

OmniCRM (2FA)

OmniCRM 2FA 設定方法

OmniCRM は RBAC (2FA) をサポートしています (TOTP) を使用してログイン時に追加の認証を行います。

RBAC <rbac> の 2FA を有効にするには Authentication Flows <authentication_flows> を設定する必要があります。

手順

2FA を有効にする

1. ユーザー — ユーザー設定画面を開く
2. ユーザー — ユーザー設定画面を開く
3. ユーザー — ユーザー設定画面を開く
4. ユーザー — ユーザー Google Authenticator、Authy、Microsoft Authenticator をインストールして QR コードをスキャンして TOTP を生成

2FA 設定方法

OmniCRM 2FA の設定

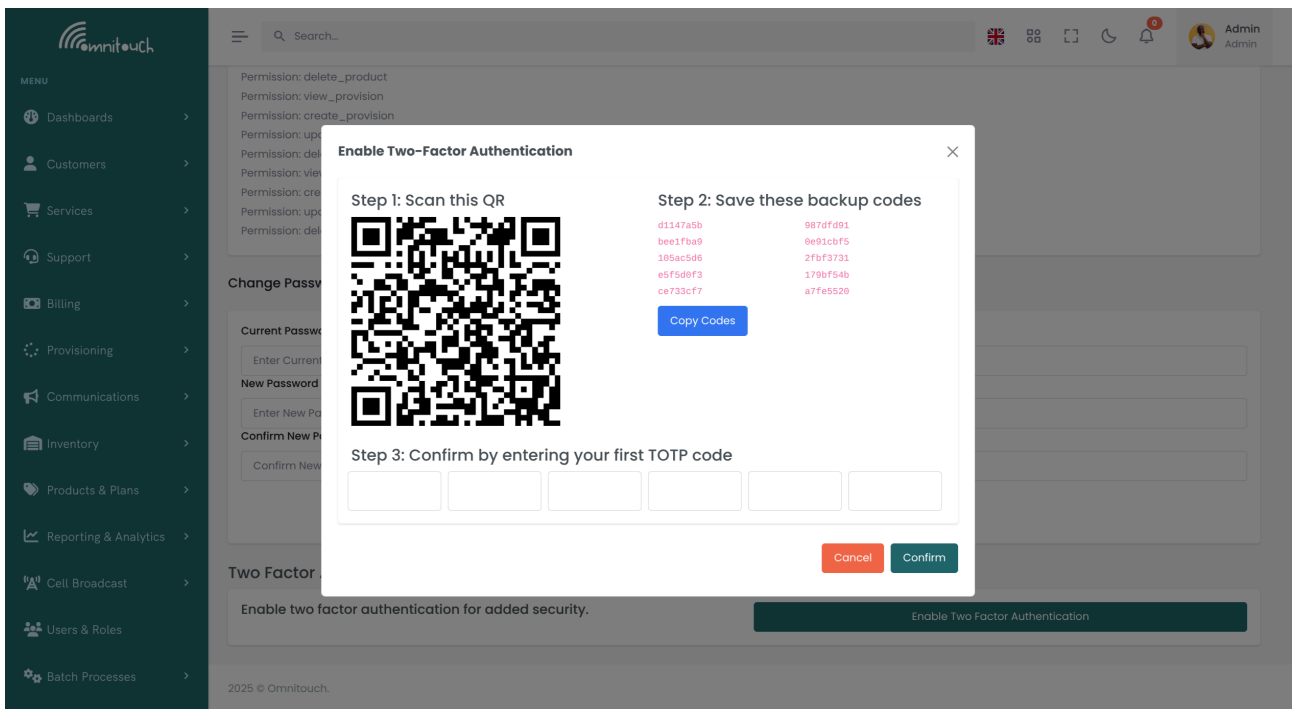
1. ユーザー — ユーザー設定画面を開く
2. ユーザー — ユーザー設定画面を開く 6 桁のコードを入力
3. ユーザー — ユーザー設定画面を開く
4. ユーザー — ユーザー設定画面を開く

2FA

2FA

2FA

1. 2FA
2. 2FA
3. 2FA
4. 2FA
5. 2FA



2FA

2FA

2FA

- 2FA
- 2FA

- 2FA 2FA

2FA 2FA

2FA 2FA
totp_secret
2FA 2FA

API

2FA

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

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

URI

2FA

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

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

2FA

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

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


API 概念

API 是应用程序编程接口，是 **Web** 应用程序的重要组成部分。OmniCRM API 是 API 的

概念

API 的概念包括 `concepts_api`

API

API 是应用程序编程接口，是

- 应用程序编程接口
- 应用程序
- 应用程序
- 应用程序 cron 任务
- 应用程序

JWT 令牌/应用程序API 令牌

API 概念

API

API

API `MANAGE_API_KEYS` 令牌

API 概念

API 概念

API

- API 令牌“令牌”“令牌”

- 〇〇〇〇 - 〇〇〇〇〇〇〇〇〇〇〇〇
- **API** 〇〇 - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- 〇〇 - 〇〇〇〇〇〇〇〇〇〇
- 〇〇〇〇 - 〇〇〇〇〇〇〇〇
- 〇〇〇〇 - 〇〇〇〇〇〇〇〇〇〇〇〇
- 〇〇 - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇

〇〇〇〇〇

〇〇〇〇〇〇〇

〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇

- 〇 **API** 〇〇 - 〇〇 API 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- 〇〇〇〇 - 〇〇〇〇〇〇〇〇
- 〇〇〇〇 - 〇〇〇〇〇〇 30 〇〇〇〇〇〇〇〇
- 〇〇〇〇 - 〇〇〇〇〇〇〇〇〇〇

〇〇 **API** 〇〇

〇〇 **1**〇〇〇“〇〇 **API** 〇〇”

〇〇 API 〇〇〇〇〇〇〇〇 + 〇〇 〇〇〇

〇〇 **2**〇〇〇〇〇〇〇〇

〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇

〇〇〇 _____
〇〇〇〇“〇〇〇〇〇〇”

〇〇〇 _____
〇〇〇 - 〇〇〇〇〇〇〇〇

〇〇〇〇〇〇〔〇〇〇〇〇〇〕
〇〇〇 - 〇〇〇〇〇〇〇〇〇〇

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

[□□] [□□□□]

□□□□

□□□□□□

- □□□□□□□□□□
- □□□“□□□”□“□□□□”□“□□□”
- □□□□□□□□□□□□□□

□□□□□□

- □□□□□□
- □□□“□ Ansible □□□□□□□□□□”□“□□□□□□□□□□”
- □□□□□□□□□□□□□□□□□□

□□□□□□□□

- □□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□90 □□ 1 □□

□□

- □□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□
- □□□□□□ □□□□□□□□□□

□□ **3**□□□□□□□□□□

□□ “□□□□□” □□□□□□□□□□□□□□ API □□□□

⚠ □□□□□□□□□□ - □□□□□□□□□□

sk_live_a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0

[□□□□□□□□]

[]

API

sk_live_...XYZ

4

API

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

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

API

-
-
-
-
-
-

API 调用

API 调用步骤

1. 获取 API 地址
2. 使用 “API 地址”
3. 调用

注意

API 调用失败

API 调用失败原因

原因

- API 地址错误
- API 调用频率过高
- API 调用超时
- API 调用失败

API 调用

API 调用

1. 获取 API 地址
2. 使用 “API 地址”
3. 调用

注意

- API 地址错误
- API 调用频率过高 **401 Unauthorized**
- API 调用超时
- API 调用失败 - 调用失败

原因

- API 地址错误

- 000000
- 00000000000000
- 00000000000000

00 **API** 000000

000 API 00000000

1. 0000000 : 00000
2. 00 “00”
3. 00000000000000
4. 00 “0000”

000000

- 00
- 00
- 0000
- 00

00000

- 0000000000000000
- 0000
- 00000

API 0000

API 0000000000

00

- 0000000000
- 0000000000000000
- 00000
- 0000000

□□□□

- □□□□□□□□ 30 □□□□
- □□□□□/□□□□
- □□□□□□□□□□

□□

- □□□□□□
- □□□□□□
- □□□□□□□
- □□□□□□□□□□

□□□

- □□□□□/□□
- □□□□□
- □□□□□□□□□□

□□□□□□

API □□□□□□□□

□□□

□□□□□□□□□□□□□□□□

□□□□□□

□□□□□□□□□□

- □□□□□
- □□□□
- □□□□□□□□□□ 30 □□
- □□

□□□

API 키 관리

- API 키
- API 키
- API 키
- API 키

API 키

API 키

- API 키 32 바이트
- API 키
- API 키 sk_live_

API 키

CRM

- API 키
- API 키
- API 키
- API 키

API 키

- API 키
- API 키 AWS Secrets Manager HashiCorp Vault
- API 키
- API 키 90-365

API 키

- API 키 - API 키
- API 키

- 認證/授權
- 認證

認證

- API 認證
- 認證
- “認證”
- 認證

認證

認證

1. 認證
2. 認證
3. 認證
4. 認證

認證

API 認證“401 Unauthorized”

- 認證
- 認證
 - 認證
 - 認證
 - 認證
 - 認證 X-API-KEY 認證 Authorization

認證“API 認證”

- 認證
- 認證
 - API 認證

- 0000000000000000
- 0000000000

API 00000000

- 000 000000000030 000
- 000000
 - 0000000000000000
 - 0000000000000000
 - 000000000000

00000 API 00

- 000 000000000000
- 000000
 - 000000000000
 - 000000000000/00
 - 000000000000???

API 0000000000000000

API 00000000 API 0000000000000000

00 API 00

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

00 API 00

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

```
{
  "name": "New Integration",
  "description": "Third-party billing sync",
  "expiry_date": "2026-01-10",
  "permissions": ["view_customer", "view_service"]
}
```

□□□

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

□□ **API** □□

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

□□ **API** □□

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

```
{
  "name": "Updated Name",
  "expiry_date": "2026-12-31"
}
```


4. API 概念

概念

- `concepts_api` - API 概念
- `rbac` - 權限管理
- `2fa` - 雙重認證

- `regulatory_entity = "FCC-123456"`

4. `Account`

`Account`

- `preferred_contact_method = "email"`
- `account_manager = "<john.smith@company.com>"`
- `onboarding_date = "2025-01-15"`
- `churn_risk_score = "0.23"`

5. `Radius`

`Radius`

- `radius_username_format = "email%RADIUS"`
- `vlan_id = "100"`
- `ipv6_enabled = "true"`

`Radius`

`Radius`

- `radius_username_format = "email%RADIUS"`
- `vlan_id = "100"`
- `ipv6_enabled = "true"`
- `radius_username_format = "email%RADIUS"`
- `vlan_id = "100"`

`Radius`

- `radius_username_format = "email%RADIUS"`
- `vlan_id = "100"`
- `ipv6_enabled = "true"`
- `radius_username_format = "email%RADIUS"`

UI 說明

說明

說明

- 說明
- 說明
- 說明
 - 說明
 - 說明
 - 說明
 - 說明

說明

說明

- 說明
- 說明
- 說明
- 說明
 - 說明 external_crm_id
 - 說明 SF-12345
- 說明

說明

- 說明 external_system_id ✓
- 說明 external system id ✗
- 說明
- 說明

說明

說明

POST create_customer_attribute

Request

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

Response

```
{
  "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"
}
```

PATCH

PATCH /crm/attribute/attribute_id/{attribute_id}

POST update_customer_attribute

Request

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

GET ID

GET /crm/attribute/attribute_id/{attribute_id}

POST view_customer_attribute

□□□

```
{
  "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"
}
```

□□□□**ID**□□□□□□

□□□ GET /crm/attribute/customer_id/{customer_id}

□□□□□ view_customer_attribute

□□□

```
[
  {
    "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"
  }
]
```


API 호출

API 호출을 위한 Python 코드 예시

```
# API 호출을 위한 Python 코드 예시
customers = requests.get("https://api.example.com/crm/customer/").json()

for customer in customers:
    attributes = requests.get(
        f"https://api.example.com/crm/attribute/customer_id/{customer['customer_id']}"
    ).json()

    # Premium 고객 찾기
    for attr in attributes:
        if attr['attribute_name'] == 'support_tier' and
        attr['attribute_value'] == 'premium':
            print(f"Premium customer: {customer['customer_name']}")
```

결과

API 호출 시 발생할 수 있는 에러

요청

1. 헤더

- `external_system_id` ✓
- `billing_method` ✓ `method` ✗
- `customer_id`
- `customer_name`

2. 본문

- `customer_id` 150 이하
- `customer_name` "true"/"false" 형식
- `timestamp` ISO 8601 형식 "2025-01-04 10:30:00"
- `data` JSON 형식

3. []

- []
- []
- []

4. []

- []
- []/[]
- []

5. []

- []
- []
- [] `api_key_v2` []

[][][]

[][]

[][][]

```
# 更新 legacy_crm_id 属性
PUT /crm/attribute/
{
  "customer_id": 123,
  "attribute_name": "legacy_crm_id",
  "attribute_value": "OLD-12345"
}
```

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

更新属性

更新属性

```
# 更新 billing_cycle 属性
PUT /crm/attribute/
{
  "customer_id": 123,
  "attribute_name": "billing_cycle",
  "attribute_value": "quarterly"
}

# 获取 billing_cycle 属性
attributes = GET /crm/attribute/customer_id/123
billing_cycle = next(
    (a['attribute_value'] for a in attributes if
     a['attribute_name'] == 'billing_cycle'),
    'monthly' # 默认值
)
```

更新属性

更新属性

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

API

API endpoints

- `view_customer_attribute` - GET
- `create_customer_attribute` - POST
- `update_customer_attribute` - PUT
- `delete_customer_attribute` - DELETE

API uses `rbac` for permissions

UI

UI components

- API endpoints
- UI components
- `view_customer_attribute` UI

UI components

- `update_customer_attribute` UI
- `attribute_id` UI
- UI components

UI components

- UI components

OmniCRM

OmniCRM 環境構築ガイド

目次

OmniCRM 環境構築

環境構築

- `OmniCRM-API/crm_config.yaml`
- YAML
- API
- 環境構築

環境構築

- `OmniCRM-UI/.env`
- 環境構築
- UI
- 環境構築

環境構築 (crm_config.yaml)

`crm_config.yaml` 環境構築

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

Database

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

MySQL

- `username` - MySQL 用户名
- `password` - MySQL 密码
- `server` - 数据库服务器 IP 地址 localhost

连接字符串

- 数据库名称 `omnicrm`
- 端口号 3306 MySQL 默认
- 连接字符串 `mysql+pymysql://username:password@server/omnicrm`

数据库连接字符串格式: `mysql+pymysql://username:password@server/omnicrm`

在 `.env` 文件中配置数据库连接信息 `db` Docker 容器

Service Types

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

数据库连接字符串

数据库连接字符串

- 数据库

- 配置 HSS 代理
- 配置 HSS 代理
- 配置 HSS 代理

配置 HSS 代理

HSS 配置

配置 HSS 代理

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

配置

- `hss_peers` - 配置 HSS 代理
- `apn_list` - 配置 APN ID 列表

配置 HSS 代理

Mailjet 配置

OmniCRM 配置 Mailjet 代理

```
mailjet:
  api_key: your_mailjet_api_key
  api_secret: your_mailjet_api_secret

# 00000000
api_crmCommunicationCustomerWelcome:
  from_email: "support@yourcompany.com"
  from_name: "Your Company Support"
  template_id: 5977509
  subject: "Welcome to YourCompany"

# 00000000
api_crmCommunicationCustomerInvoice:
  from_email: "billing@yourcompany.com"
  from_name: "Your Company Billing"
  template_id: 6759851
  subject: "Your Invoice - "

# 00000
api_crmCommunicationCustomerInvoiceReminder:
  from_email: "billing@yourcompany.com"
  from_name: "Your Company Billing"
  template_id: 5977570
  subject: "Invoice Payment Reminder"

# 0000000000/00000
api_crmCommunicationUserWelcome:
  from_email: "admin@yourcompany.com"
  from_name: "Your Company Admin"
  template_id: 6118112
  subject: "Welcome to the Team"

# 00000000
api_crmCommunicationUserPasswordReset:
  from_email: "security@yourcompany.com"
  from_name: "Your Company Security"
  template_id: 6735666
  subject: "Password Reset Request"

# 0000000000
api_crmCommunicationUserPasswordResetSuccess:
  from_email: "security@yourcompany.com"
  from_name: "Your Company Security"
```

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

```
# 日本語
```

```
api_crmCommunicationUserPasswordChange:
  from_email: "security@yourcompany.com"
  from_name: "Your Company Security"
  template_id: 6118423
  subject: "Password Changed"
```

```
# 英語
```

```
api_crmCommunicationEmailVerification:
  from_email: "verify@yourcompany.com"
  from_name: "Your Company Verification"
  template_id: 6267350
  subject: "Verify Your Email Address"
```

```
# 日本語
```

```
api_crmCommunicationsBalanceExpired:
  from_email: "alerts@yourcompany.com"
  from_name: "Your Company Alerts"
  template_id: 7238252
  subject: "Service Balance Expired"
```

```
# 英語
```

```
api_crmCommunicationsBalanceLow:
  from_email: "alerts@yourcompany.com"
  from_name: "Your Company Alerts"
  template_id: 7238263
  subject: "Low Balance Warning"
```

Mailjet

1. Mailjet (<https://app.mailjet.com>)
2. **Transactional** → **Templates**
- 3.
4. **Template ID**
5. OmniCRM
6. `crm_config.yaml` ID

Variables

- `{{customer_name}}` - Customer name
- `{{service_name}}` - Service name
- `{{invoice_id}}` - Invoice ID
- `{{invoice_amount}}` - Invoice amount
- `{{due_date}}` - Due date
- `{{reset_link}}` - Reset URL
- `{{verification_link}}` - Verification URL
- `{{balance}}` - Balance
- `{{expiry_date}}` - Expiry date

Example

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

Example

- `failure_list` - Ansible list of email addresses
- ...
- ...

Example

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

Example

Example PDF generated from Jinja2 HTML

Example `/OmniCRM-API/invoice_templates/`

Example PDF generated from Jinja2 HTML

CRM URL

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

URL

- Ansible playbook API URL
- Ansible playbook CRM URL
- Ansible playbook API URL

URL

- `http://localhost:5000`
- `https://api.yourcompany.com`
- Docker `http://omnicrm-api:5000`

URL

- `http://localhost:5000`
- `https://api.yourcompany.com` URL
- `http://omnicrm-api:5000`

OCS CGRates

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

URL

- `ocsApi` - OCS API URL
- `ocsTenant` - OCS Tenant
- `cgrates` - CGRates JSON-RPC API URL

URL

SMSC 設定

```
smc:  
  source_msisdn: 'YourCompany'  
  smc_url: 'http://10.179.2.216/SMSc/'  
  api_key: 'your_smc_api_key'
```

設定

- SMS 送信に必要の 2FA 設定
- `source_msisdn` - 送信元 ID
- `smc_url` - SMSC API URL
- `api_key` - SMSC API キー

設定

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

設定 CBC API 接続

```
features_cell_broadcast
```

JWT 設定

```
jwt_secret: 'CHANGE_ME_ON_DEPLOYMENT' # Ansible 設定
```

設定

- JWT 生成
- 署名 - 検証
- 署名/検証
- 署名

設定 Ansible 設定

Ansible JWT secrets

1. Generate Ansible secrets 64 characters long
2. Add Ansible secrets to the code
3. Deploy the code - secrets are generated

Steps

- Git repository
- Ansible playbook
- Ansible inventory
- Ansible roles

Ansible secrets

Ansible secrets

```
# Generate secrets
python3 -c "import secrets; print(secrets.token_hex(32))"
# Save secrets
openssl rand -hex 32
```

crm_config.yaml

```
jwt_secret: 'your_generated_secret_here'
```

Notes

- Change `CHANGE_ME_ON_DEPLOYMENT`
- Ansible secrets - secrets are generated
- Ansible inventory
- Ansible roles

Stripe 設定

```
stripe:  
  secret_key: 'sk_live_XXXXXXXXXX'  
  publishable_key: 'pk_live_XXXXXXXXXX'  
  currency: 'aud'  
  statement_descriptor_suffix: 'YOURCOMPANY'
```

設定

- `secret_key` - Stripe API キー
- `publishable_key` - Stripe キー
- `currency` - ISO 4217 コード (aud, usd, gbp, eur)
- `statement_descriptor_suffix` - 明細書接尾辞

テスト

- `sk_test_...` / `pk_test_...`
- `sk_live_...` / `pk_live_...`

注意事項

- 明細書接尾辞 "YOURCOMPANY"
- 最大 22 文字
- 英数字のみ
- PDF 名: YOURCOMPANY_12345.pdf

Payment Vendor Integrations
<integrations_payment_vendors>

API IP 設定

API IP 設定

```

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"]

# IP 白名单 API 接口
ip_whitelist:
  "10.179.2.142":
    roles: ["admin"]

```

接口

- 配置 API 接口
- IP 白名单
- API 接口
- 配置 API 接口

API 接口

```
openssl rand -base64 48
```

接口

- `admin` - 配置 API 接口
- RBAC 配置 API 接口

配置 API 接口

- 配置 API 接口 32 接口
- IP 白名单
- 配置 API 接口
- API 接口
- API 接口

配置 API 接口

- 本地 IP 地址
- 本地 IP 地址 127.0.0.1::1
- 本地 API 地址
- 本地端口

本地 IP 地址 `administration_api_keys` 和 `concepts_api`

环境变量 (.env)

