المنتجات مقابل الخدمات

هذه واحدة من أهم المفاهيم في OmniCRM:

- **المنتج** = قالب أو عرض في الكتالوج الخاص بك (مثل، "خطة الهاتف المحمول غير المحدودة")
- **الخدمة** = حالة نشطة من منتج لعميل محدد (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.

عرف 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 حسابات في إرسال رسائل الترحيب عبر البريد الإلكتروني. وإذا فشل أي شيء، فإنه يتراجع تلقائيًا.

نظام التزويد | Ansible غوص عميق: كتب تشغيل

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

تسلسل العملاء

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

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 بأ��وات 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

OmniCRM من تعريفات المنتجات إلى كتب تشغيل التزويد إلى السمات المخصصة - خصص ليتناسب مع عمليات عملك الدقيقة.

مصمم للاتصالات

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

أمان المؤسسات

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

API مدفوع بـ

مع أي أنظمة موجودة أو بناء أدوات OmniCRM الشاملة أنه يمكنك دمج API REST تضمن مخصصة.

احصل على الدعم

موارد الوثائق

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

هل 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