React UI 项目 `OmniCRM-UI/.env` 文件

路径 `/OmniCRM/OmniCRM-UI/.env`

API 配置

```
# Google Maps API 配置
VITE_GOOGLE_API_KEY=your_google_api_key

# Stripe 配置
VITE_STRIPE_PUBLISHABLE_KEY=pk_test_xxxxx

# 本地 npm start 配置
BROWSER=none
```

配置

`VITE_STRIPE_PUBLISHABLE_KEY` 配置 `crm_config.yaml` 文件
`stripe.publishable_key` 配置

环境变量

```
# 公司
VITE_COMPANY_NAME="ShellFone"
VITE_PORTAL_NAME="ShellManager"
VITE_SELF_CARE_NAME="ShellCare"
VITE_COMPANY_TAGLINE="Phones with Shells"
```

环境变量 UI 配置

- `COMPANY_NAME` - 公司名称
- `PORTAL_NAME` - 门户名称 "ShellManager"
- `SELF_CARE_NAME` - 自助服务名称 "ShellCare"
- `COMPANY_TAGLINE` - 公司标语

环境变量

```
# 语言
# 支持语言: ar ch en fr gr it ru sp
VITE_DEFAULT_LANGUAGE=en
VITE_LOCALE="en-GB"

# 默认位置
VITE_DEFAULT_LOCATION="Sydney, Australia"
VITE_DEFAULT_COUNTRY="Australia"

# 货币
VITE_CURRENCY_CODE="GBP"
VITE_CURRENCY_SYMBOL="£"
```

语言支持

- `ar` - 阿拉伯语
- `ch` - 中文
- `en` - 英语
- `fr` - 法语
- `gr` - 希腊语

- `it` - `it`
- `ru` - `ru`
- `sp` - `sp`

Moneda

- USD - `$`
- GBP - `£`
- EUR - `€`
- AUD - `$`
- CAD - `$`

Se define en `crm_config.yaml` el `stripe.currency`

Colores

```
# Color primario
VITE_PRIMARY_COLOR=#405189

# Color secundario
# VITE_SECONDARY_COLOR=#2bFFcf
# VITE_TERTIARY_COLOR=#1a9fbf
# VITE_SUCCESS_COLOR=#28a745
# VITE_INFO_COLOR=#17a2b8
# VITE_WARNING_COLOR=#ffc107
# VITE_DANGER_COLOR=#dc3545
```

Definición

- `VITE_PRIMARY_COLOR` - `#405189`
- `VITE_SECONDARY_COLOR` - `#2bFFcf`
- `VITE_TERTIARY_COLOR` - `#1a9fbf`
- `VITE_SUCCESS_COLOR` - `#28a745`
- `VITE_INFO_COLOR` - `#17a2b8`
- `VITE_WARNING_COLOR` - `#ffc107`
- `VITE_DANGER_COLOR` - `#dc3545`
- `VITE_LIGHT_COLOR` - `#f8f9fa`

- `VITE_DARK_COLOR` - `#343a40`
- `VITE_PRIMARY_DARK_COLOR` - `#343a40`

`#RRGGBB`

`#RRGGBB`

- `#RRGGBB`
- `#RRGGBB`
- `#RRGGBB`
- `#RRGGBB`
- `#RRGGBB`

`#RRGGBB`

```
# #RRGGBB
# Inter Roboto Open Sans Lato Quicksand Poppins Nunito
Montserrat Work Sans Source Sans Pro Raleway Ubuntu Josefin Sans
HKGrotesk
# Merriweather Lora Playfair Display
# System
# Quicksand
VITE_FONT_FAMILY=Quicksand
```

OmniCRM-UI

- `#RRGGBB` - `#RRGGBB`
- `#RRGGBB` - `#RRGGBB`
- `#RRGGBB` - `#RRGGBB`
- `#RRGGBB` - `#RRGGBB` Google Fonts Adobe Fonts CDN
- `#RRGGBB` - `#RRGGBB`
- `#RRGGBB` - `#RRGGBB`

`#RRGGBB`

`#RRGGBB`

- Inter, Roboto, Open Sans, Lato, Quicksand, Poppins, Nunito, Montserrat, Work Sans, Source Sans Pro, Raleway, Ubuntu, Josefin Sans, HKGrotesk

Font families

- Merriweather, Lora, Playfair Display

Font families

- System - default font

Font families

Font families

Font families

1. Create a folder `OmniCRM-UI/src/assets/fonts/your-font-name/`
 - Add WOFF2 font files
 - Add font weights (300, 400, 500, 600, 700)
 - Name files `your-font-name-300.woff2` and `your-font-name-400.woff2`
2. Add `OmniCRM-UI/src/assets/scss/fonts/_google-fonts.scss` with `@font-face`

```
//
// 日本語 - 西
//

@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. `.env` 設定

```
VITE_FONT_FAMILY=Your Font Name
```

設定

- 300 - 太字
- 400 - 標準
- 500 - 太字
- 600 - 太字
- 700 - 太字

Web 設定

6 設定 Web 設定

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

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

# Web 3-6
# NAME URL ICON_PATH
```

- `VITE_WEB_APP_N_NAME` -
- `VITE_WEB_APP_N_URL` - URL
- `VITE_WEB_APP_N_ICON_PATH` - public/

GitHub Xero Monday.com Gmail MailJet Slack

Grafana

OmniCRM Grafana `grafana_access` `/grafana`
Grafana OmniCRM

```
# Grafana
# ID - /grafana/d/ID/name URL
VITE_GRAFANA_DASHBOARD_IDS=abc123,def456,ghi789

# ID
VITE_GRAFANA_DASHBOARD_LABELS=System Metrics,User Analytics,Network Performance
```

- `VITE_GRAFANA_DASHBOARD_IDS` - URL Grafana ID
- `VITE_GRAFANA_DASHBOARD_LABELS` - ID

Dashboard ID

Grafana dashboard URL: `/grafana/d/abc123/dashboard-name` ID: `/d/abc123`

Example

OmniCRM "Sales" dashboard ID: `OmniCRM-Sales`

Example

URL: `http://your-server/grafana/d/system-overview-123/system-metrics`

- Dashboard ID: `system-overview-123`
- Dashboard name: "Sales"

Example

- ID: `system-overview-123`
- Dashboard name: "Sales"
- Environment: `grafana_access`
- UI: `UI`

Environment Variables

```
# Environment Variables
VITE_FAQS_URL=https://support.yourcompany.com/faqs
VITE_SUPPORT_URL=https://support.yourcompany.com
```

Example: `UI`

Example

```
# Environment Variables
VITE_ALLOW_SOCIAL_LOGINS=yes
```

□□

- **yes** - □□□□□□□□Google□Facebook □□
- **no** - □□□□□□

□□ □□□□□□□□□□□□

□□□□□□

```
# Google Analytics 4 □ Google Tag Manager□□□□  
# □□ GA4□□□□□□G-XXXXXXXXXX  
# □□ GTM□□□□□□GTM-XXXXXXX  
VITE_GOOGLE_TAG_ID=G-XXXXXXXXXX
```

□□ □□ Google Analytics 4□GA4□□ Google Tag Manager□GTM□□□□□□□□□□□□□□□□□□
□□□□□

□□□□□□

- **GA4** □□ ID□ G-XXXXXXXXXX - □□ Google Analytics 4 □□
- **GTM** □□ ID□ GTM-XXXXXXX - □□ Google Tag Manager
- □□□ □□□□□□□□□□□

□□□□□

□□□□□ ID □□□□□□□□□□□□

- □ **G-** □□□ ID □□□ Google Analytics 4
- □ **GTM-** □□□ ID □□□ Google Tag Manager
- □□□□□□□□□□□□□

□□□

- □□□□□□ - □□□□□□□□□□□□□□□□
- □□□□□□□□ - □□□□□□□□□□□□□□□□□
- □□□□□□ - □□□□□□□□□□□□ □□□ UI

□□□□□

1. Measurement ID

Google Analytics 4

- Google Analytics
- GA4
- Admin → Data Streams
- Web
- Measurement ID: G-XXXXXXXXXX

Google Tag Manager

- Google Tag Manager
-
- Container ID: GTM-XXXXXXX

2. .env

```
VITE_GOOGLE_TAG_ID=G-XXXXXXXXXX
```

3. UI

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

4. Deployment

-
-
- Google Analytics

Deployment -

-

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

// checkout
const handleCheckout = () => {
  trackEvent('checkout_initiated', {
    cart_value: 99.99,
    item_count: 3,
    page: '/cart'
  });
  // checkout
};
```

00 - checkout

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

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

00 - form_submission

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

// pageview
trackPageView('/virtual/page-path', 'Page Title');
```

00000000

Method	Parameters	Return
<code>initializeGoogleTracking()</code>	Tracking ID []	void
<code>trackEvent(eventName, eventParams)</code>	Event Name Event Params	void
<code>trackPageView(path, title)</code>	Page Path Page Title	void

Implementation

Google Analytics Implementation

- **GDPR** - General Data Protection Regulation
- **CCPA** - California Consumer Privacy Act
- **Analytics** - Google Analytics
- **Cookie** - Cookie Consent
- **Integration** - Google Analytics Integration

Configuration

Environment	Configuration	Implementation
GA Tracking ID	Tracking ID	GA ID: G-XXXXXXXXXX GTM-XXXXXXXX
Environment	Environment ID	GA .env configuration
UI	UI .env	Run <code>npm run build</code> to build UI
Environment	Environment	Configuration
Environment	GA Tracking ID	Configuration

Implementation

- 環境変数 VITE_GOOGLE_TAG_ID を追加
- 環境変数 < 50KB を追加
- 環境変数 VITE_GOOGLE_TAG_ID を追加
- 環境変数 VITE_GOOGLE_TAG_ID を追加

環境変数

- 環境変数 OmniCRM-UI/src/utils/googleAnalytics.js
- 環境変数 OmniCRM-UI/src/utils/usePageTracking.js
- 環境変数 App.js と index.js を追加

環境変数

VITE_TOPUP_PRICE_PER_DAY=10

環境変数 VITE_TOPUP_PRICE_PER_DAY/環境変数 VITE_TOPUP_PRICE_PER_DAY

環境変数

- 環境変数 VITE_TOPUP_PRICE_PER_DAY - 環境変数 VITE_TOPUP_PRICE_PER_DAY

環境変数 VITE_TOPUP_PRICE_PER_DAY 10 環境変数 VITE_TOPUP_PRICE_PER_DAY USD 環境変数 VITE_TOPUP_PRICE_PER_DAY \$10 環境変数 VITE_TOPUP_PRICE_PER_DAY

環境変数 VITE_TOPUP_PRICE_PER_DAY 環境変数 VITE_TOPUP_PRICE_PER_DAY features_topup_recharge

環境変数

OmniCRM 環境変数 VITE_TOPUP_PRICE_PER_DAY/環境変数 VITE_TOPUP_PRICE_PER_DAY

環境変数

環境変数 /OmniCRM-UI/src/assets/images/omnitouch/ 環境変数 VITE_TOPUP_PRICE_PER_DAY

環境変数

- `DefaultLogoDark.png` - 100x100px
- `DefaultLogoLight.png` - 100x100px

100x100px

- `logoSm.png` - 100x100px
- `logoDark.png` - 200x50px
- `logoLight.png` - 200x50px

100x100px

100x100px

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

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

100x100px

- **logoSm.png** - 100x100px
- **logoDark.png** - 200x50px
- **logoLight.png** - 200x50px

100x100px

1. 100x100px

- PNG 100x100px
- 100x100px
- 100x100px

2. OmniCRM

```
# 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. UI

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

4. Assets

- logoDark.png
- logoLight.png
- logoSm.png
- logoLight.png

Assets

- -
- -
- - PNG
- - DPI 2
- -

Assets

/OmniCRM-UI/src/pages/AuthenticationInner/authCarousel.js

```

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

// ████████████████████
<!-- ![Company Logo](../absolute/path/to/your-logo.png) -->
    <h1>INVOICE</h1>
  </div>

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

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

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

```

```

</table>

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

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

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

```

📌 Notes

- 📁📁📁📁📁 - `file:///absolute/path/to/image.png`
- 📄 **CSS** - WeasyPrint 📄📄📄📄📄📄
- 📄📄📄📄📄 - 📄 `invoice_templates/rendered/` 📄 HTML
- 📄📄 - 📄 `<div style="page-break-after: always;"></div>` 📄📄📄📄
- 📄📄📄 - 📄 `@page` CSS 📄📄📄📄
- 📄📄📄 - 📄 Jinja2 📄📄📄📄 `{{ "%.2f"|format(amount) }}`

📌 Steps to Run

1. 📄📄📄📄📄

```

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

```

2. 設定

- 設定ファイルの作成
- 設定ファイルの読み込み
- 設定ファイルのバックアップ

3. 設定

例 `crm_config.yaml`

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

4. 実行

- CRM 実行
- PDF 生成
- 例 `invoice_templates/rendered/{invoice_id}.html` 生成

5. テスト

実行コマンド

```
-- テスト PDF生成  
DELETE FROM Invoice_PDF_Cache;
```

PDF 生成

OmniCRM から PDF 生成

実行

- 実行 - PDF 生成
- 実行 - テスト PDF 生成
- 実行 - テスト PDF 生成
- 実行 - 30 分間隔で実行

□□□□

- Base64 □□□ PDF □□□ Invoice_PDF_Cache □□
- SHA256 □□□□□□□□□□
- □□□□□□□□□□□□□□□□

□□□□□□

```
# □ OmniCRM API □ 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()

# □□□□ 30 □□□□
result = cleanup_old_pdf_cache(session, days_old=30)
print(result) # {'status': 'success', 'deleted_count': 15}

# □□□□□□□□□□
invalidate_invoice_cache(session, invoice_id='12345')
```

API □□□

□□/□□□□ PDF□

GET /invoice/pdf/{invoice_id}

□□□ PDF □□□□□□□□□□ Stripe □□□□□

□□□□

- □□□□□□□□□□□□□□□□
- □□□□□□□□□□
- □□□□/□□□□□□□□□□□□



□□□□□□

□□ (crm_config.yaml)

1. □□ OmniCRM-API/crm_config.yaml
2. □□□□
3. □□ API □□□

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

□□□□□□□□□□

□□ (.env)

1. □□ OmniCRM-UI/.env
2. ❓❓□□□
3. □□ UI□

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

4. □□ UI □□□ Web □□□

□□□□□

□□□ npm start □□□□□□□□□□□□□□□□□□□□□□



□□□□□□

□□□

- □□□□□□ - □□□□□□□□□□□□ `.gitignore`
- □□□□□ - □□ 16 □□□□□□□□□□□□□□□□
- □□□□□□ - □□□□□□□□□□
- □□□□□□□□ - □□ IP □□□□□□□□□□
- □□□□□□□□□□ - □□□□/□□□□/□□□□□
- □□ **API** □□□□ - □□□□□□□□□□
- □□□□ **IP** □□□□ - □□□□□□ API □□□□□□□□□□

□□

- □□□□ - □□□□□□□□□□□□□□
- □□□□ - □□□□□□□□□□□□□□
- □□□□□□□□□□ - □□□□□□□□□□□□□□
- □□□□ - □ git □□□□□□□□□□□□□□□□

□□

- □□□□□□□□ - □□□□□□□□□□□□□□
- □□□□ - □□□□□□□□ OCS □□
- □□ **Grafana** - □□□□□□□□□□□□

□□

- □□□□ - □□ UI □□□□□□□□□□
 - □□□□□ - □□□□□□□□□□□□□□□
 - □□□□ - □□□□□□□□□□□□
 - □□□□ - □□□□□□□□□□□□□□□□
-

UI

UI

UI

- UI UI
- UI API/UI

YAML

- YAML
- YAML `yamllint crm_config.yaml`

-
-

Stripe

-
- `publishable_key`

- Mailjet ID
- Mailjet API / ID

PDF

- WeasyPrint `pip install weasyprint`
- `crm_config.yaml`
- `invoice_templates/rendered/` HTML
- API Jinja2

PDF

- `file:///full/path/to/image.png`
- `file:///full/path/to/image.png`
- `file:///full/path/to/image.png` PNG `file:///full/path/to/image.png` JPEG `file:///full/path/to/image.png`
- `file:///full/path/to/image.png`

UI

- CSS `flexbox` `grid`
- CSS `flexbox` `grid`
- `div`

API

- `invalidate_invoice_cache()`
- `Invoice_PDF_Cache`

Config

OmniCRM `crm_config.yaml`

Env

- `.env.example` `.env`
- `CGRates`
- `crm_config.yaml`
 - `Mailjet API ID`
 - `CRM URL`

- OCS/CGRates
- SMSC
- JWT
- Stripe
- API IP

UI

- OmniCRM-UI/.env.example → OmniCRM-UI/.env
- Google Maps API
- Stripe
- -
 -
 -
- -
 -
 -
-
- Web
- URL
- Google Analytics/Tag Manager ID

Assets

- logoSm.png logoDark.png logoLight.png
- OmniCRM-UI/src/assets/images/omnitouch/
- HTML
- OmniCRM-API/invoice_templates/
- crm_config.yaml
- PDF

- UI `npm run build`

UI

- Authentication
- JWT
- Stripe
- Mailjet API
- Localization
- API IP
- SSL/TLS
- HTTPS
- CORS
- Security
- Performance

Backend

- Authentication
- Authorization
- Database
- PDF
- Stripe
- 2FA
- Localization
- OCS
- ActionPlan
- Security

Infrastructure

- Docker
- MySQL/PostgreSQL
- CGRates

- OmniCRM API
 - OmniCRM UI
 - nginx traefik
 - Grafana Prometheus
 -
 -
 -
 -
-

API

- RBAC
- concepts_products_and_services
- Ansible
- administration_inventory
- payments_invoices
- 2fa
- customer_care
- integrations_payment_vendors
- administration_api_keys - API
- integrations_mailjet -
- concepts_api - API

OmniCRM

OmniCRM

IP GPON ONT CPE

Customer Attributes <administration_attributes>

Customer Tags <administration_tags>

CPE

inventory_items_list

UI Ansible Complete Product Lifecycle Guide - Inventory

Requirements <guide_product_lifecycle>

OmniCRM

1. SIM
- 2.
- 3.
4. GPON CPE

SIM

OmniTouch 1000 SIM SIM SIM SIM SIM SIM SIM SIM SIM SIM

SIM SIM SIM SIM SIM SIM SIM SIM SIM - SIM
SIM

SIM SIM SIM SIM SIM SIM SIM SIM SIM SIM

SIM SIM SIM SIM

GPON

GPON GPON GPON GPON GPON GPON

GPON GPON GPON GPON GPON GPON

GPON GPON GPON GPON GPON GPON

OmniCRM GPON GPON GPON GPON GPON GPON GPON GPON

Inventory

InventoryTemplate InventoryTemplate InventoryTemplate InventoryTemplate

- MAC itemtext1 itemtext2
-
- itemtext1 itemtext2
- IP

20 itemtext1 itemtext20
itemtext1_label itemtext20_label
itemtext1_label "MAC" itemtext2_label "

InventoryTemplates InventoryTemplates InventoryTemplates
InventoryTemplates

- 資料をすべて取得してデータを取得する

取得

データベースからデータを取得する

取得

- 項目**1**をデータベースから取得する項目を“項目”に
 - 項目を“ICCID”項目SIM項目“MAC項目”項目項目項目“項目”項目
- 項目**2**をデータベースから取得する項目を“項目”に
 - 項目を“IMSI”項目SIM項目“項目”項目項目項目“項目”項目
- 項目**3-20**をデータベースから取得する項目を**◆◆**項目
 - 項目 項目 項目項目項目
 - 項目項目項目項目項目項目

取得

- 項目項目項目項目項目項目項目項目
- 項目項目項目項目項目項目/項目項目項目項目項目

4. 項目 項目 項目項目

取得

取得

1. 項目 項目 → 項目
2. 項目項目項目
3. 項目 項目 項目
4. 項目項目項目
5. 項目 項目

取得

項目項目項目項目 itemtext1_label 項目項目項目項目項目項目項目項目項目項目項目項目項目

取得


項目 inventory_items_list 項目項目項目項目項目項目項目項目項目項目項目項目項目

□□□□□□

SIM□□□□

- □□□“SIM□”
- □□□□□2.50
- □□□□□10.00
- □□□□1□□□“ICCID”
- □□□□2□□□“IMSI”
- □□□□3□□□“SIM□□”□□□/□□□□□

□□□□□□□□

- □□“□□□□”
- □□□□□1.00
- □□□□□0.00
- □□□□1□□□“□□□□”
- □□□□2□□□“□□□□”
- □□□□3□□□“□□□□”□□□/□□□□□

□□□□CPE□□□□

- □□□“□□□□CPE”
- □□□□□250.00
- □□□□□450.00
- □□□□1□□□“MAC□□”
- □□□□2□□□“□□□□”
- □□□□3□□□“□□□□”
- □□□□4□□□“□□□□”
- □□□□5□□□“□□□□”

GPON ONT□□□□

- □□□“GPON ONT”
- □□□□□45.00
- □□□□□0.00□□□□□□□□□□

- 1 " "
- 2 "MAC "
- 3 "PON "
- 4 " "

Inventory

InventoryTemplate

-
-
- `sold_date`

OmniCRM

Ansible

Inventory

Inventory Item

Attributes

- **inventory_id** - Inventory ID
- **item** - Item name "SIM" "Phone" "Accessory"
- **inventory_template_id** - Inventory Template ID
- **customer_id** - Customer ID
- **service_id** - Service ID
- **item_location** - Item Location
 - Location Name
 - IP Address
- **item_state** - Item State
 - State
 - Item - Inventory
 - Item - Service
 - Item - Customer
 - Item - Location
 - Item - Service
 - Item - Customer
 - Item - Location

Inventory Item

Inventory Item 20

- **itemtext1** - Item Text 1 `itemtext1_label`
 - Item Text 1 Label "MAC"
 - Item Text 1 Label "SIM" "ICCID"
 - Item Text 1 Label "IMEI"
- **itemtext2** `itemtext20` - Item Text 2
 - Item Text 2 Label "IMEI"
 - Item Text 2 Label "IMSI"

- `itemtext3` “ ”

`itemtext1_label` `itemtext2_label`

- **wholesale_cost** - /
- **retail_cost** -
- **sold_date** -

CPE ONT

- **address_line_1** -
- **address_line_2** -
- **city** -
- **state** -
- **zip_code** - /
- **country** -

Web UI

Web UI

- **google_maps_place_id** - Google ID
- **plus_code** - Google
- **latitude** -
- **longitude** -

-
-
-
-

管理 URL

management_url 管理 URL

- **management_url** - 管理 URL
 - Web URL `https://192.168.1.1` ONT CPE
 - **eSIM** eSIM LPA `LPA:1$smdp.example.com$ACTIVATION-CODE-HERE`
 - 管理 URL

管理

management_url 管理 URL

- 128x128 URL
- 64x64
-

管理

- IP
- **eSIM** CRM eSIM
-

- **management_username** - 管理
- **management_password** - 管理

管理

管理

- **config_content** - 管理
 -
 - CPE
- **config_file_path** - 管理
 -
 -

Inventory

- **inventory_notes** - Inventory notes
 - Note 1
 - Note 2
 - Note 3
 - Note 4
 - Note 5
- **created** - Date and time when the inventory was created
- **last_modified** - Date and time when the inventory was last modified

Example

Item 1: SIM Card

```
{
  "inventory_id": 1001,
  "item": "SIM Card",
  "inventory_template_id": 5,
  "itemtext1": "8961234567890123456",
  "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"
}
```

Item 2: eSIM

```
{
  "inventory_id": 1002,
  "item": "eSIM",
  "inventory_template_id": 6,
  "itemtext1": "8961234567890123457",
  "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"
}
```

eSIM UI LPA eSIM

3 CPE -

```
{
  "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°"
}
```

4 GPON ONT

- 34 IP 5


```
{
  "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"
}
```

item_state

item_state

- → -
- → -
- → -
- → - RMA
- → -
- → -
- / → -

-
-
-
-
-

- 
-
-
-

1.
2.
3.
4.
 -
 -
 - URL
 -
 - 2099-01-01
5.
6.

□□□□

□□□□□□□□

1. □□□□□□□□□□
2. □□ □□ □□□
3. □□□□□□□□□□□□□□
4. □□ □□□□□□□□
5. □□□□□□□□□
6. □□ □□□□

□□□□

□□□□□□

1. □□□□□□□□□□
2. □□ □□ □□□
3. □□□□□□□□□□□□□□
4. □□ □□□□□□□□
5. □□□□□□□□□□

□□□□□□□□

API □□

□□□□□□□ API □□□□□□□□□□□□

□□□□□□

```
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
}
```

□□□□□

```
PATCH /crm/tag/tag_id/{tag_id}
{
  "tag_text": "Updated Tag Text",
  "tag_hex_color": "FF0000"
}
```

□□□□□□□□

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

□□□□□

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


□□□□□□

- □□□□□□□□
- □□□□□□□□□□
- □□□□□□□□
- □□□ UI □□□□□□□□□□□□

2. API □

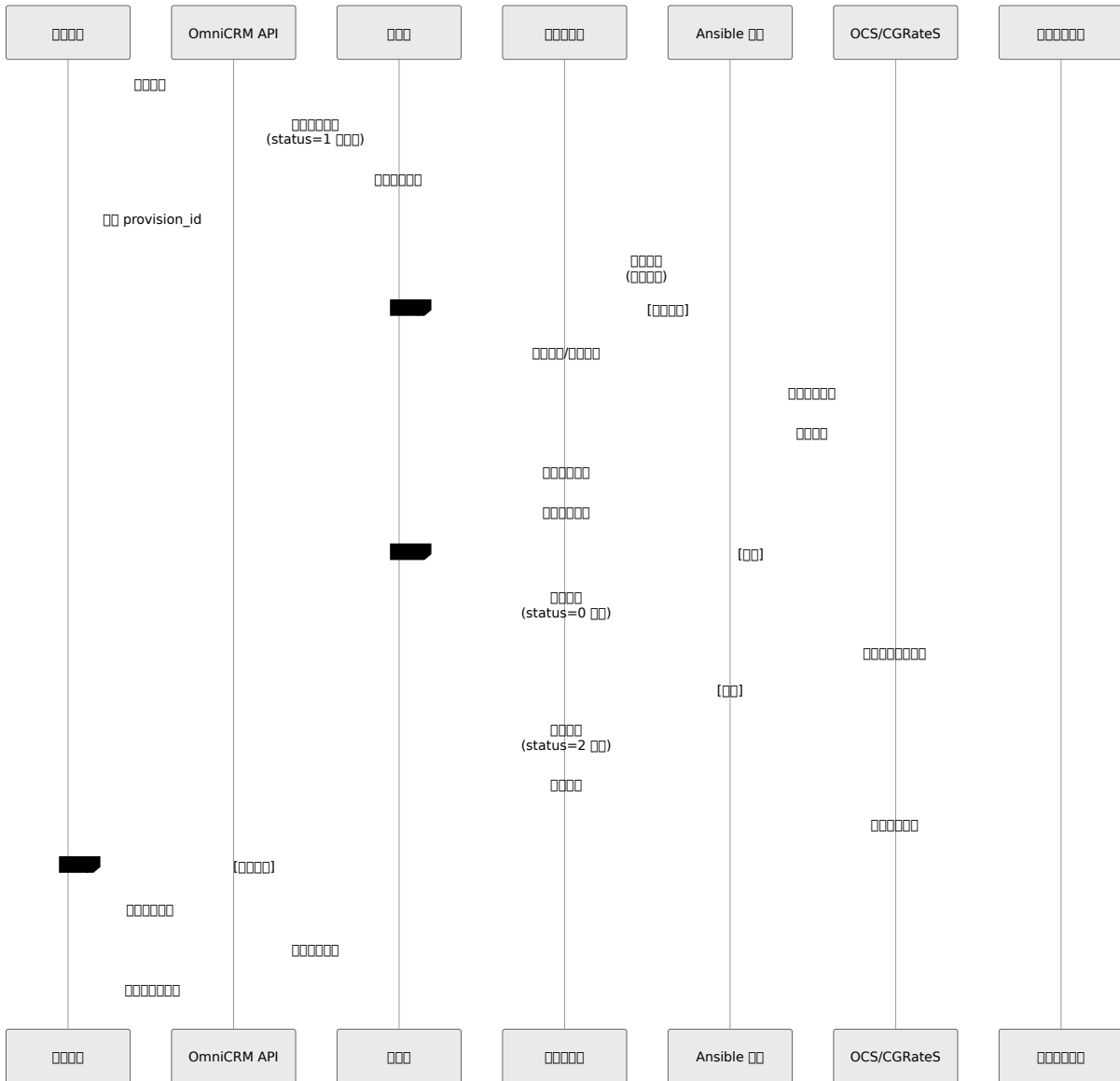
OmniCRM API (Flask/Python)

- □□□□□ RESTful API
- OpenAPI/Swagger □□
- □□ JWT □□□
- □□□□□□□□
- WebSocket □□□□□□

□□ **API** □□:






- `/crm/customer/*` - □□□□
- `/crm/service/*` - □□□□
- `/crm/product/*` - □□□□
- `/crm/provision/*` - □□□□
- `/crm/transaction/*` - □□□□
- `/crm/invoice/*` - □□□□

3.

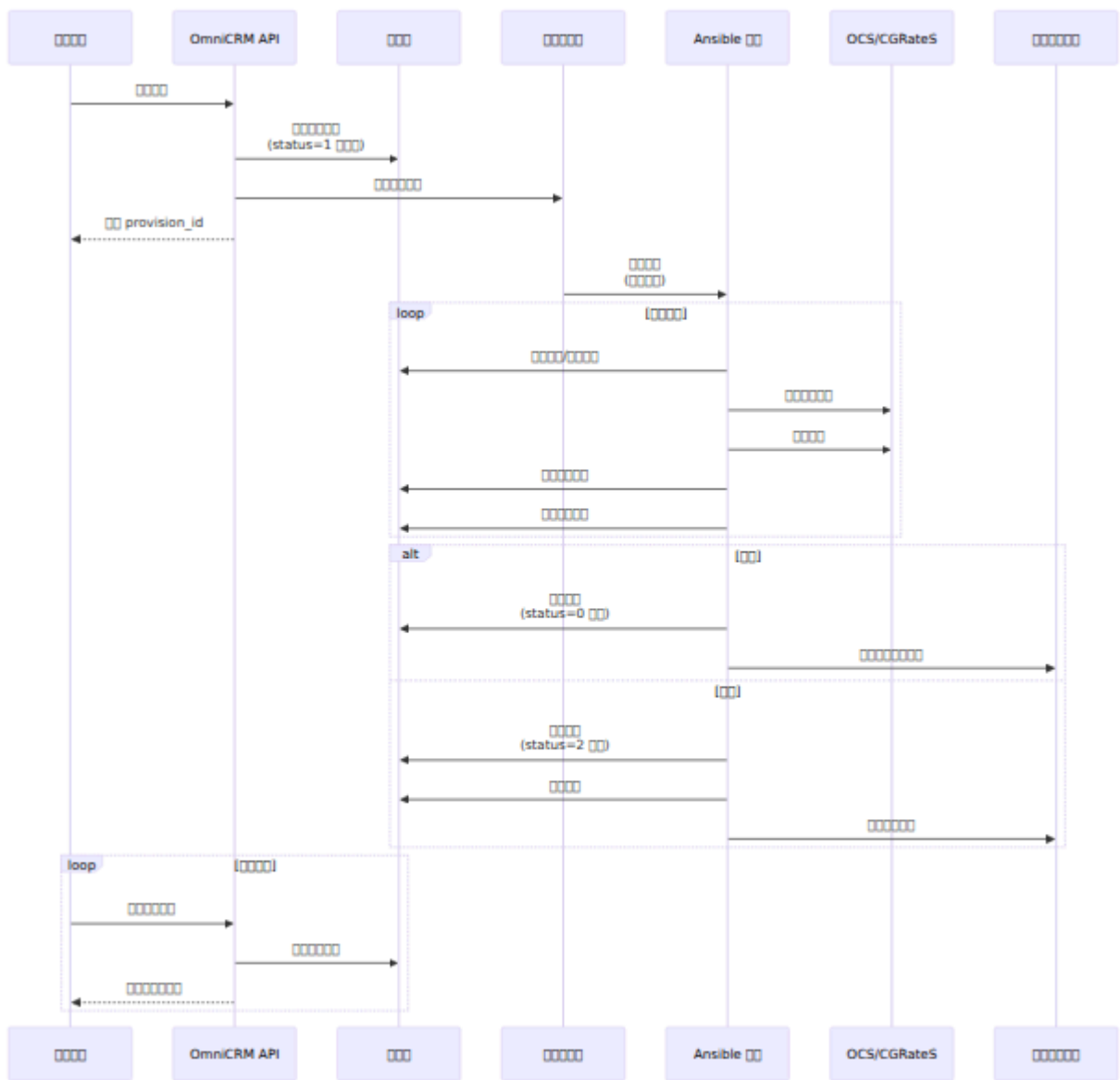


4.

OCS/CGRateS

- 
- 
- 
- 
- 

:



5. 测试

OmniCRM 测试用例

测试用例

测试 - 测试用例

欄名	データ型	制約
customer_id	整数 (PK)	一意
customer_name	文字列	一意
customer_account_type	文字列	'個人' または '法人'
customer_status	文字列	'有効', '無効', '凍結', '削除'
customer_payment_type	文字列	'現金' または 'クレジット'
customer_enabled	ブール値	一意
tax_identifier	文字列	VAT/GST 番号
contract_start_date	日付	一意
contract_end_date	日付	一意

欄名 - 一意制約

欄名	制約	データ型
contact_id	PK	INT
customer_id	FK	INT
contact_firstname		VARCHAR
contact_lastname		VARCHAR
contact_email		VARCHAR
contact_phone		VARCHAR
contact_types		TEXT ('', '', '')

住所 - 住所情報

欄名	制約	データ型
site_id	PK	INT
customer_id	FK	INT
site_name		VARCHAR
address_line_1		VARCHAR
city, state, zip_code		VARCHAR
latitude, longitude		GPS 座標
google_maps_place_id		Google Maps ID
plus_code		VARCHAR

住所情報

00 - 0000000

00	00	00
service_id	00 (PK)	000000
customer_id	00 (FK)	0000
product_id	00 (FK)	00000
site_id	00 (FK)	00000
service_name	000	00000
service_uuid	000	000000000
service_status	00	00000
service_billed	00	00000
wholesale_cost	00	000000
retail_cost	00	00000
bundled_parent	00 (FK)	000000000

00 - 0000000

欄名	データ型	制約
product_id	整数 (PK)	一意
product_name	文字列	一意
product_slug	文字列	URL 一意
category	整数	'1', '2', '3', '4'
provisioning_play	文字列	Ansible プレイブック
provisioning_json_vars	JSON	変数
inventory_items_list	文字列	一意
retail_cost	整数	一意
retail_setup_cost	整数	一意
enabled	ブール値	一意

一意

PK - 一意/一意

欄名	制約	データ型
transaction_id	PK (主キー)	整数
customer_id	FK (外キー)	整数
invoice_id	FK (外キー)	整数
service_id	FK (外キー)	整数
title		文字列
retail_cost		小数
tax_percentage		小数
tax_amount		小数
void		ブール値

欄名 - データ型

欄名	制約	説明
invoice_id	PK (PK)	請求ID
customer_id	FK (FK)	顧客ID
paid	ブール値	支払済フラグ
void	ブール値	キャンセルフラグ
payment_reference	文字列	Stripe 請求ID
start_date, end_date	日付	有効期間
due_date	日付	支払期限
retail_cost	数値	小売価格

備考

請求 - 請求明細

欄名	制約	説明
inventory_id	PK	在庫ID
customer_id	FK	顧客ID
service_id	FK	サービスID
inventory_template_id	FK	在庫テンプレートID
item		品名 SIM 品
item_state		'0', '000', '0000' 等
itemtext1-20		品名
management_url		管理 URL
config_content		設定内容

品名 - SIM 品

欄名	制約	説明
inventory_template_id	PK	在庫テンプレートID
item		品名
itemtext1_label		itemtext1 品名
itemtext2_label		itemtext2 品名

品名

品名 - SIM 品

欄名	制約	説明
provision_id	PK	主キー
product_id	FK	製品ID
customer_id	FK	顧客ID
service_id	FK	サービスID
provisioning_play		Ansible プレイブック
provisioning_status		0=成功, 1=失敗, 2=保留

プロビジョニング履歴

欄名	制約	説明
provision_event_id	PK	主キー
provision_id	FK	プロビジョニングID
event_name		イベント名
event_number		イベント番号
provisioning_status		0=成功, 1=失敗, 2=保留
provisioning_result_json	JSON	結果JSON

プロビジョニング結果

履歴 - 一覧

欄名	データ型	制約
id	整数 (PK)	一意
username	文字列	一意
email	文字列	一意
email_verified	ブール値	
is_2fa_enabled	ブール値	
totp_secret	文字列	TOTP 秘密鍵

ユーザ - ユーザ情報

欄名	データ型	制約
id	整数 (PK)	一意
name	文字列	一意
description	文字列	

ユーザ - ユーザ設定

欄名	データ型	制約
id	整数 (PK)	一意
name	文字列	一意
description	文字列	

ユーザ:

-
-
-

□□□

Stripe □□□□

-
- PCI
-
-
-

Mailjet □□□□□□

-
-
-
-

Google Maps

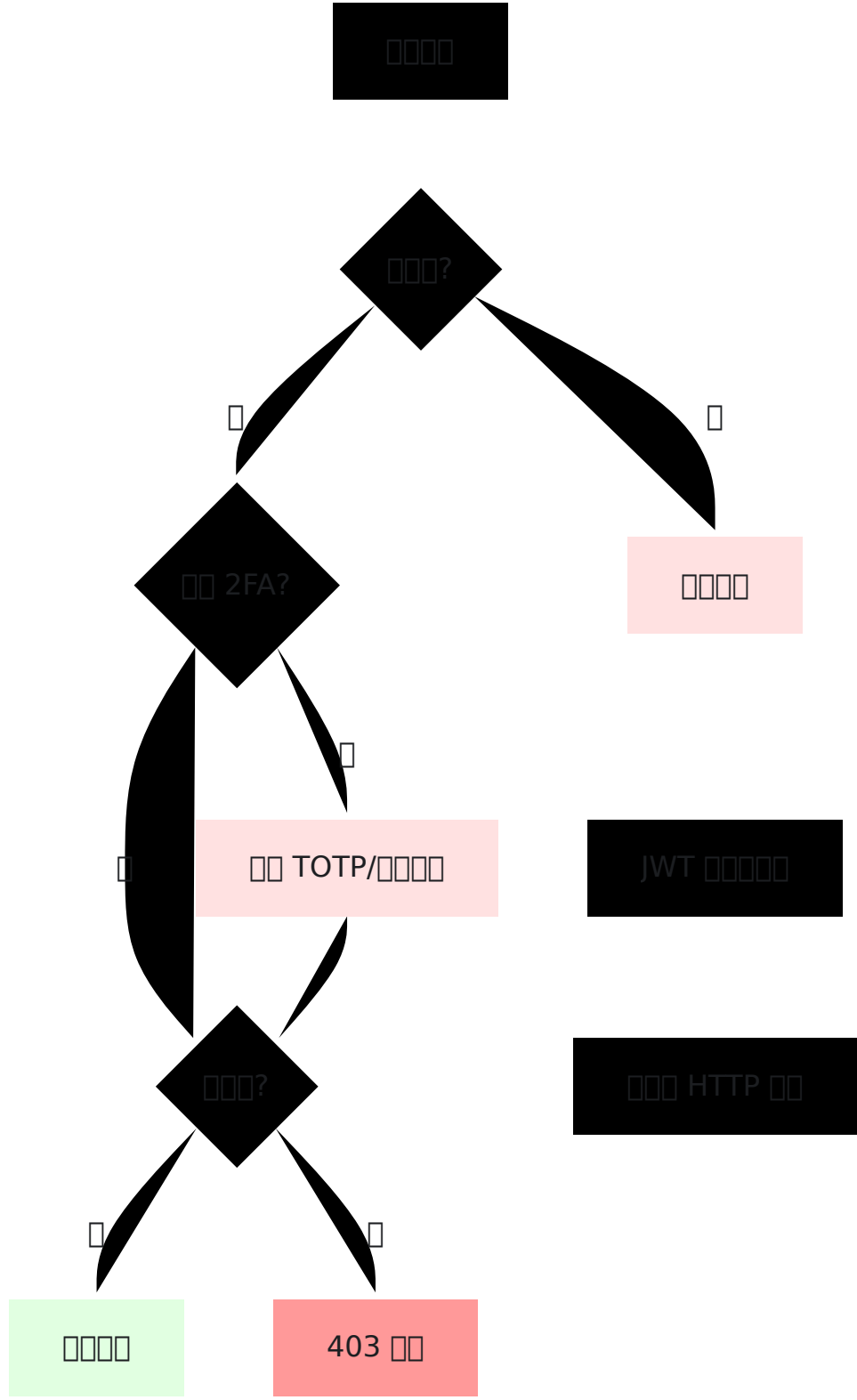
-
-
- Plus Code
-

OCS/CGRateS

-
-
-
- CDR

- □□□□□□□□

□□□□

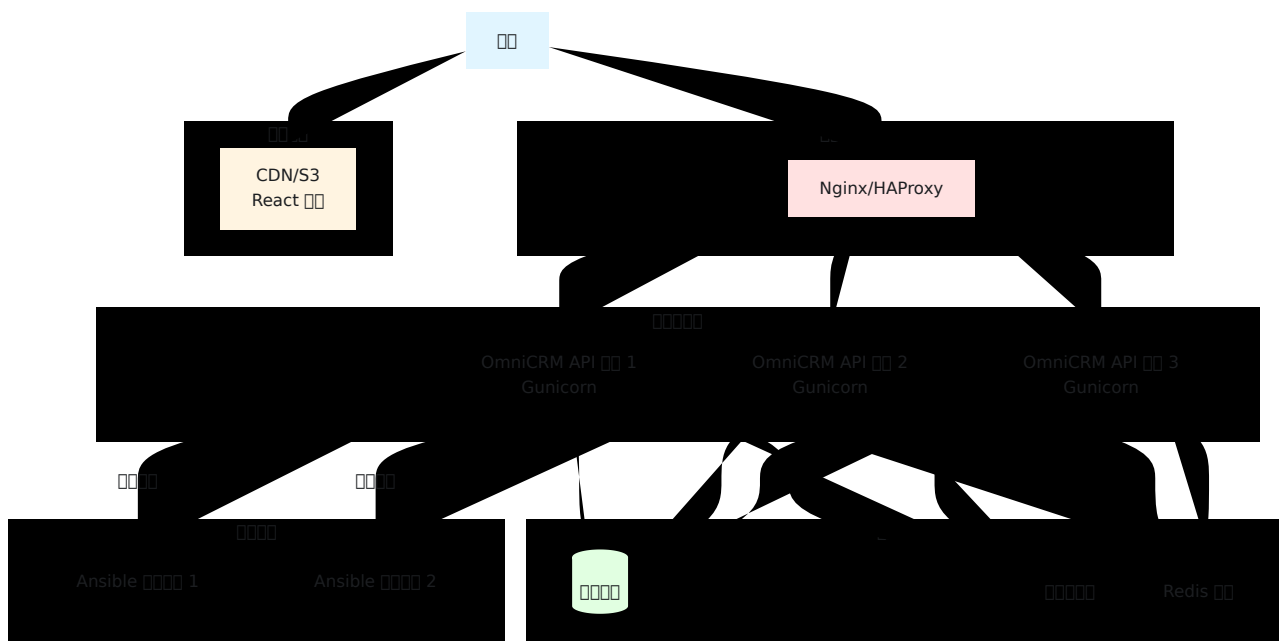


□□□□:

- JWT
- TOTP
- RBAC
-
- Redis
- bcrypt
-
-

架构

部署:



技术栈

栈:

- Python 3.x
- Flask
- SQLAlchemy ORM
- Alembic

- Ansible 配置

UI:

- React
- 路由
- React Router
- Axios 调用 API 接口

数据库:

- 数据库连接池
- Redis 缓存

第三方:

- CGRateS 计费/充值
- Stripe 支付
- Mailjet 邮件
- Google Maps 地图

部署

部署:

- 调用 API 接口
- 数据库
- Redis 缓存

部署:

- 数据库
- 路由
- 配置

部署:

- Python
- Ansible
- Docker
- Kubernetes

Architecture

OmniCRM uses Prometheus for monitoring and alerting.

Features:

- API
- Frontend/Backend
- Database
- Integrations (OCS, Stripe, Mailjet)
- Reporting

Stack:

- Python Flask
- Ansible
- Docker
- Kubernetes

Endpoint: `/crm/metrics` Prometheus monitoring and alerting.

Tools

- Ansible - Configuration
- Docker - Containerization
- Kubernetes - Orchestration
- API - RESTful
- RBAC - Access Control

OmniCRM

OmniCRM 2FA RBAC UI

Self-Care Portal <self_care_portal> RBAC <rbac>

OmniCRM

OmniCRM

- RBAC - RBAC
- RBAC (2FA) - RBAC TOTP
- RBAC - RBAC 30
- RBAC - RBAC
- RBAC - RBAC 2FA
- RBAC - RBAC Google Apple Facebook
- RBAC - RBAC

OmniCRM

OmniCRM

□□□□

□□□□

1. □□ □□□□□□□□□□□□□□□□
2. □□ □□
3. □~~◆◆◆~~□□□ “□□□ 30 □” □□□□□
4. □□ “□□”

□□□□□□□□

- □□ **2FA** □□□□□□□□□□□□□□
 - □□ → □□□□□□ (/self-care)
 - □□/□□□ → □□□□□□ (/customers)
 - **CBC** □□ → □□□□□□ (/create-cell-broadcast)
- □□ **2FA** □□□□ 2FA □□□□

□□□□□□

□□□□□□□□ **30** □□□□□□□□□□□□□□□□□□□□ HTTP-only cookies□

□□/□□□□□□

□□ □□□□ (□) □□□□□□□□□□

如何 2FA 如何

如何设置 2FA/如何设置 2FA 如何设置

如何设置

1. 如何设置 Google Authenticator/AUTHY 如何
2. 如何 OmniCRM 如何
3. 如何 6 如何
4. 如何 6 如何
5. 如何设置

如何设置

如何设置

1. 如何 “如何” 如何
2. 如何设置 3fa5b9c2 如何
3. 如何 “如何”
4. 如何设置

如何

如何 “如何” 如何

環境変数

REACT_APP_ALLOW_SOCIAL_LOGINS=yes

[Google] [• Apple] [Facebook]

“”

環境変数

“”

2FA

2FA TOTP

2FA

2FA 2FA

1

[] []

2FA

2

[] []

1. 設定方法
 - iOS → Apple App Store → “Google Authenticator”
 - Android → Google Play → “Google Authenticator”
 - 代替として Authy、Microsoft Authenticator、1Password
2. 設定手順
 - 設定画面を開く
 - “+” をタップ
 - “QRコード” を選択
 - QRコードを読み取る
 - “OmniCRM” を検索して 6 桁のコードを入力
3. 設定完了
 - 設定完了後 8 桁のコードが表示される
 - 設定が完了したことを確認する
 - 設定が完了したことを確認する
 - 設定が完了したことを確認する
 - “設定” をタップして設定を確認する
4. 設定完了
 - 設定完了後 6 桁のコードが表示される
 - “設定” をタップ
 - 設定完了後 2FA を有効にする

3 2FA 設定

設定方法

設定完了後 2FA 設定

Enable Two-Factor Authentication

Step 1: Scan this QR

Step 2: Save these backup codes

d1147a5b	987df91
bee1fba9	ee91cbf5
185ac5d6	2fbf3731
e5f5d9f3	178bf54b
ce733cf7	a7fe5529

Copy Codes

Step 3: Confirm by entering your first TOTP code

Cancel Confirm

Change Password

Current Password:

New Password:

Confirm New Password:

Two Factor:

Enable two factor authentication for added security.

2025 © Omnitouch.

□ □ □ □ □ □ □ □ □ □ □ □

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

□ **1** □ □ □ □ □ □ □ □

□ □ □ □ □ □ □ □ “□ □ □ □ □ □”

1. 00 0000000
2. 00 “0000000”

0000000

- 000000000000000000000000
- 00000000 Mailjet 00000000???
- 000000000000000000000000 1 000000
- 00000000“0000000000000000”

000000000000

000000“0000000000000000”

0 2 0000000000

0000000000000000

0 [00]0

OmniCRM

<<https://yourcompany.com/reset-password/abc123token456>>

1

3

[]

- 1.
- 2.
3. “ ”

- 8
-

- “ ”
-
-

/

XXXXXXXXXXXXXXXXXX

[XXXXXXXXXX]

XXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXX ❗❗XXXXXXXXXXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

XX XXXX XX XX John Smith <john@example.com> +44 123... ➡ XXX Jane
Doe <jane@example.com> +44 456... ➡ XX Bob Wilson
<bob@example.com> +44 789... ➡ XXX ☒

XXXXXX

- ➡ XX - XXXXXXXXXXXXXXXXXXXX
- XX XX - XXXXXXX
- XX XXXX - XXXXXXX
- XX XX **2FA** - XXXXX 2FAXXXXXX 2FA XXXX
- ☒ XXXXXXX - XXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

XX **1** XXXXXXXXXXXXXXX (XX)

XXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

XXX John Smith (<john@example.com>)

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[] []

2

“ ”

3

John Smith

[]

• • • •

[]

-
-
-
-

- 1.
2. “ ”
- 3.
- 4.

2FA

1 2FA ()

2FA

- 設定
2. 設定
 3. 設定 2FA

設定

設定

設定

設定 (✉) 設定 `login_count = 0`

設定 (✉)

設定

設定 Bob Wilson (<bob@example.com>)

設定 • 設定 • 設定 • 設定 • 設定

[設定] [設定]

設定 “設定”

設定

設定 **Mailjet** 設定

設定 `api_crmCommunicationUserWelcome`

設定

設定 (⇒) 設定

設定

設定

設定

□□□□

□□ admin customer_service_agent_1 customer

[□□] [□□□□]

□□□□□□

- □□□□□□□□□□
- □□ - □□/□□□□□□□□□□
- □□/🔒🔒🔒□□□□

□□□□□□□□

□□ □□□□ (□) □□□□□

□□□□□□□□

□□□ John Smith (<john@example.com>)

⚠ □□□□□□□□□□

□□□□□□□ • □□□□□□□ • 2FA □□ • □□□□

□□□□□□□□□□□□□□

[□□] [□□□□]

□□ “□□□□” □□□□

□□□□□

□□□□□

□□□□□□□

□□□□□

- □□□□□□□□□□

- 2FA 2FA 2FA
- 2FA 2FA 2FA
- 2FA 2FA 2FA

2FA 2FA

- 2FA 2FA 2FA
- 2FA 2FA 2FA
- 2FA 2FA 2FA
- 2FA 2FA 2FA
- 2FA 2FA 2FA

2FA 2FA

- 2FA 2FA 2FA
- 2FA 2FA 2FA
- 2FA 2FA 2FA
- 2FA 2FA 2FA

2FA 2FA

2FA 2FA

- 2FA 2FA 2FA
- 2FA 2FA 2FA
- 2FA 2FA 2FA
- 2FA 2FA 2FA
- 2FA 2FA 2FA

2FA 2FA

- 2FA 2FA 2FA
- 2FA 2FA 2FA
- 2FA 2FA 2FA
- 2FA 2FA 2FA

2FA 2FA

- 2FA 認證
- 認證碼
- 認證碼
- 認證碼 2FA
- 認證碼 2FA

認證碼

- 認證碼
- 認證碼
- 認證碼
- 認證碼
- 認證碼

認證碼

“認證碼”

- 認證碼
- 認證碼
 - 認證碼
 - 認證碼
 - 認證碼
 - 認證碼

2FA 認證碼

- 認證碼
- 認證碼
 - 認證碼 → 認證碼 → 認證碼
 - 認證碼 30 秒
 - 認證碼
 - 認證碼
 - 認證碼 2FA

“”

- Cookies
- - Cookies
 - Cookies
 - OmniCRM
 -

-
- - /
 -
 - 5-10
 - Mailjet
 -

- 1
- -
 -
 -

2FA

-
- -
 -
 -

##

- 密碼 密碼/密碼密碼密碼
- 密碼密碼密碼
 - 密碼密碼
 - 密碼密碼密碼 2FA
 - 密碼密碼密碼
 - 密碼 2FA密碼密碼密碼

密碼“密碼 2FA”

- 密碼 密碼
- 密碼
 - 密碼密碼密碼
 - 密碼 API 密碼
 - 密碼密碼

密碼密碼密碼

- 密碼 API 密碼密碼
- 密碼
 - 密碼密碼
 - 密碼密碼
 - 密碼 API 密碼密碼
 - 密碼密碼

密碼密碼

密碼密碼

- 密碼密碼密碼
- “密碼”密碼密碼 30 秒
- 密碼密碼 HTTP-only cookies密碼密碼 JavaScript 密碼
- 密碼密碼 cookies 密碼 HTTPS 密碼

密碼密碼

- 密碼密碼密碼密碼密碼

- `auth`
- `auth/2fa`
- `auth/2fa/enable`

2FA `auth`

- `TOTP` `auth`
- `auth`
- `auth`
- `auth`

`auth`

- `2FA` `auth`
- `auth`
- `auth` `auth`
- `auth` IP `auth`

`auth`

- `2fa` - `auth` 2FA API `auth` API `auth`
- `rbac` - `auth`
- `administration_configuration` - Mailjet `auth`
- `integrations_mailjet` - `auth`
- `customer_care` - `auth`



0fT52ZvoZBE

XXXXXXXXXX

XX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/XXXX

XX
XX

XX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX

XX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXX <csa_activity_log> ❖❖ XXXXXXXXXXX <administration_tags>
<administration_attributes> XXXXXXXXXXX
<administration_inventory> XXXXXXXXXXX <payments_transaction>
<payments_invoices> XXXXXXX <basics_payment>

XXXXXXXXXXXXXXXX <csa_add_service> XXXXXXXXXXXXXXX

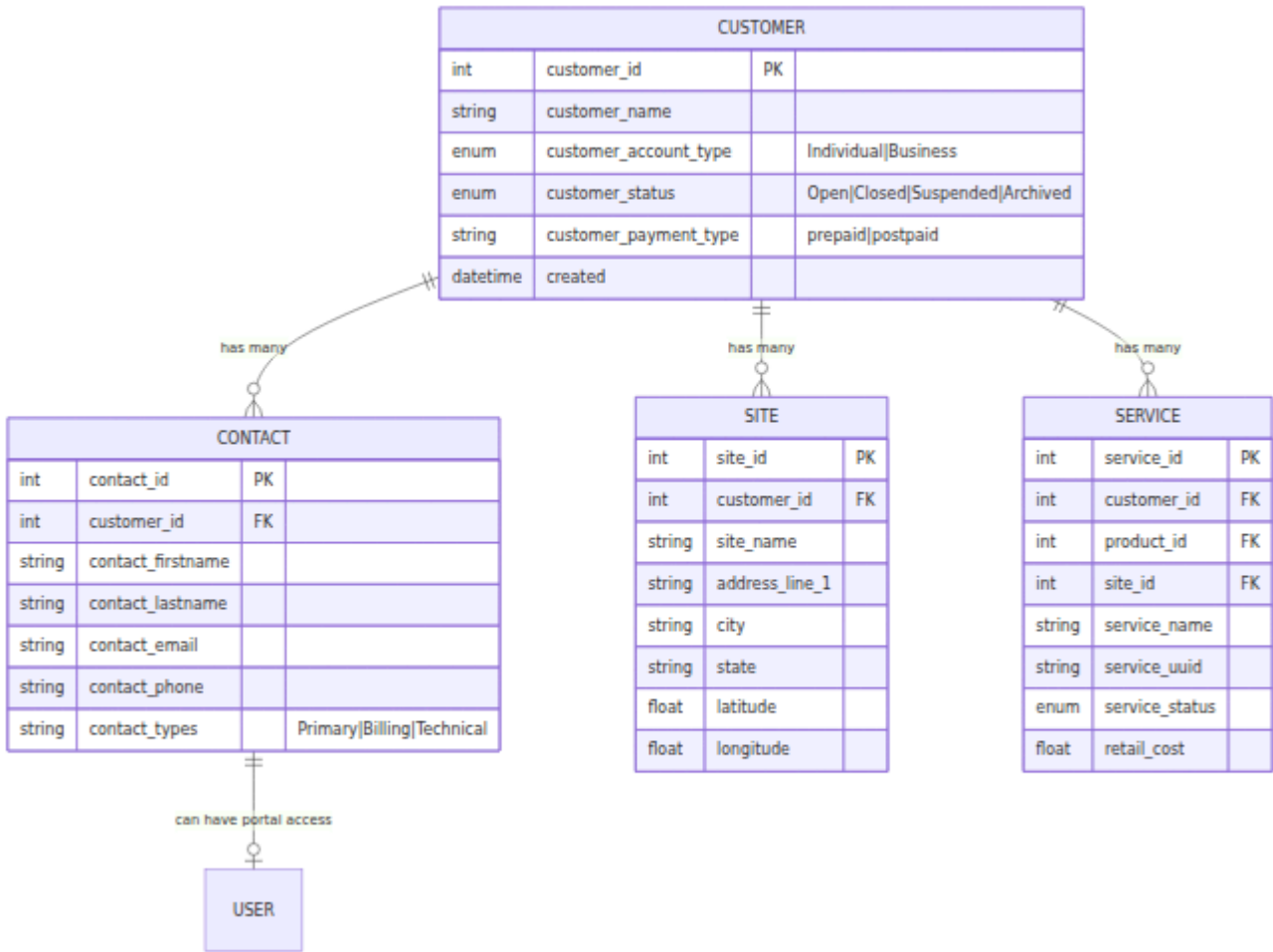
XXXXXXXXXXXXXXXXXXXX

XXXXXX

OmniCRM XXX

XXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

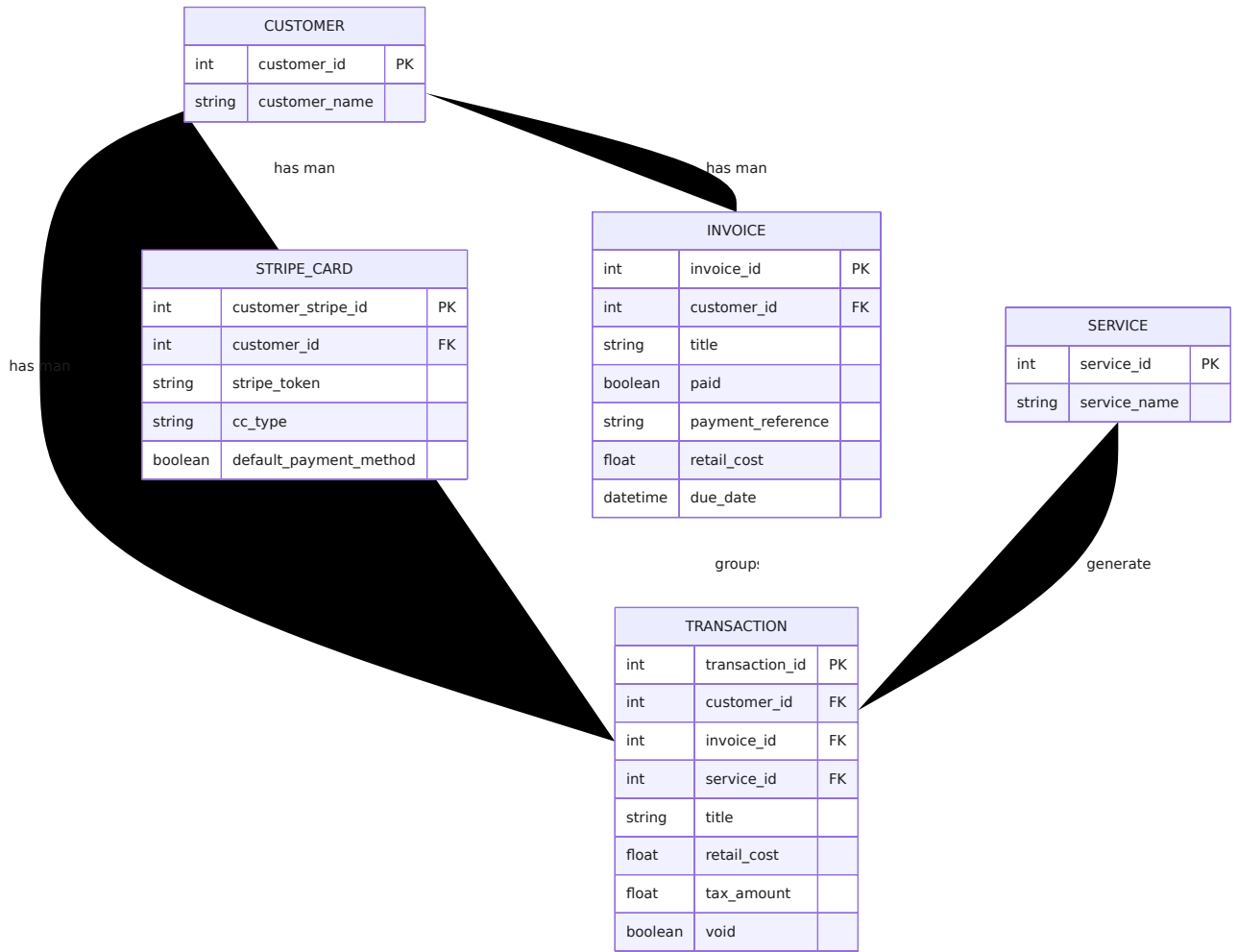


□□□□

- □□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□
- □□□□□□□□
- □□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□

□□□□□□□□□□□□□□

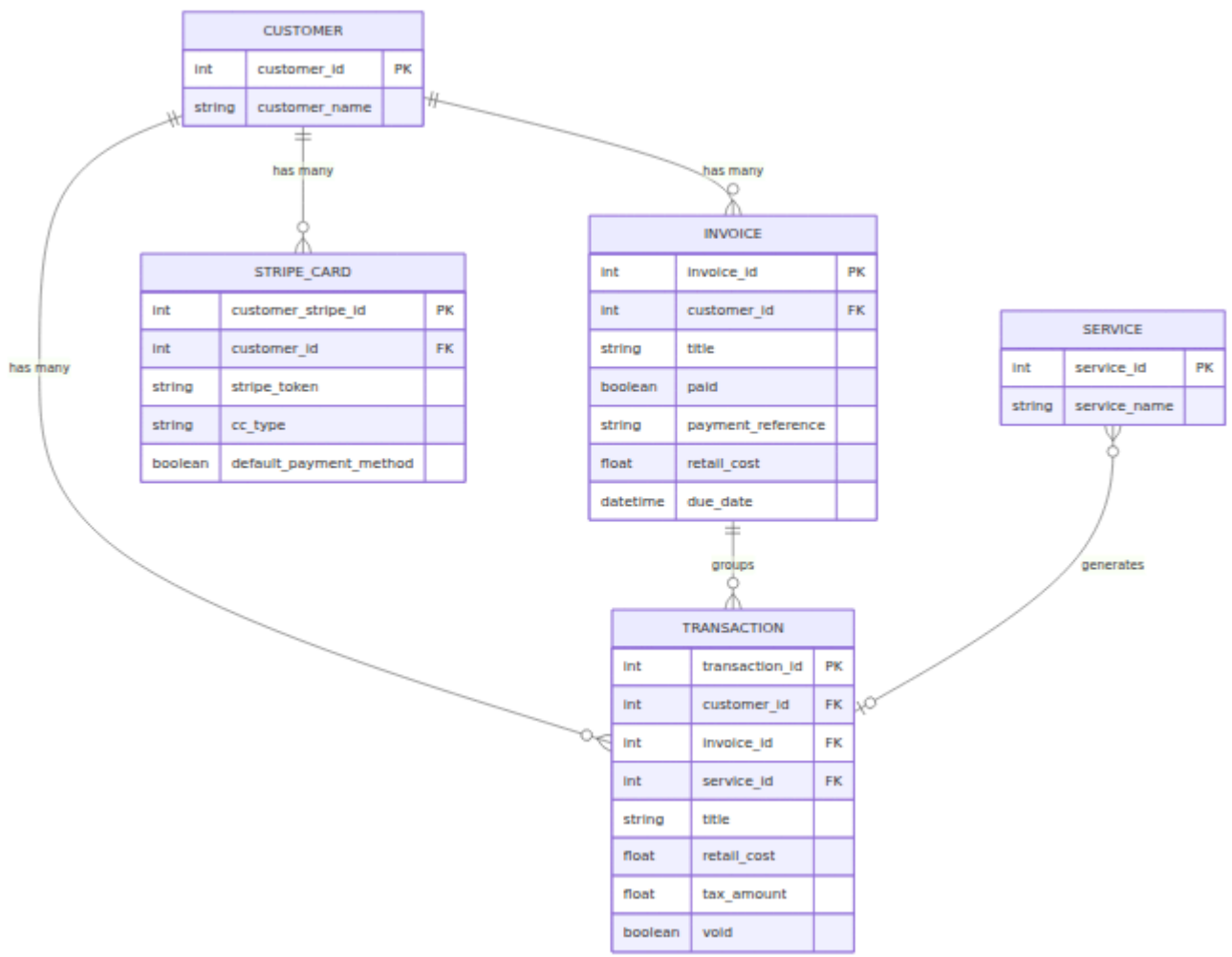


□□□□

- □□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□
- Stripe□□□□□□□□□□
- □□□□□□□□□□□□□□

□□□□□□

□□□□□□□□□□□□□□□□□□



□□□□

- □□□□□□□□□□□□□□□□
- □□□□Ansible□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□□□□□□□□

INVENTORY_TEMPLATE			
int	inventory_template_id	PK	
string	item		SIM Card Router Phone Number
string	itemtext1_label		Serial Number
string	itemtext2_label		MAC Address

CUSTOMER		
int	customer_id	PK
string	customer_name	

SERVICE		
int	service_id	PK
string	service_name	

defines structure

owns

uses

INVENTORY			
int	inventory_id	PK	
int	customer_id	FK	
int	service_id	FK	
int	inventory_template_id	FK	
string	item		
enum	item_state		New Assigned Used Damaged
string	itemtext1		
string	itemtext2		
string	management_url		
text	config_content		

□□□□

- □□□□□□□□□□□□□□□□□□□□
- □□□itemtext1-20□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□

□□□□□□

□□□□□□□□□□□□□□□□□□

USER		
int	id	PK
string	username	
string	email	
boolean	email_verified	
boolean	is_2fa_enabled	
string	totp_secret	

assigns

linked to

ROLE		
int	id	PK
string	name	Admin CSR Customer
string	description	

CONTACT		
int	contact_id	PK
string	contact_email	

grants

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

□□□□

- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□2FA□□□□□□
- □□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□

□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□

- 2 - /
- -
- / -
- -
- -
- -
- **Plus** - 11 8C3MFJV8+2F
- **Google Maps** ID -

2.

-
-
-
-

Plus

Plus / Plus 11

✓ "123 Main Street" ✓ 1 "123 Main Street" ✓
 "London" ✓ "Greater London" ✓ "United Kingdom" ✓ "SW1A
 1AA" ✓ 51.5074 ✓ -0.1278 ✓ Plus "9C3XGPHC+3Q" ✓ Google
 ID "ChIjdd4hrwug2EcRmSrV3Vo6III"

-
- REACT_APP_DEFAULT_COUNTRY
- **Plus** 12 11 + 1

□

□□□□□□□□

□□□□□□□□□□□□□□□□Plus□□□□OmniCRM□□□□

1. □□□□□□

- □□□□□□ - □□Plus□□□□□□□□□□□□□□
- □□□□ - □□□□□□□□□□□□□□
- □□□□ - □□□□□□□□□□□□□□□□□□

2. □□□□

- □□□□□□ - □□□□□□□□□□□□□□□□/□□□□□□□□
- □□□□ - □□□□□□□□□□□□□□□□□□□□ Mailjet <integrations_mailjet>
- □□□□ - □□□□□□□□□□□□□□

□□□

```

□□□SELECT * FROM Customer_Site
WHERE distance(latitude, longitude, 51.5074, -0.1278) < 5

□□□47□□□□□□□□ □□□□47□□□□□□□□□□□□

```

3. □□□□□□

- □□□□ - □□□□□□□□□□□□□□
- □□□□□□ - □□□□□□□□
- □□□□ - □□□□□□□□□□

4. □□□□□□□□

- □□□□ - □□□□□□□□□□
- □□□□ - □□□□□□□□□□□□
- □□□□ - □□□□□□□□□□□□□□□□□□



OmniCRM □□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□

WebApp Web

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□

□□□□

□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

□□□

□□□

□□□□□□□□ □□ <payments_invoices> □□□□□□□□□□

□□□□

□□□□□□□□□□□□

□□□□□□

- 1. □□□ □□ → **【□□□□】**
- 2. □□ □□ □□
- 3. □□□□□□□□□□□□

□□□□□□

1. □□□ □□ → □□□□□□□□
2. □□□□□□□□□□

□□□□□□□□

□□□□□□□□□□◆◆◆□□□□□□□□

□□□□□□

- □□□□ - □□□□□□□□□□□□□□□□
- □□□□□□ - □□□□□□□□□□□□
- □□□□□□ - □□□□□□□□□□□
- □□□□□□ - □□□□□□□□□□□

□□□□□□

- □□1,000□□□□□□□□“k”□□□□□□\$1.5k□
- □□1,000,000□□□□□□□□“M”□□□□□□\$2.3M□
- □□1,000,000,000□□□□□□□□“B”□□□□□□\$1.1B□

□□□□

□□□□□□□□□□□□□□□□

□□□□

- **ID** - □□□□ID
- □□ - □□□□□□
- □□ - □□□□□□
- □□ - □□□□□□□□
- □□ - □□□□□□□□□□□□□□□□
- □□ - □□□□□□□□□□□□ID□□□□□□□□
- □□ - □□□□□□□□□□□□□□□□□□□□□□

□□□□□

□□□□□□□□□□□□ : □□□□□□□□

- □□□□ - □□□□□□□□□□
- □□□□**PDF** - □□PDF□□□□□□□□□□
- □□□□ - □□□□□□□□□□□□□□□□

□□□□

□□□□□□□□

□□□□□□□□□□

□□□□□□□□□□□□

- □□□□□□ - □□□□□□□□□□□□
- □□□□□□ - □□□□□□□□
- □□□□ - □□□□□□□□□□□□
- □□□□ - □□□□□□□□□□□□SIM□□□□□□
- □□□□□□ - □□□□□□□□

- 0000 - 0000000000

00000000/0000

00000000000000

- 0000 - 00000000
- 000 - 0000000/000000
- 0000000 - 000000000000
- 0000 - 00000000
- 00 - 00000000
- 00 - 0000000000

00000000

00 "+ 0000" 000000000000

0000000000

00000000/0000

□□□□

- □□□□ - □□□□□□□□□□□□□□□□□□□□
- □□□□ - □□□□□□□□□□□□□□□□□□□□
- □□ - □□□□□□□□□□□□
- □□ - □□□□□□□□
- □□□□ - □□□□□□□□□□□□□□
- □□□□ - □□□□□□□□□□□□□□
- □□ - □□□□□□□□□□□□□□□□□□□□0%□
- □□ - □□□□□□□□□□□□□□
- □□ - □□□□□□□□□□□□□□
- □□□□ - □□□□□□□□□□□□

□□□

- □□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□

□□□□□

1. □□□□□□□□□□
2. □□□□□□□□□□

3. 000“00000”000
4. 000000000
5. 000000000

00000000

00

000000000000000000

- 00ID
- 00
- 00
- 00ID

000

00000000000000

000000

- 0000 - 0000000000
- 0000 - 0000000000

00000

- 00000 - 00000000000
- 00000 - 000000000000000

00

000000000000

- **ID** - 000ID00000/000
- 00 - 0000000
- 00 - 0000000

- 00 - 000000000000/0000
- 00 - 0000!D00

0000000000000000 ↔ 0000

0000

0000000000 000000000000

000

- 00000000
- 00000000000000000000

00000

1. 00000000
2. 00000000 : 0
3. 00 "0000"
4. 00000000

000000

- 00000 void = true
- 00000000000000

- 零售商店
- 零售商店“折扣”
- “零售商店”

零售商店“折扣”零售商店“零售商店”

零售

零售商店零售商店零售商店零售商店零售商店零售商店零售商店

零售

- 零售商店 - 零售商店零售商店
 - 零售 - 零售商店零售商店零售商店零售商店
 - 零售 - 零售商店零售商店零售商店
 - 零售 - 零售商店 $retail_cost \times (tax_percentage / 100)$
 - 零售 - 零售商店零售商店 $\$10.00 (10\%)$
- 零售商店/零售 - 零售商店
 - 零售商店零售商店
 - 零售商店零售商店零售商店 0%
 - 零售商店零售商店零售商店零售商店零售商店

零售

- 10% of \$50.00
- $\$50.00 \times 0.10 = \5.00
- \$5.00 (10%)

NIL/

- 0
- 0%
- "-"

-
-
-
- " " "

- 00000000-0

000000

- 000000
- 0000000000000000
- 00ID000000000000
- 0000000000✓0
- 0000

00000

000000000000

1. 0000000000000000
2. 000000000000
3. 000000000000
4. 00 invoice_id 00000
5. 00000000"0000"

000 payments_invoices 0000000000000000

00000000

00000**1**0000000000000000

1. 000000"00000020"
2. 00000000£10000
3. 000000 00 00
4. 00 "+ 00000"
5. 00 00 0000
6. 00 0000 0000
7. 000000"0000000"
8. 000000"0108-90000000"
9. 0000000010.00

10. 0000000000000000
11. 00 "0000"
12. 000-£10.00000000
13. 0000000000000000

0000**2**0000000000

1. 000000000000
2. 000000£75000000
3. 000000 00 00
4. 00 "+ 0000"
5. 00 00 0000
6. 000000"0000"
7. 000000"00000000000000"
8. 0000000075.00
9. 0000000045.00000000000000
10. 0000000000
11. 00000000
12. 00 "0000❖❖❖"
13. 00000000000000
14. 000000000000

0000**3**0000000000

1. 000000000000
2. 000000000000
3. 000000000000000 : 0
4. 00 "0000"
5. 0000000000
6. 000000000000
7. 00000000000000

□□□□

- `payments_invoices` - □□□□□□
- `payments_process` - □□□□□□
- `basics_payment` - □□□□□
- `csa_activity_log` - □□□□□□□□□□

OmniCRM API

OmniCRM API HTML templates using Jinja2

Invoice Templates

OmniCRM API invoice templates/

Files

- norfone_invoice_template.html - Norfone
- cifi_invoice_template.html - CIFI

Config

OmniCRM API crm_config.yaml

```
invoice:  
  template_filename: 'norfone_invoice_template.html'
```

Invoice Data

Jinja2 template

Variables

- {{ invoice_number }} - Invoice ID INV-2025-001234
- {{ date }} - ISO 2025-01-10T12:00:00
- {{ due_date }} - Due date 2025-02-10
- {{ start_date }} - Start date
- {{ end_date }} - End date
- {{ total_amount }} - Total amount
- {{ total_tax }} - Total tax

Context

- {{ client.name }} - Client name

- `{{ client.address.address_line_1 }}` - 0001
- `{{ client.address.address_line_2 }}` - 0002
- `{{ client.address.city }}` - 00
- `{{ client.address.state }}` - 0/0
- `{{ client.address.zip_code }}` - 0000
- `{{ client.address.country }}` - 00

000000

0000000000000000

```
{% 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 %}
```

00000

- `sub_transaction.transaction_id` - 00ID
- `sub_transaction.created` - 0000/00
- `sub_transaction.title` - 0000
- `sub_transaction.description` - 0000
- `sub_transaction.retail_cost` - 00000
- `sub_transaction.tax_percentage` - 0000000000100010%0
- `sub_transaction.tax_amount` - 000000000

000000000

```

<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>

```

□□□□□□□□□□

□□**1**□□□□□□□□

```

cd OmniCRM-API/invoice_templates/
cp norfone_invoice_template.html
your_company_invoice_template.html

```

□□**2**□□□□**HTML/CSS**

□□ `your_company_invoice_template.html` □□□□□□□□

□□□□□□□□

1. □□□□□□□□

```

<!-- □□□□□URL□□ -->
![[□□□□]](https://yourcompany.com/logo.png)

<!-- □□□□□□□□ -->
<h1>□□□□□□□□</h1>

```

2. □□□□

```

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

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

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

```

3. Footer

```

<footer>
  <p>Your company name</p>
  <p>123 Street, Sydney, NSW 1234</p>
  <p>Phone: +1-555-123-4567 | Email: billing@yourcompany.com</p>
  <p>ABN/12345678900</p>
</footer>

```

4. Links

```

<div class="payment-info">
  <h3>Payment</h3>
  <p><strong>Pay to:</strong> https://yourcompany.com/pay</p>
</div>
<p><strong>Account Details</strong></p>
<ul>
  <li>Account Number</li>
  <li>BSB 123-456</li>
  <li>Account ID 987654321</li>
  <li>Invoice Number {{ invoice_number }}</li>
</ul>
</div>

```

5. Terms

```

<div class="terms">
  <h4>Terms</h4>
  <p>Payment is due within 30 days of invoice.</p>
  <p>Interest rate is 2% per annum.</p>
  <p>Contact billing@yourcompany.com for more information.</p>
</div>

```

3. Configuration

cd OmniCRM-API/crm_config.yaml

```

invoice:
  template_filename: 'your_company_invoice_template.html'

```

4. API

```

cd OmniCRM-API
sudo systemctl restart omnicrm-api

```

5. Deployment

1. Create a new release
2. Deploy the release

3. PDF

4.

Jinja2

```

{% if total_amount > 1000 %}
  <div class="high-value-notice">
    <p><strong></strong> - </p>
  </div>
{% endif %}

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

```

```

invoice_template_en.html
invoice_template_es.html
invoice_template_fr.html

```

```

<!-- 结账小票 -->
<tr>
  <td colspan="4" class="text-right"><strong>合计</strong></td>
  <td>${{ "%.2f"|format(total_amount) }}</td>
</tr>
<tr>
  <td colspan="4" class="text-right"><strong>税金</strong></td>
  <td>${{ "%.2f"|format(total_tax) }}</td>
</tr>
<tr>
  <td colspan="4" class="text-right"><strong>合计</strong></td>
  <td>${{ "%.2f"|format(total_amount + total_tax) }}</td>
</tr>

```

total_tax 结账小票 tax_amount 结账小票
 tax_percentage 结账小票 tax_percentage 0%结账小票

结账小票

结账小票

```

<div class="qr-payment">
  ![[二维码]](https://api.qrserver.com/v1/create-qr-code/?
  size=150x150&data={{ payment_url }})
  <p>结账小票</p>
</div>

```

PDF 打印

OmniCRM 使用 **WeasyPrint** 将 HTML 转换为 PDF

使用 CSS

- 使用 CSS 2.1
- 使用 CSS3 的 flexbox
- 使用 @font-face 嵌入字体

打印

- JavaScript
- CSS Grid
-
- CSS

```
@page {  
  size: A4;  
  margin: 1cm;  
}  
  
body {  
  font-family: Arial, sans-serif;  
  font-size: 10pt;  
}
```

```
@media print {  
  .no-print {  
    display: none;  
  }  
  
  .page-break {  
    page-break-after: always;  
  }  
}
```

```

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

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

```

□□□□□

□□□□□□□◀◀□□□□□□□□□□□□

```

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

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

```

□□□□□□

□□□□□

1. □□□□□
 - □□□□□□
 - □□□□□□□□
 - □□□□□□□□□□
 - □□□□□□
 - □□□□□□
2. □□□□□□
 - □□□□□□□□
 - □□□□□□□□

- 文字列
- 数値

3. PDF

- 最大サイズ<5MB
- 文字
- 画像
- 埋め込み

4. 文字列

- 文字列
- 文字
- 埋め込み

5. 埋め込み

- PDF
- Mailjet 15MB
- Gmail Outlook Apple Mail

埋め込み

API

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

埋め込み

埋め込み

- 埋め込み
- 埋め込み

PDF

- 埋め込みCSS
- CSS 2.1 WeasyPrint CSS

URL

- URL
- HTTPS URL

Table

-
- `table-layout: fixed`

Font

-
- Arial, Times New Roman



PDF

- HTML WeasyPrint
- HTML WeasyPrint

PDF

PDF OmniCRM PDF PDF

PDF

1. - PDF
 - PDF
 - PDF Base64
 - PDF SHA256
 - `Invoice_PDF_Cache`  
 - ID
 - PDF Base64
 -
 -
 -

2. 查詢 - 查詢所有PDF
 - 查詢invoice_id查詢PDF
 - 查詢所有PDF
 - 查詢 last_accessed 查詢所有PDF
3. 查詢 - 查詢所有PDF
 - 查詢所有PDF/查詢所有PDF
 - 查詢所有PDF
 - 查詢所有PDF

查詢

- 查詢 - 查詢所有PDF
- 查詢 - 查詢所有PDF
- 查詢 - 查詢PDF CPU
- 查詢 - 查詢所有PDF

查詢

查詢PDF

- 查詢所有PDF 90
- 查詢所有PDF 30
- 查詢所有PDF

API

API UI

- 查詢所有PDF
- 查詢所有PDF
- 查詢所有PDF

查詢 查詢所有PDF

```
-- 查詢所有PDF MySQL
DELETE FROM Invoice_PDF_Cache;
```

crm_config.yaml

crm

crm

crm

1. crm → [crm]
2. crm
3. crm

crm

1. crm → crm
2. crm

crm

crm

crm

- crm - crm

- 000000 - 00000000000000000000
- 00000 - 0000000000000000
- 00000 - 0000000000000000

000000

- 001,0000000000“k”0000000\$1.5k
- 001,000,0000000000“M”0000000\$2.3M
- 001,000,000,0000000000“B”0000000\$1.1B

0000000

- “00”“00”000000000000
- 000000000000000000
- 000000000000000000
- 000000000000

00000

00000000000000000000

- **ID** - ID
- - /
- - "N/A"
- -
- -
- -
- -
- -

□□□□

- ↓ (□□) - PDF
- □ (□□) -
- □ (□□) -
- ☒ (□□□□) -
- □ (□□) - Stripe Stripe

□□□□

□□ "+ □□□□" □□□□

□□□□

- 𠆩𠆩 - 𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩
- 𠆩𠆩 - 𠆩𠆩𠆩𠆩/𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩“𠆩𠆩 [𠆩𠆩]”𠆩
- 𠆩𠆩𠆩𠆩 - 𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩14𠆩𠆩𠆩
- 𠆩𠆩𠆩𠆩 - 𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩
- 𠆩𠆩𠆩 - 𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩
- 𠆩𠆩𠆩𠆩 - 𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩/𠆩𠆩𠆩𠆩𠆩

𠆩𠆩𠆩𠆩

- ✓ (𠆩𠆩𠆩𠆩) - 𠆩𠆩𠆩𠆩𠆩𠆩𠆩
- × (𠆩𠆩X) - 𠆩𠆩𠆩𠆩𠆩𠆩𠆩
- 𠆩𠆩 - 𠆩𠆩𠆩𠆩𠆩𠆩
- 𠆩𠆩𠆩𠆩 - 𠆩𠆩𠆩𠆩
- 𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩
- 𠆩𠆩𠆩𠆩𠆩/𠆩𠆩𠆩𠆩𠆩

𠆩𠆩𠆩𠆩

1. 𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩
2. 𠆩𠆩𠆩𠆩𠆩𠆩/𠆩𠆩𠆩𠆩
3. 𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩
4. 𠆩𠆩𠆩𠆩𠆩𠆩𠆩
5. 𠆩𠆩𠆩PDF𠆩𠆩
6. 𠆩𠆩𠆩𠆩𠆩𠆩 invoice_id 𠆩𠆩𠆩
7. 𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩
8. 𠆩𠆩𠆩𠆩𠆩“𠆩𠆩”𠆩

𠆩𠆩𠆩

𠆩𠆩𠆩 𠆩𠆩 𠆩𠆩 𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩

𠆩𠆩𠆩𠆩 𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩

𠆩𠆩𠆩𠆩 𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩𠆩

XXXXXXXXXX

XX

XXXXXXXXXX

- XXXXX - XXXXXXIDXXXXXXXXXXXXXXXXXXXX
- XXXXX - XXXXXXXXXXXXXXXXXXXXXXX
 - XXXXX
 - XXXXXX
 - XXXXX
 - XXXXXXXXXXXXX \$10.00 (10%)
 - XXXXXXXXXXXXX“-”
- XXXXX - XXXXXXXXXXX
 - XXXXX
 - XXXXXXXXXXXXXXXXXXX
 - XXXXXXXXXXXXXXX
 - XXXXXXXXXXX + XX
- XXXXX - XXXXXXXXXXXXXXX
 - XX **PDF** - XXXXXPDFXXXXXXXX
 - XXXXXXX - XXXXXXXXXXXXXXXXXXXXXXXXXXX
 - XXXXX - XXXXXXXXXXXXXXXXXXXXXXX
 - XX - XX StripeXXXXXXXXXXXXXXXX StripeXXXX

- 00 - 00000000000000000000

0000PDF

00000 00000↓ 000000000000 "00PDF" 00000000PDF

00000

1. 0000000000
2. 00000000000000000000
3. 0000000000 Invoice_01234.pdf
4. PDF000000000000

PDF00000

- 0000 - 00000PDF0000000000002-300
- 0000 - 00000000
- 0000 - 00000000000000000000

0000000000

- 0000000000 - 0000000000API000000
- **PDF**00000 - 0000000000000000
- 0000 - 0000000000000000000000

00000

00 00000000 00000000

📌

1. 📌
2. 📌📌
3. 📌
 - **Stripe**📌 - 📌
 - 📌 - 📌
 - 📌 - 📌
 - **POS**📌 - 📌
 - 📌 - 📌
4. 📌Stripe📌
 - 📌
 - 📌
 - 📌
5. 📌
 - 📌
6. 📌 "📌" 📌
7. 📌
 - **Stripe** - 📌Stripe API📌
 - 📌 - 📌
8. 📌"📌"
9. 📌

2. PDF
3. Mailjet <integrations_mailjet> api_crmCommunicationCustomerInvoice
4.
 - o PDF
 - o
 - o
 - o
 - o
 - o
 - o
5. “”

“”

- {{ var:customer_name }} -
- {{ var:invoice_number }} - ID
- {{ var:invoice_date }} -
- {{ var:due_date }} -
- {{ var:total_amount }} -
- {{ var:invoice_url }} - PDF
- {{ var:pay_url }} -

- - crm_config.yaml Mailjet API
- -
- **PDF** - PDF

- `ref` = “`payment_reference`”`StripePaymentID`

退款

如何退款

退款

- 退款
- 使用 **Stripe**
- `payment_reference` `StripePaymentID`
- 退款

退款

1. 使用 `Stripe`
2. 退款
3. 退款

4. 使用 “退款”
5. 使用 `Stripe`
 - 使用 `Stripe API`
 - 使用 `Stripe`
 - 使用 `refund_reference`
6. 退款 “退款”

7. 環境構築

環境構築

- 環境構築
- 環境構築
- 環境構築
- 環境構築3-7
- Stripe

環境構築

- POS
- 環境構築
- 環境構築

環境構築

環境構築

環境構築

- ID
- 環境構築
- 環境構築

環境構築

環境構築

環境構築

- 環境構築 - 環境構築
- 環境構築 - 環境構築

環境構築

- -
- -

- **ID** - ID /
- -
- -
- -
- - /
- -

↔

10 25 50 100

1

1. 1 31
2. →
3. "+
4.
5.
 - 2025-01-01
 - 2025-01-31
 - 2025-02-15 15
 - "2025 1
6. 1

7. 000000
 - o 0000000000
 - o 0000000000000000
 - o 00000000000
8. 00 "0000"000000000000"000000150"0
9. 0000000000000000
10. 0000000000000000
11. 0000000000 "000000" 00000000000000
12. 000000PDF0000000000000000

0000200000000000

1. 00000000000+00000000000
2. 0000000000000000
3. 0000000000000000
 - o 00 "+ 0000000"
 - o 000"00000 - 2025010"
 - o 00/00010100310
 - o 00000020150
 - o 00000000000000000000
 - 0000000000 X 00
 - 0000000000 X 00
 - 0000000000
 - o 00000000000000
 - o 00 "0000"00000000000000
4. 000000000000000000
 - o 0000 "+ 0000000"
 - o 000"00000 - 2025010"
 - o 00/0001010031000000000
 - o 000000000
 - 0000000000000000
 - 00 X 0000000000
 - 00000000000000000000
 - o 00 "0000"

5.
 - "+
 - " - 2025 "
 -
 - " "
 - "
6.

3

1.
2.
3. "+
4.
5. 20
6.
7.
 - " "
 - **X**
 - "
 - **X**
 - " 18
 -
8.
9. " **18**
10.
11. /

4

1.
2.
3.
 -

- **X** **XXXXXXXXXX**
- **XXXXXXXXXXXXXXXXXXXX**
- **X** **XXXXXXXXXX**

4. **XXXXXXXXXX**
5. **XXXXXXXXXXXXXXXXXXXX**
6. **"XXXX"****XXXXXXXXXXXX**
7. **XXXXXXXXXXXX/XXXXXXXXXX**
8. **XXXXXXXXXXXXXXXXXXXX**

XXXX**5**XXXXXXXXXXXX

1. **XXXXXXXXXXXX**
2. **XXXXXXXXXXXX**
3. **XXXX** **XX** **XX**
4. **" + XXXXXXX"**
5. **◆◆◆**
 - **XXXXXXXXXX**
 - **XXXXXXXXXX**
 - **XXXXXXXXXX + 7**
 - **"XXXX"**
6. **XXXXXXXXXXXXXXXXXXXX**
7. **XXXXXXXXXXXX**
8. **X** **XXXXXXXXXXXXXXXXXXXX**
9. **"XXXX"**
10. **XXXXXXXXXXXX**
11. **Stripe****XXXX**

XXXX**6**XXXXXXXXXXXX

1. **XXXXXXXXXXXX**
2. **XXXXXXXXXXXXXXXXXXXX**
3. **XXXX** **XXXXXXXXXXXX**
4. **XXXXXXXXXX**
 - **ID****XXXXXXXXXX**

- 0000000000
 - 0000000000
 - 0000000000
5. 0000000000000000
 6. 000000PDF00000000 "00**PDF**" 00
 7. 000000000000000000 "000000" 00
 8. 0000000000

0000**7**0000000000

1. 0000000000
2. 00000000000000
3. 000000000000000000
4. 00000000
5. 0000000000000000
6. 000000000 "00" 00
7. 00000
8. 000000000000
9. 00000000000000000000
10. 000000000000000000
 - 000000000000000000000000
 - 0000000000
11. 000000000000000000000000

0000**8**0000000000

1. 0000000000000000
2. 000000 00 00
3. 0000000000
4. 000000000000000000
5. 0000000000
6. 0000000000 "0000" 00
7. 00 "00" 00000
8. 000000"00002025-01-15"

9. `"` 閉じる
10. `"` 閉じる
11. `"` 閉じる
12. `"` 閉じる

9. Stripe API

1. Stripe API
2. Stripe API
3. Stripe API
4. Stripe API
5. Stripe API
6. Stripe API "Stripe API"
7. Stripe API
8. Stripe API
9. Stripe API "Stripe API"
10. Stripe API 3-7
11. Stripe API

PDF

PDF - PDF

- PDF
- PDF

PDF

- WeasyPrint
- HTML API

Stripe API

- Stripe API
- Stripe API

Stripe

- Stripe API
- Stripe API

Mailjet

- Mailjet API
- Mailjet API `crm_config.yaml`

Stripe

- Stripe API/POS
- Stripe API

PDF

- PDF
- PDF `DELETE FROM Invoice_PDF_Cache WHERE invoice_id = X;`

Stripe

- Stripe API
- Stripe API

Stripe

- Stripe API
- Stripe API

Stripe

- Stripe API
- Stripe API

Stripe

- Stripe API
- Stripe API

API Security

- API Security: Protecting your data and users
- API Security: Protecting your data and users 

API Security

- API Security: Protecting your data and users
- API Security: Protecting your data and users

API Security

- API Security: Protecting your data and users
- API Security: Protecting your data and users

API Security

- API Security: Protecting your data and users
- API Security: Protecting your data and users

API Security

- API Security: Protecting your data and users
- API Security: Protecting your data and users PDF

API Security

- `integrations_mailjet` - API Security
- `administration_configuration` - API Security
- `payments_transaction` - API Security
- `payments_process` - API Security
- `payment_system_guide` - API Security

Table of Contents

OmniCRM Stripe PayPal OmniCRM

<payment_system_guide> <billing_overview> <payments_process> <payments_invoices>

Table of Contents

Table of Contents

- Stripe PayPal OmniCRM
- Stripe PayPal
-
-
-
- self_care_portal
-

Table of Contents

- Visa Mastercard American Express Discover
-
-

OmniCRM

OmniCRM

- stripe paypal
- Visa Mastercard
- 4
-
-
-

□□□□□□

□□□□□□

1. □□□ □□ → **【□□□□】**
2. □□ □□ □□
3. □□□ □□□□ □□

□□□□

□□□□□□□□□□

□□□□□□□□□□□□

□□□□□□□□□□□□□□60□□□□□□□□□□

□□□□□□

□□□□□□□□□□□□□□

□□□□

- □□ - □□□□□□□□□□“□□□”□“□□Visa”□
- □□□ - □□□□□□4□□□
- □□ - □□□/□□MM/YYYY□□□□

- 00 - 0000000000
- 00 - 000000000000000000

00000000

000000000000 : 00000000

- 0000 - 000000000000
- 00 - 00000000

00000000

00 "000000" 000000000000

00**1**000000000000

000000000000 Stripe Elements PayPal SDK 00000000

000000

- 000 - 000CVC Stripe 000
- 000000 - 000000
- 00/00 - 0000

Overview

- Introduction - Project Goals

Requirements

- Integrate payment gateways (Stripe Elements / PayPal SDK)
- OmniCRM integration
- PCI DSS compliance
- User authentication

2. Implementation

2.1. "Payment" Module

1. Frontend
 - Card input fields
 - Address input fields
 - CVC field
2. Backend
 - Connect to OmniCRM
 - Connect to Stripe (API key: pm_1A2B3C4D)
 - Connect to OmniCRM
3. Security
 - OmniCRM security audit
 - 4-step security process
 - OmniCRM security audit

3. Testing

3.1. Unit Tests

Unit tests for Visa card number 1234

3.2. Integration Tests

3.3. Security Tests

- 00000000000000000000
- 00000000000000000000
- 0000000000000000

0000000000

0000000000

- 00000000
- 0000
- 00000
- 0000000000000000

00000000

1. 00000000000000000000
2. 000000 000000 : 0
3. 00 "00000"
4. 00000000

Visa00000005678000000000000000

0000000000000000

000000

0000000000000000000000000000

0000000000

00000000000000000000

模块1 数据基础

- 数据库的基本概念
- 数据库系统组成：DBMS, DBA, DB, SDB
- 数据库"三要素"

模块2 数据操作

数据操作基础

数据库操作的基本语句

插入: INSERT INTO table (column1, column2) VALUES (value1, value2)

更新: UPDATE table SET column1 = value1, column2 = value2

删除: DELETE FROM table WHERE condition

数据库"完整性"的概念

模块3 数据应用

数据应用案例

数据库应用案例: Stripe 支付系统

数据应用案例

- 数据库的索引 - 提高查询效率
- 数据库的分区 - 提高可扩展性
- 数据库的备份 - 保证数据安全

模块4 数据生态

OmniCRM 数据生态系统的构建

□□□□□□□□

□□□ □□ → □□□□□□ □□□□□□□□□□

□□ □□ □□□□□□□□ John Smith Visa ****1234 02/2025 12** □□ **Acme Corp**
MC5678 03/2025 45 □□ **Jane Doe Amex****9012 01/2025 □□□ □□

□□□□

- □□□□ - □□□□30/60/90□□□□□□
- □□□□ - □□□□□□
- □□□□ - □□□□□□□□□□□□

□□□□

- □□ - □□□□□□□□□□□□□□□□
- □□ - □□□□□□□□□□□□□□□□[Mailjet](#)

□□□□□

□□□□□[Mailjet](#)□□□□□□□□□□□□

- □□□**60**□ - □□□□□□

- **30** -
- **7** -
- -

Mailjet

-
- **4**
-
-

`integrations_mailjet`

- 1.
- 2.
3. →
4. " "
- 5.
- 6.
7. /

- 1.
2. →
- 3.
- 4.

5. 圖表

6. 圖表

圖

圖表

- 圖表
- 圖表
- 圖表

圖表

圖表

圖表

1. 圖表

- 圖表
- 圖表
- 圖表
- 圖表

2. 圖表

- 圖表
- 圖表
- 圖表“圖表”

3. 圖表

- 圖表 - 圖表
- 圖表 - 圖表
- 圖表 - 圖表

圖表

1-3 圖表

- 圖表
- 圖表

- 000000000000000

4-7

- 000000000000000
- 00000000
- 00000000000

8-14

- 000000000000000
- 000000000000000
- 000000000000000000000000000

15

- 000000000000000
- SIM 000000000000
- 0000000
- 000000000000000

0000000

00000000

- 0000300000
- 000000000000
- 000000000
- 000000000000000

00000000

000000000000000

1. 00000000000
2. 0000000
3. 00000000000
4. 0000000000000

5. 銀行振込

銀行振込

銀行

OmniCRM銀行振込

1. 銀行 → 銀行振込
2. 銀行振込 → 銀行
3. 銀行OmniCRM → 銀行振込
4. 銀行 → 銀行振込

OmniCRM銀行振込

```
{
  "vendor": "stripe",
  "vendor_payment_method_id": "pm_1A2B3C4D5E6F",
  "payment_type": "card",
  "brand": "visa",
  "last4": "1234",
  "exp_month": 12,
  "exp_year": 2026,
  "name": "John Smith",
  "nickname": "銀行",
  "is_default": true
}
```

OmniCRM銀行振込

- 銀行
- CVV/CVC
- 銀行
- PIN

PCI

- PCI - OmniCRM
- - iframe
- -
- - HTTPS/TLS

<payment_system_guide>

1

- 1.
- 2.
- 3.
4. Stripe
- 5.
- 6.
- 7.
- 8.

2

□□ **1**□□□□□□

◆◆ □□ □□□□□□□□□□

□□□□□

- □□□□□□□□□□
- □□□□□□□□
- □□□□□□□□□□ Stripe□
- □□□□□□□□□□

📁 basics_payment

📄 2📄

📄 📄

📄

- 📄
- 📄/📄
- 📄
- 📄
- 📄

📁 payments_transaction

📄 3📄

📄 📄

📄

- 📄
- 📄/📄
- 📄
- 📄 PDF
- 📄
- 📄Stripe 📄

- payments_invoices

payments_invoices

payments_invoices

payments_invoices

payments

payments → invoices

1. payments

payments

- invoice_id
- invoice_id
- invoice_id null

payments

2. 查詢

查詢所有發票

- 查詢所有發票
- 查詢所有發票的總金額
- 查詢 `invoice_id` 的發票
- 查詢所有發票的日期

查詢

3. 更新

- 更新發票
- 刪除發票
- 更新發票的日期

查詢 → 更新

1. 查詢

查詢所有發票

- 查詢所有發票
- 查詢所有發票的總金額
- 查詢所有發票的日期

2. 更新

查詢所有發票 Stripe

- 查詢所有發票
- 查詢所有發票的總金額
- 查詢所有發票的日期
- 查詢 `invoice_id` 的發票

查詢

3. 設定

- 設定項目
- 設定項目

設定 → 設定

Stripe 設定

1. 設定項目
2. Stripe 設定項目
3. 設定項目
4. Stripe 設定
5. 設定項目
6. `payment_reference` 設定 Stripe 設定 ID

設定項目

1. 設定項目/POS/設定項目
2. 設定項目
3. 設定項目
4. 設定項目
5. 設定項目

設定項目

設定項目

設定項目 **1** 設定項目

設定項目

1. 設定項目
 - 設定項目 → 設定項目
 - 設定項目 → 設定 "設定項目"
 - 設定項目 Stripe 設定項目

- 確認金額

2. 確認

- 金額を確認
 - 金額 \$50.00
 - 金額 \$45.00
 - 金額 \$95.00

3. 確認

- 金額 → 金額 "金額"
- 金額
- 金額 "金額"
- 金額 #INV-2025-001234 金額 \$95.00

4. 確認

- 金額 #INV-2025-001234
- 金額 \$0.00

5. 確認

- 金額
- 金額 PDF 金額

6. 確認

- 金額
- 金額
- 金額 "金額" 金額
- 金額
- 金額 "金額"
- Stripe 金額

7. 確認

- 金額 "金額"
- 金額 \$95.00

金額

金額 2金額

金額



1. 確認

- 00000103100
- 00000000000000000000
- 00 000000000000

2. 00000000

- 000 00 00
- 000 00000000
- 00000000000000
- 000000000000

3. 00000

- 000 00 → 00000000
- 00   00000000000000
 - 00 "000000"
 - 0000
 - 000002025-01-01
 - 000002025-01-31
 - 000002025-02-15
 - 00 "0000"

4. 00000

- 000000000000
- 00000 00000 \$0.00

5. 000000000000

- 0000000000000000
- 000000000000

6. 00000

- 0000000000000000000000
- 0000000000000000/POS 00
- 00000 000000000000

000 00000100000000000000000000

00000 **3**000000000000

000 0000000000000000000000

1. 00000000

- Stripe 查詢 "ID"

- 查詢 ID 的資料

2. 查詢 ID

- 查詢 ID 的資料
- 查詢 ID 的資料 → 查詢 "ID"
- 查詢 ID 的資料
- 查詢 ID

3. 查詢 ID

- 查詢 "ID" 的資料
- 查詢 ID 的資料

4. 查詢 ID

- 查詢 ID 的資料
- 查詢 ID 的資料
- 查詢 "ID" 的資料
- 查詢 ID 的資料
- 查詢 "ID"
- Stripe 查詢 ID

5. 查詢 ID

- 查詢 ID 的資料 "ID"
- `payment_reference` 查詢 Stripe 查詢 ID

查詢 ID 的資料

查詢 ID 5 查詢 ID 的資料

查詢 ID 的資料 - 查詢 ID 的資料

1. 查詢 ID

- 查詢 #INV-2025-001234 查詢 ID 的資料
- 查詢 ID 的資料
- 查詢 ID

2. 查詢 ID

- 查詢 ID 的資料 → 查詢 ID
- 查詢 ID 的資料
- 查詢 ID (ID)

- 0000
- 00000

3. 00000

- 000 00 00
- 00000000000000000000 000
- 00000 0000
- 000000000

4. 000000000

- 00 00 → 00 "000000"
- 0000000000
- 000000000000000000"00"0000000000
- 00 "0000"
- 0000000000000000

5. 0000000000

- 00000000
- 000000000000
- 00000000000000

000 00000000000000000000000000000000000000

0000 **6**0000000000000000

000 00000000000000000000

1. 0000000000

- 0000 \$300 0000000000
- 00000 → 00 000

2. 0000000000

- 00 00 → 000 0000000000
- 0000000000

3. 0000000000

- 0000 #1234 00000000
- 00000000
- 00 "00" 0000
- 000000"0000 2025-02-10 - 00 #001"

- 姓名 "张小明"
- 身份证号 #1234 姓名 "张小明"

4. 工作经历

- 公司 #1235 工作经历
 - 职位 "产品经理 2025-02-10 - 职位 #001"
- 公司 #1236 工作经历
 - 职位 "产品经理 2025-02-10 - 职位 #001"

5. 教育背景

- 学校 姓名 性别
- 学历 学位 专业
- 毕业时间 毕业院校

6. 其他信息

- 电子邮箱 → 手机号码 金额 \$300.00
- 联系地址 姓名 电话

姓名 身份证号 姓名 身份证号 姓名 身份证号

姓名

姓名

姓名

- 姓名
- 姓名
- 姓名
- 姓名



姓名

- 姓名 身份证号 姓名 身份证号 100
- 姓名
- 姓名
- 姓名
- 姓名 PDF

□□□□

- □□□□□□□□□□□□□□□□
- □□□□□□□□□□POS□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□ Stripe □□□□□□□□□□□□□□□□□□□□□□

□□□□

- □□□□□□□□□□
- □□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□

□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□

□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□ PDF □□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

- □□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□

□□□□□

- □□ Mailjet □□□□□□□□□□□□□□
- □□□□□ PDF □□□□□□□□
- □□□□□□□□□□ Stripe □ PayPal □□□□□□□□□□
- □□□□□□□□□□□□

□□□□□□

- □□□□□□□□□□□□□□
- □□ □□□□□ □□ - □□□□□□□□□□
- □□ □□□□□ □□ - □□□□□□□
- □□ □□ □ □□ □□□□□□□□

□□□□□

- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□

□ □□□□□□□□□□

□□□□□□□□□□□□

□□□

- □□□□□□□□□□□□□□
- □□□□□□“□□□□□□□□□□”

□□□

1. □□□□□ → □□ □□□ → □□□□□ □□
2. □□□□□□□□□□□□□□□□
3. □□□□□□□□□□□□

1. 卡 牌
2. 卡 號
3. 卡 類 別 - 卡 種
4. 卡 片 號 碼

卡 號

- 卡 號 的 最 後 四 位 數 字 是 卡 片 號 碼
- 卡 號 的 前 六 位 數 字 是 卡 類 別
- 卡 號 的 中 間 七 位 數 字 是 卡 種

卡 類 別

卡 種

- Stripe 卡 號 類 別 “ 卡 號 ”
- 卡 號 類 別
- 卡 號 類 別

卡 種

1. 卡 號 Stripe 卡 號 類 別
2. 卡 號 類 別
 - 卡 號
 - 卡 號
 - 卡 號 類 別 / 卡 號
 - 卡 號
 - 卡 號

卡 種

- 卡 號 類 別
- 卡 號 類 別
- 卡 號 類 別 POS 卡 號 類 別
- 卡 號 類 別

□□ ? ? ? □□□

□□□

- □□□□□□□□□□□□ → **B** □□□□□□□□□□□□

□□□

- □□□ □□□□□□□□□□ "□□□" □□□□□□□□□□□□
- □□□ □□ "□□□□" □□□□□□□□□□□□□□□□□□

□□□□□

- □□□□□□□□□□□□□□□□/□□□□□□□□□□□□
- □□□□□□□□ ID □□□□□□□□□□

□□□□□

- □□□□□/□□□□□□□□□□□□□□□□
- □□□□□□□□□ (:) □□□□□□□□□□□□□□□□□□□□

□□□□□

- `basics_payment` - □□□□□□□□□□□□□□□□
- `payment_system_guide` - □□ API □□□□□□□□□□
- `payments_transaction` - □□□□□□□□
- `payments_invoices` - □□□□□□□□
- `integrations_mailjet` - □□□□□□□□□□
- `csa_activity_log` - □□□□□□□□□□
- `□□□□□` - OCS/CGRateS □□□□□□□□□□

CGRateS

CGRateS is a Python-based interface for OmniCRM that allows you to interact with the system via REST API.

API

OmniCRM provides a REST API (CGRateS) that allows you to interact with the system via REST API.

CGRateS provides a REST API that allows you to interact with the system via REST API. Playbooks are a set of actions that can be executed in a specific order.

API

API - [CGRateS API](#) CDR API

API - [CGRateS API](#)

API ID - [CGRateS API](#) "API" "API"

API - [CGRateS API](#)

API - [CGRateS API](#) "+5 API"

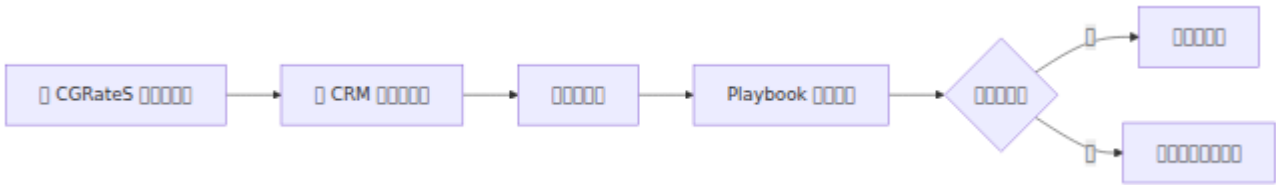
API - [CGRateS API](#) [API](#)

API

ExecuteAction **Playbook** is a Python-based interface for CGRateS that allows you to interact with the system via REST API.

API

CGRateS is a Python-based interface for CRM and OCS.



□□□□□□□□

□□□□□□□□□□□□

- **CGRateS** □□□ `ActionsId = "Action_50gb-data-pack"`
- **CRM** □□□ `product_slug = "50gb-data-pack"`
- □ **Playbook** □□ `cgr_action_name = "Action_" + product_slug`

□ Playbook □□□□□□ `product_slug` □□□□□□□□□□ `ExecuteAction` □□□□□□□□ `CGRateS` □□□□□□□□□□

□□□□□□□□

□□□□□□□□□□□□□□□□□□

1. □□□□□□ - □□□□□□□□
2. □□□□□□ - □□□□□□□□□□□□
3. □□□□□□ - □□ OCS □ CRM □ Python □□

□□□□□□□□□□□□□□

```

import cgrateshttpapi

OCS_Obj = cgrateshttpapi.CGRateS("ocs.example.com", "2080")
tenant = "your_tenant"

# 1. Create CGRateS object
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"

# 2. Create CRM object
# (product_slug = "50gb-data-pack" Action_50gb-data-pack)

```

XXXXXXXXXXXXXXXXXX

```

# Playbook
- name:
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body:
      {
        "method": "APIerSv1.ExecuteAction",
        "params": [{
          "ActionsId": "Action_50gb-data-pack"
        }]
      }
  register: response

#
# response.json.error = "SERVER_ERROR: "
#

```

-
-
-

□□□

```
# □□□□□
□□□
  ID: "Data_Package__5368709120"
  □: 1073741824 # □□ 1GB
  □□: 2024-12-24□□□ 1 □□

# □□ *topup □□□
□□□
  □□□: "*topup"
  □□□
    ID: "Data_Package__5368709120" # □□ ID - □□□□
    □: 5368709120 # 5GB
    ExpiryTime: "+5d"

# *topup □□□□□
□□□
  ID: "Data_Package__5368709120"
  □: 6442450944 # 6GB□1GB + 5GB □□□
  OriginalValue: 5368709120 # □□□□□□ 5GB
  Value_hr: "6 GB"
  OriginalValue_hr: "5 GB"
  Remaining_hr: "6 GB of 5 GB (1 GB rolled over)"
  □□: 2024-12-29□□□□□ 5 □□
```

□□□

- □□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□
- □□□□□

*topup_reset - □□□□

*topup_reset □□ □□□□□□□□□□□□□□ ID□

□□□

- □□□□□□ ID □□□□□□

- 0000000000000000
- 0000000000
- 0000000000000000

000

```
# 000000
000
ID: "Data_Package__5368709120"
[]: 1073741824 # [] 1GB
[]: 2024-12-240000 1 []

# [] *topup_reset []
[]
[]: "*topup_reset"
[]
ID: "Data_Package__5368709120" # [] ID - []
[]: 5368709120 # 5GB
ExpiryTime: "+5d"

# *topup_reset []
[]
ID: "Data_Package__5368709120"
[]: 5368709120 # 5GB[] 1GB []
OriginalValue: 5368709120
Value_hr: "5 GB"
Remaining_hr: "5 GB of 5 GB"
PercentUsed: 0
[]: 2024-12-29000000 5 []
```

000

- 0000000000000000
- 0000000000000000
- 0000000000000000
- 0000000000000000

顧客 ID 取得

顧客 ID 取得

顧客 ID 取得

OmniCRM 顧客 ID 取得 API Web UI

顧客 ID

```
{顧客ID}__{顧客ID}
```

顧客 ID

```
# 顧客ID100GB
"AU_Data_Domestic__107374182400"
# 顧客ID
# - 顧客ID"AU_Data_Domestic"顧客ID - 顧客ID/顧客ID
# - 顧客ID"__"顧客ID
# - 顧客ID"107374182400"顧客ID100GB 顧客ID
# - UI 顧客ID"AU Data Domestic - 100 GB"

# 顧客ID3000min
"AU_Voice_Domestic__1800000000000000"
# - 顧客ID"AU_Voice_Domestic"顧客ID "AU_Voice_Domestic_3000min"
# - 顧客ID"1800000000000000"顧客ID3000顧客ID
# - UI 顧客ID"AU Voice Domestic - 3000 min"

# 顧客ID3000msg
"AU_SMS_Domestic__3000"
# - 顧客ID"AU_SMS_Domestic"
# - 顧客ID"3000"顧客ID
# - UI 顧客ID"AU SMS Domestic - 3000 msgs"
```

顧客 ID 取得 - 顧客 ID 顧客 ID API 顧客 ID

API 顧客 ID

OmniCRM API CGRateS 顧客 ID 顧客 ID _hr 顧客 ID

```

{
  "BalanceMap": {
    "*data": [
      {
        "ID": "AU_Data_Domestic__107374182400",
        "Value": 53687091200,
        "ExpiryTime": "2025-01-25T23:59:59Z",
        "Weight": 1200,

        // 残り容量
        "ID_hr": "AU Data Domestic",
        "OriginalValue": 107374182400,
        "OriginalValue_hr": "100 GB",
        "Value_hr": "50 GB",
        "Remaining_hr": "50 GB of 100 GB",
        "PercentUsed": 50,
        "ExpiryTime_hr": "2025年1月25日22時"
      }
    ]
  }
}

```

API 実装

1. ID 取得

```

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. 残り容量取得

```

# 残り容量
id_hr = descriptive_name.replace("_", " ") # "AU Data
Domestic"

```

3. 〇〇〇〇〇〇〇〇〇〇〇〇

```
# 〇〇〇〇〇〇〇〇〇〇
if balance_type == "*data":
    original_value_hr = convert_bytes_to_gb(original_size) #
    "100 GB"

# 〇〇〇〇〇〇〇〇〇〇〇〇
elif balance_type == "*voice":
    original_value_hr = convert_ns_to_minutes(original_size) #
    "3000 min"

# 〇〇〇〇〇〇〇〇〇〇〇〇
elif balance_type == "*sms":
    original_value_hr = f"{original_size} msgs" # "3000 msgs"
```

4. 〇〇〇〇〇〇〇〇

```
if original_size and original_size > 0:
    percent_used = ((original_size - current_value) /
original_size) * 100
```

5. 〇〇〇〇〇〇〇〇

```
remaining_hr = f"{current_value_hr} of {original_value_hr}"
# "50 GB of 100 GB"
```

Web UI 〇〇〇

〇〇〇〇〇〇 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇

```
// 0000000000
// ID: "AU_Data_Domestic__107374182400"
// 0: 53687091200

// 00000000
<BalanceCard>
  <Title>{balance.ID_hr}</Title>           {/* "AU Data Domestic"
*/}
  <Progress value={balance.PercentUsed}> {/* 50% */}
    {balance.Remaining_hr}               {/* "50 GB of 100 GB"
*/}
  </Progress>
  <Expiry>{balance.ExpiryTime_hr}</Expiry> {/* "2025-1-25-22"
*/}
</BalanceCard>
```

00000000

- 000000 - 0000000000 UI 000000 "50 GB of 100 GB"000000 "50 GB 00"
- 000000 - 000000000000000000
- 000000 - 0000 ID 0000000000000000000000
- 0000 - 0000 *topup000000000000 70 GB 000000 100 GB
 - 00 ID 000000 "AU_Data_Domestic__107374182400" 0000 100 GB
 - 00000000170 GB
 - UI 0000 "170 GB (70 GB rolled over + 100 GB new)"

0000 - 0000 ID

0000 0000 0000000000000000 UI 000000000000000000

```
# 1 - 100GB + 100GB
Action_Data_100GB = {
    "Actions": [
        {
            "BalanceId": f"AU_Data_Domestic_{100 * 1024 * 1024 *
1024}",
            "Units": 100 * 1024 * 1024 * 1024
        }
    ]
}

# 2 - 100GB
Action_Data_100GB = {
    "Actions": [
        {
            "BalanceId": f"AU_Data_Domestic_100GB_{100 * 1024 *
1024 * 1024}", # 100
            "Units": 100 * 1024 * 1024 * 1024
        }
    ]
}

# 3 - 100GB UI 100GB
Action_Data_100GB = {
    "Actions": [
        {
            "BalanceId": "AU_Data_Domestic", # 100 __size
            "Units": 100 * 1024 * 1024 * 1024
        }
    ]
}
```

100GB - 100GB

100GB 100GB 100GB 100GB 100GB

```
# ██████████  
{  
  "BalanceId": "PAYG_Monetary_Balance",  
  "BalanceType": "*monetary",  
  "Units": 5000 # $50.00  
}  
  
# UI ██████████  
# "$50.00"
```

1 ID

ID UUID

```

- name: 充值套餐
  set_fact:
    uuid: "{{ 99999999 | random | to_uuid }}"
    balance_id: "Data_5days__5368709120_{{ uuid[0:8] }}"

- name: 充值 *topup 套餐
  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
          }
        }]
      }

```

充值套餐 *topup 套餐

```
# 0000 "5 00000000" 0000
00 1:
  ID: "Data_5days__5368709120_a1b2c3d4"
  0: 5368709120
  Value_hr: "5 GB"
  OriginalValue_hr: "5 GB"
  Remaining_hr: "5 GB of 5 GB"
  00: 2024-12-29

00 2:
  ID: "Data_5days__5368709120_e5f6g7h8"
  0: 5368709120
  Value_hr: "5 GB"
  Remaining_hr: "5 GB of 5 GB"
  00: 2024-12-30

00 3:
  ID: "Data_5days__5368709120_i9j0k1l2"
  0: 5368709120
  Value_hr: "5 GB"
  Remaining_hr: "5 GB of 5 GB"
  00: 2024-12-31

# 000015GB000000000000
# 000 UI 0000 "Data 5days - 5 GB of 5 GB"
```

00 2000 ID 0 *topup0000

0000000 ID 0 *topup 0000000000000000

00000

```
- name: 账单 ID
  set_fact:
    balance_id: "Data_5days__5368709120"

- name: 账单 *topup 账单 ID
  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
          }
        }]
      }
}
```

账单 ID 账单 ID

```
# 1 5GB
ID: "Data_5days__5368709120"
: 5368709120
Value_hr: "5 GB"
Remaining_hr: "5 GB of 5 GB"
PercentUsed: 0
: 2024-12-29
```

```
# 3 1GB
ID: "Data_5days__5368709120"
: 9663676416 # 4GB + 5GB
Value_hr: "9 GB"
OriginalValue_hr: "5 GB"
Remaining_hr: "9 GB (4 GB rolled over + 5 GB new)"
PercentUsed: -80 #
: 2024-12-27 +5
```

3 ID *topup_reset

ID *topup_reset

```
- name: 000000 ID
  set_fact:
    balance_id: "Monthly_Plan__32212254720"

- name: 00 *topup_reset 0000000000
  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
          }]
        }]
      }
```

000 00000000000000 30GB0000000000

```
# 1 30GB 1 GB
ID: "Monthly_Plan__32212254720"
#: 32212254720
Value_hr: "30 GB"
Remaining_hr: "30 GB of 30 GB"
PercentUsed: 0
: 2024-12-31

# 1 25 28GB
ID: "Monthly_Plan__32212254720"
#: 2147483648
Value_hr: "2 GB"
Remaining_hr: "2 GB of 30 GB"
PercentUsed: 93
: 2024-12-31

# 2 1 ActionPlan *topup_reset
ID: "Monthly_Plan__32212254720"
#: 32212254720 # 2GB
Value_hr: "30 GB"
Remaining_hr: "30 GB of 30 GB"
PercentUsed: 0
: 2025-01-31
```

CGRateS

CGRateS 阻塞器 2GB 阻塞器

阻塞器

Blocker: true

1. 阻塞器 → 阻塞器
2. 阻塞器 → 阻塞器
3. 阻塞器 → INSUFFICIENT_CREDIT_BALANCE_BLOCKER 阻塞器

□□□□

- □□□□□□□□□□ □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□CGRateS □□□□ □□□□
- □□□□□□□□□□ □*voice□□*data□□*sms□□*monetary

□□□□□□

1. □□□□

□□□□□□□□□□□□□□□□□□□□

```

Action_Suspend_Account = {
  "id": "0",
  "method": "ApierV1.SetActions",
  "params": [{
    "ActionsId": "Action_suspend-account",
    "Overwrite": True,
    "Tenant": tenant,
    "Actions": [
      # □□□□□□□□□□□□□□□□
      {
        "Identifier": "*topup",
        "BalanceId": "Suspension_Blocker",
        "BalanceType": "*monetary",
        "DestinationIDs": "*any",
        "Units": 0, # □□
        "BalanceWeight": 9999, # □□□□□ - □□□□
        "Blocker": True, # □□□□□□
        "Weight": 10
      }
    ]
  }]
}

result = OCS_Obj.SendData(Action_Suspend_Account)

```

□□ □□□□/□□/□□□□ □□□□□□□□□□□□

2. □□□□

□□□□□□□□□□□□□□

```
Action_Monthly_Plan_With_Cap = {
  "id": "0",
  "method": "ApierV1.SetActions",
  "params": [{
    "ActionsId": "Action_monthly-with-cap",
    "Overwrite": True,
    "Tenant": tenant,
    "Actions": [
      # 10GB □□□□
      {
        "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, # □□□□
        "Weight": 95
      },
      # $50 □□□□□□□□□□
      {
        "Identifier": "*topup_reset",
        "BalanceId": "Overage_Cap",
        "BalanceType": "*monetary",
        "DestinationIDs": "*any",
        "Units": 5000, # $50.00 □□□□
        "ExpiryTime": "*month",
        "BalanceWeight": 1000, # □□□□□□
        "Blocker": True, # □ $50 □□□□□□□□
        "Weight": 90
      }
    ]
  }
}]
}
```

□□□

1. □□□□ 10GB □□ → □□□□□□ Included_Data_10GB□
2. □□□□□□□□ 5GB → □ Overage_Cap □□□□□□ PAYG □□□

3. Overage_Cap 0 \$0 →

3.

```

Action_Trial_Account = {
  "id": "0",
  "method": "ApierV1.SetActions",
  "params": [{
    "ActionsId": "Action_trial-100-minutes",
    "Overwrite": True,
    "Tenant": tenant,
    "Actions": [
      # 100 -
      {
        "Identifier": "*topup",
        "BalanceId": f"Trial_Voice_{100 * 60 *
10000000000}",
        "BalanceType": "*voice",
        "DestinationIDs": "Dest_Domestic_All",
        "Units": 100 * 60 * 10000000000, # 100
        "ExpiryTime": "+720h", # 30
        "BalanceWeight": 1200,
        "Blocker": True, # 100
        "Weight": 10
      }
    ]
  }
]
}

```

100

4.

```

Action_Block_Premium_Numbers = {
  "id": "0",
  "method": "ApierV1.SetActions",
  "params": [{
    "ActionsId": "Action_block-premium",
    "Overwrite": True,
    "Tenant": tenant,
    "Actions": [
      # 普通充值
      {
        "Identifier": "*topup_reset",
        "BalanceId": "Regular_Usage",
        "BalanceType": "*monetary",
        "DestinationIDs": "*any",
        "Units": 10000, # $100
        "ExpiryTime": "*month",
        "BalanceWeight": 1000,
        "Weight": 20
      },
      # 普通充值0900 限制
      {
        "Identifier": "*topup",
        "BalanceId": "Premium_Blocker",
        "BalanceType": "*monetary",
        "DestinationIDs": "Dest_Domestic_Premium",
        "Units": 0, # 0
        "BalanceWeight": 2000, # 2000 - 10000
        "Blocker": True,
        "Weight": 10
      }
    ]
  }]
}

```

普通

- 普通 → 普通 Regular_Usage 普通
- 普通限制 → 普通 Premium_Blocker 2000 > 1000 → 普通

□□□□□□□□□□

□□□ Blocker Disabled □□□

□□	□□□	□□
□□	□□□□□□□□□□	□□□□□□
□□ > 0	□□□□□□	□□□□□□/□□
□□ = 0	□□□□□□□□□□	□□□□□□□□□□□□□□
□□	□□□□□□□□□□□□	□□□□□□□□

```
# □□□□□□ 10GB□□□□□□□□□□
{
  "BalanceId": f"Data_Cap_{10 * 1024 * 1024 * 1024}",
  "Units": 10 * 1024 * 1024 * 1024,
  "Blocker": True #  10GB □□□□□
}

# □□□□□□□□□□□□□□□□□□
{
  "BalanceId": f"Bonus_Data_{5 * 1024 * 1024 * 1024}",
  "Units": 5 * 1024 * 1024 * 1024,
  "Disabled": True # □□□□□□□□□□
}
```

□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□

```

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 000000
      {
        "Identifier": "*topup_reset",
        "BalanceId": f"Domestic_Voice__{500 * 60 *
10000000000}",
        "BalanceType": "*voice",
        "DestinationIDs": "Dest_Domestic_All",
        "Units": 500 * 60 * 10000000000,
        "ExpiryTime": "*month",
        "BalanceWeight": 1200,
        "Weight": 95
      },
      # $20 0000
      {
        "Identifier": "*topup_reset",
        "BalanceId": "Overage_Allowance",
        "BalanceType": "*monetary",
        "DestinationIDs": "*any",
        "Units": 2000, # $20.00
        "ExpiryTime": "*month",
        "BalanceWeight": 1000,
        "Weight": 90
      },
      # $50 00000000
      {
        "Identifier": "*topup_reset",
        "BalanceId": "Hard_Spending_Cap",
        "BalanceType": "*monetary",
        "DestinationIDs": "*any",
        "Units": 5000, # $50.00 0000
        "ExpiryTime": "*month",

```

```

    "BalanceWeight": 500, # 0000 - 0000
    "Blocker": True, # 00000000
    "Weight": 85
  }
]
}]
}

```

00000

1. **0-500** 000 00 Domestic_Voice_500min 0000
2. **500-700** 000 00 Overage_Allowance 0000 \$0.10 = \$20 200 000
3. **700-1200** 000 00 Hard_Spending_Cap 0000 \$0.10 = \$50 500 000
4. **1200** 0000 Hard_Spending_Cap 00 → 00000000

000000 1200 00000000 \$50

0000000000

1. 0000000000

```

"BalanceWeight": 9999 # 0000000000

```

2. 000000000000

```

"Units": 0, # 0000
"Blocker": True

```

3. 000000000000

- 00 ActionTriggers 0 80% 90% 100% 0000000000
- 00000000000000000000

4. 000000000000

```
# 00 *remove_balance 000000
{
  "Identifier": "*remove_balance",
  "BalanceId": "Suspension_Blocker"
}
```

5. 00000000

- 00000000 INSUFFICIENT_CREDIT_BALANCE_BLOCKER 00
- 00 CDR 0000 = -1.0 000000
- 000000000000000000000000

00000000

000000000000000000000000

000000

- 000000000000000000000000
- DestinationIDs 0000000000
- 0000000000 True

000000

```
# 00000000
OCS_Obj.SendData({
  'method': 'ApierV2.GetAccount',
  'params': [{"Tenant": tenant, "Account": "service_uuid"}]
})

# 0000
# - Blocker: true
# - BalanceWeight 000000009999
# - DestinationIDs 00000000
# - 0 = 0
```

0000000000

000000 00000000000000000000

Item 1:
ID: "Premium_Data"
Size: 5GB
Count: 20

Item 2:
ID: "Standard_Data"
Size: 10GB
Count: 10

Item 1 count (20) > Item 2 count (10)
Item 2 size (10GB)

Item 1 is larger than Item 2

Item 3

Item 3 is the largest

Item 1:
ID: "Data_Package_A"
Size: 5GB
Date: 2024-12-25
Count: 10

Item 2:
ID: "Data_Package_B"
Size: 10GB
Date: 2025-01-15
Count: 10

Item 2 is larger than Item 1

```
# 数据包的配置信息
包 1:
  ID: "Data_Package_A"
  容量: 5GB
  生效日期: 2024-12-25
  包号: 11 # 包号 = 容量

包 2:
  ID: "Data_Package_B"
  容量: 10GB
  生效日期: 2025-01-15
  包号: 10 # 包号 = 容量

# 包 1 的容量 11 > 包 10
```

数据包的配置信息

包号

包 1

包的容量“5GB 5”

包

```

- name: 随机 UUID 随机字符串
  set_fact:
    uuid: "{{ 99999999 | random | to_uuid }}"

- name: 随机生成 5GB 数据
  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天
            "Weight": 10
          }
        }]
      }

```

测试

- 12月24日测试 → 5GB 2024年12月29日
- 12月25日测试 → 5GB 2024年12月30日
- 测试权重为10

2

“5GB”

```

- name: 充值接口
  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天
            "Weight": 10
          }
        }]
      }

```

充值接口 *topup 充值接口

充值接口

- 1 充值 50GB 30 天
- 20 充值 30GB 20GB
- 20 充值 → 70GB 20GB + 50GB 20 天 + 30 天

充值 3 充值接口

充值 “100GB 充值” 充值接口 100GB 充值

充值

```
- name: []
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: [] ActionPlan
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: [] ActionPlan []
```

```
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
      }]
    }
  }
```

□□□□□

- □ 1 □□□□ 1 □□□□ 100GB□□□□ 95GB□□□□ 5GB
- □ 2 □□□□□□□□□□ 100GB□□□□□□□□ 5GB □□□□
- □ 2 □□□□□□ 20GB□□□□ 80GB
- □ 3 □□□□□□□□□□ 100GB□□□□□□□□ 80GB □□□□

□□ **4**□□□□□□□□□□□□□□

□□□□ □□□□“□□□□”□□□□□□□□□□“□□□□”□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

```
# 充值套餐
- name: 充值套餐
  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天
            "Weight": 20
          }
        }]
      }
```

```
# 充值套餐
- name: 充值套餐
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "Regular_Data",
            "Value": 53687091200, # 50GB
            "ExpiryTime": "+720h", # 30天
            "Weight": 10 # 10天
          }
        }]
      }
```

□□□□

- □□ 5GB □□□□□□ 20□+ 50GB □□□□□□ 10□
- □□□□□□ 5GB □□□□
- □□□ 50GB □□□□□□□□
- □□□□□□□□□□

□□□□□□□□

CGRateS □□□□□□□□□□□□□□□□

□□□□

- *topup** - □□□□□□□□□□□□
- *topup_reset** - □□□□□□□□□□□□□□
- *debit** - □□□□□□
- *debit_reset** - □□□□□□□□
- *reset_account** - □□□□□□

□□□□□□□□

□ OmniCRM □□□□□□□□□□□□□□□□□□□□□□

□□ **1**□□□□□□□□□□□□

- → □□□□□□ ID□□ UUID□
- → □□□□□□ ID □ ***topup_reset**
- → □□□□□□ ID □ ***topup**

□□ **2**□□□□□□□□□□□□□□□□

- → □□ ***topup** □□
- → □□ ***topup_reset** □□
- → □□□□□□ ID

3

→

→ =

→ 30 20 10

→

4

→ +720h 30

→ ActionPlan *monthly

→ *unlimited

CGRateS Playbooks

```

- name: 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, #
          "Actions": [
            {
              "Identifier": "*topup", # *topup_reset
              "BalanceType": "data", # *data*voice*sms
              "Units": 5368709120, #
              "ExpiryTime": "+120h", # +Xh*unlimited*monthly
              "DestinationIds": "*any", # *any
              "BalanceId": "Balance_Name", #
              "Weight": 10, # =
              "Blocker": false, # true
              "Disabled": false, # true
              "SharedGroups": "" #
            },
            {
              "Identifier": "*cdrlog", # CDR
              "BalanceType": "generic",
              "ExtraParameters": "{\
                \"Category\": \"^activation\", \"Destination\": \"Addon Name\"}"
            }
          ]
        }]
      }

```

CGRateS

ActionsId -

Identifier - *topup *topup_reset *cdrlog

BalanceType - 000000

- *data - 0000000000
- *voice - 00000000
- *sms - 0000000000
- *monetary - 000000000000
- *generic - 0000

Units - 00/000000000000000000000000

ExpiryTime - 00000000

- +Xh - 000000 +720h = 30 00
- *unlimited - 0000
- *monthly - 00
- 2024-12-31T23:59:59Z - 000000

BalanceId - 0000000000 ID = 000000 ID = 0000

Weight - 0000000000 = 000000000000

Blocker - 0000 true000000000000

Disabled - 0000 true000000000000

00 Python 000000000000

000000000000 Python 0000 `cgbrateshttpapi` 0000000000000000
`OCS_Obj.SendData()` 000000


```
# 1GB의 크기
1_GB = 1 * 1024 * 1024 * 1024 # 1073741824 바이트
100_GB = 100 * 1024 * 1024 * 1024

# 1분, 3000분, 3000SMS의 크기
1_minute = 60 * 1000000000 # 60 초
3000_minutes = 3000 * 60 * 1000000000

# 3000SMS의 크기
3000_sms = 3000
```

1 Python

```

Action_AU_Premium_Plan_1 = {
  "id": "0",
  "method": "ApierV1.SetActions",
  "params": [{
    "ActionsId": "Action_au-premium-plan-1",
    "Overwrite": True,
    "Tenant": str(tenant),
    "Actions": [
      # 重置账户
      {
        "Identifier": "*reset_account",
        "Weight": 700
      },
      # 100GB 国内流量
      # 国内流量 PLMN 重置
      # 国内流量 PLMN
      {
        "Identifier": "*topup_reset",
        "BalanceId": "AU_Data_Domestic__" + str(100 * 1024
* 1024 * 1024),
        "BalanceType": "*data",
        "DestinationIDs": "Dest_PLMN_OnNet", # 国内 PLMN
[mcc505.mnc057]
        "Units": 100 * 1024 * 1024 * 1024,
        "ExpiryTime": "*month",
        "BalanceWeight": 1200,
        "Weight": 90
      },
      # 3000 国内语音
      {
        "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
      },
      # 3000 国内语音
      {

```

```

        "Identifier": "*topup_reset",
        "BalanceId": "AU_SMS_Domestic__" + str(3000),
        "BalanceType": "*sms",
        "DestinationIDs": "Dest_AU_Mobile;",
        "Units": 3000,
        "ExpiryTime": "*month",
        "BalanceWeight": 1200,
        "Weight": 88
    },
    # 6GB 套餐
    {
        "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
    },
    # CDR
    {
        "Identifier": "*cdrlog",
        "BalanceType": "*generic",
        "ExtraParameters": "
{\"Category\": \"^activation\", \"Destination\": \"AU Premium Plan
1\"}"
    }
    ]
}

# CGRateS
result = OCS_Obj.SendData(Action_AU_Premium_Plan_1)
assert result['error'] is None or result['error'] == "EXISTS"
print("Action_au-premium-plan-1")

```

测试

- 测试 *reset_account 套餐重置
- 测试 *topup_reset 套餐充值

- BalanceWeight 1200 > 1100
- Weight
- `*cdrlog`

2 Python

20GB

```

Action_AU_Data_Addon_20GB = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_au-data-addon-20gb",
        "Overwrite": True,
        "Tenant": str(tenant),
        "Actions": [
            # 重置账户
            {
                "Identifier": "*reset_account",
                "Weight": 700
            },
            # 20GB 国内流量
            # 国内流量 PLMN 重置
            {
                "Identifier": "*topup_reset",
                "BalanceId": "AU_Data_Domestic__" + str(20 * 1024
* 1024 * 1024),
                "BalanceType": "*data",
                "DestinationIDs": "Dest_PLMN_OnNet", # 国内 PLMN
                #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("成功重置Action_au-data-addon-20gb")

```

3 Python

```

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
            },
            # 100 minutes
            {
                "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("Action_au-international-voice-100min")

```

100 00/SMS 100000000000000000000000000000000 PLMN 1000000000000000 0000 1000000000

Python 1000000000000

100000000000

- **ActionsId** - 100000000000000000000000000000000 CRM 00 product_slug 100000000000000000000000000000000
- **Overwrite** - 1000 True100000000000 ID 100000000000000000000000000000000
- **Tenant** - CGRateS 100000

- **Actions** - 000000000000

00000000

- **Identifier** - 000000 *topup 0 *topup_reset 0 *cdrlog 00
- **BalanceId** - 000000000000000000000000
- **BalanceType** - 000000 *data 0 *voice 0 *sms 0 *monetary 0
- **DestinationIDs** - 00000000000000
 - 00 00/SMS 00000000000000
 - 00 "Dest_AU_Mobile" 0 "Dest_International_UK" 0
 - 00 000000 PLMN 000000
 - 00 "Dest_PLMN_OnNet" 0 "Dest_PLMN_US_Verizon" 0
- **Units** - 000000000000000000000000
- **ExpiryTime** - 00000000 *month 0 +720h 0 2024-12-31 00
- **BalanceWeight** - 0000000000 = 000000
- **Weight** - 00000000000000 = 000000
- **Blocker** 0000 - 0000 True 000000000000
- **Disabled** 0000 - 0000 True 0000000000/00

000000

000000000000000000000000

```

- name: 0000000000
  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"
        }]
      }

```

000000000000000000000000

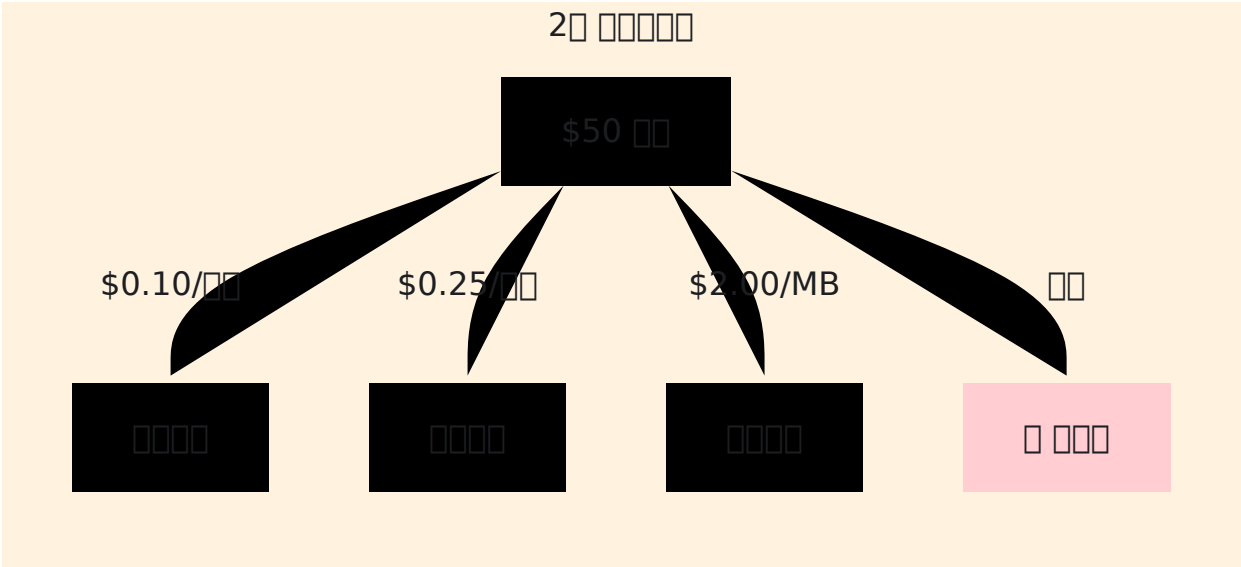
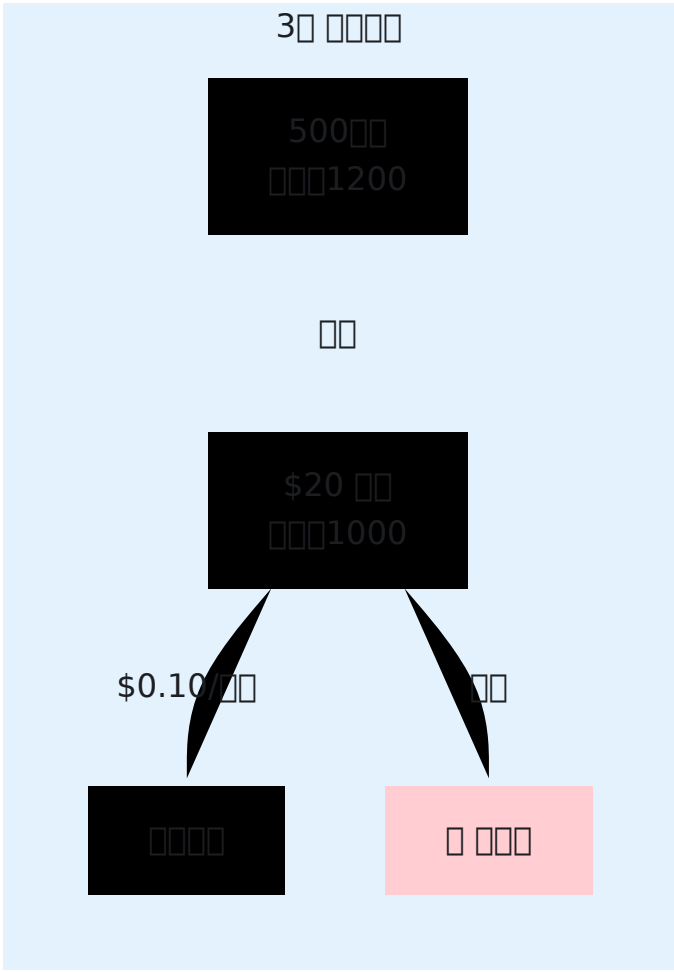
00000000

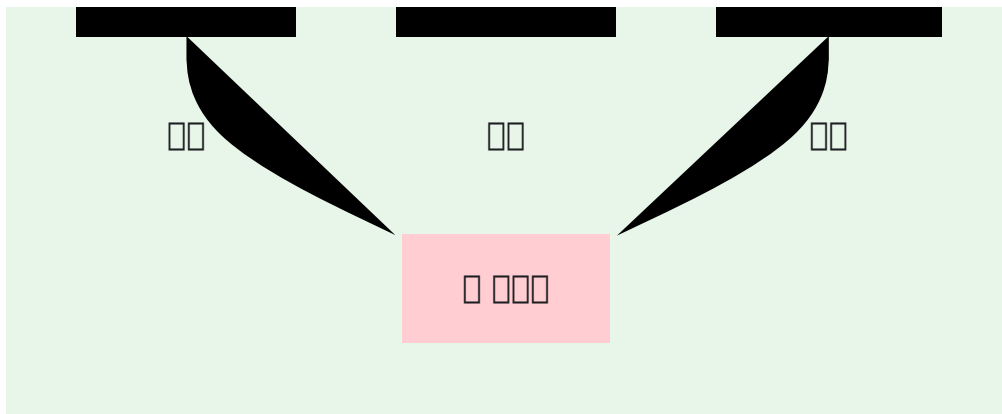
0 CGRateS 000

00000000 **VS** 0000 **VS** 00

000000000000000000000000

項目	項目	項目 PAYG	項目
項目	項目	項目	△ 項目
項目	項目	項目/項目	項目 + 項目
項目	項目	項目	項目
項目	項目	項目	△ 項目
項目	△ 項目	項目	項目
項目	△ 項目 ARPU	項目 ARPU 項目	項目 ARPU
項目	項目	項目	項目
項目	項目	項目	項目





1 GB

1 GB

1 GB

```

Action_Domestic_Plan = {
  "id": "0",
  "method": "ApierV1.SetActions",
  "params": [{
    "ActionsId": "Action_domestic-plan",
    "Overwrite": True,
    "Tenant": tenant,
    "Actions": [
      {
        "Identifier": "*reset_account",
        "Weight": 700
      },
      # 500
      {
        "Identifier": "*topup_reset",
        "BalanceId": f"Domestic_Voice_{500 * 60 *
1000000000}",
        "BalanceType": "*voice",
        "DestinationIDs": "Dest_Domestic_All", #
        "Units": 500 * 60 * 1000000000, # 500
        "ExpiryTime": "*month",
        "BalanceWeight": 1200,
        "Weight": 90
      },
      # 1000
      {
        "Identifier": "*topup_reset",
        "BalanceId": "Domestic_SMS__1000",
        "BalanceType": "*sms",
        "DestinationIDs": "Dest_Domestic_All",
        "Units": 1000,
        "ExpiryTime": "*month",
        "BalanceWeight": 1200,
        "Weight": 89
      },
      # 10GB
      {
        "Identifier": "*topup_reset",
        "BalanceId": f"Domestic_Data_{10 * 1024 * 1024 *
1024}",
        "BalanceType": "*data",
        "DestinationIDs": "Dest_PLMN_OnNet", # PLMN
        "Units": 10 * 1024 * 1024 * 1024,

```

```

        "ExpiryTime": "*month",
        "BalanceWeight": 1200,
        "Weight": 88
    },
    {
        "Identifier": "*cdrlog",
        "BalanceType": "*generic",
        "ExtraParameters": "
{\\"Category\\":\\"^activation\\",\\"Destination\\":\\"Domestic Plan\\"}",
        "Weight": 80
    }
    ]
}]]
}

```

```

result = OCS_Obj.SendData(Action_Domestic_Plan)
assert result['error'] is None or result['error'] == "EXISTS"

```

□□□□

- □□□□1-555-1234 → □□ 500 □□□□
- □□□□44-20-xxx → □□□□□□□□□□□□□□□□□□□□
- □□□□ → □□ 1000 □□□□
- □□□□□□ → □□ 10GB □□
- □□□□ → □□□□□□□□□□□□□□□□□□□□

□□□

- □□□□□□□□
- □□□□□□
- □□□□□□

□□□

- □□□□□□ - □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□

2 PAYG

CGRateS

PAYG PAYG/

\$50 PAYG

```
Action_PAYG_Credit = {
  "id": "0",
  "method": "ApierV1.SetActions",
  "params": [{
    "ActionsId": "Action_payg-50-credit",
    "Overwrite": True,
    "Tenant": tenant,
    "Actions": [
      # $50
      {
        "Identifier": "*topup",
        "BalanceId": "PAYG_Monetary_Balance",
        "BalanceType": "*monetary",
        "DestinationIDs": "*any", #
        "Units": 5000, # $50.00
        "ExpiryTime": "+2160h", # 90
        "BalanceWeight": 1000, #
        "Weight": 90
      },
      {
        "Identifier": "*cdrlog",
        "BalanceType": "*generic",
        "ExtraParameters": "
        {\"Category\": \"^activation\", \"Destination\": \"$50 PAYG
        Credit\"}",
        "Weight": 80
      }
    ]
  }
}]

result = OCS_Obj.SendData(Action_PAYG_Credit)
```

PAYG

□□ **1**□□□□□□□□**10**□□

- □□□\$0.10/□□
- □□□10 × \$0.10 = \$1.00
- □□□\$49.00









□□ **2**□□□□□□□□□□**5**□□

- □□□\$0.25/□□ + \$0.05 □□□
- □□□□5 × \$0.25□ + \$0.05 = \$1.30
- □□□\$47.70

□□ **3**□□ **Verizon** □□□□□□ **100MB** □□

- □□

CGRateS







CGRateS - CGRateS  OmniCRM   -  
  PLMN 



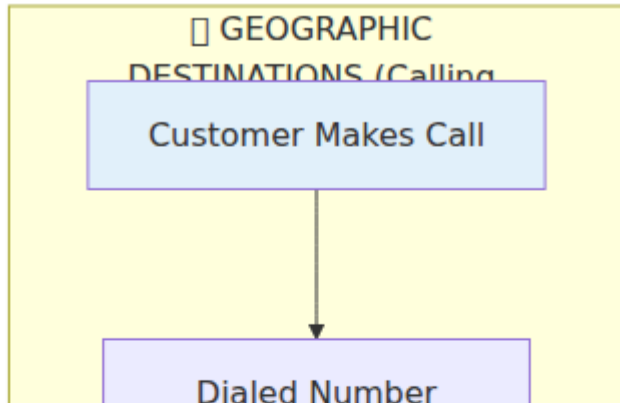
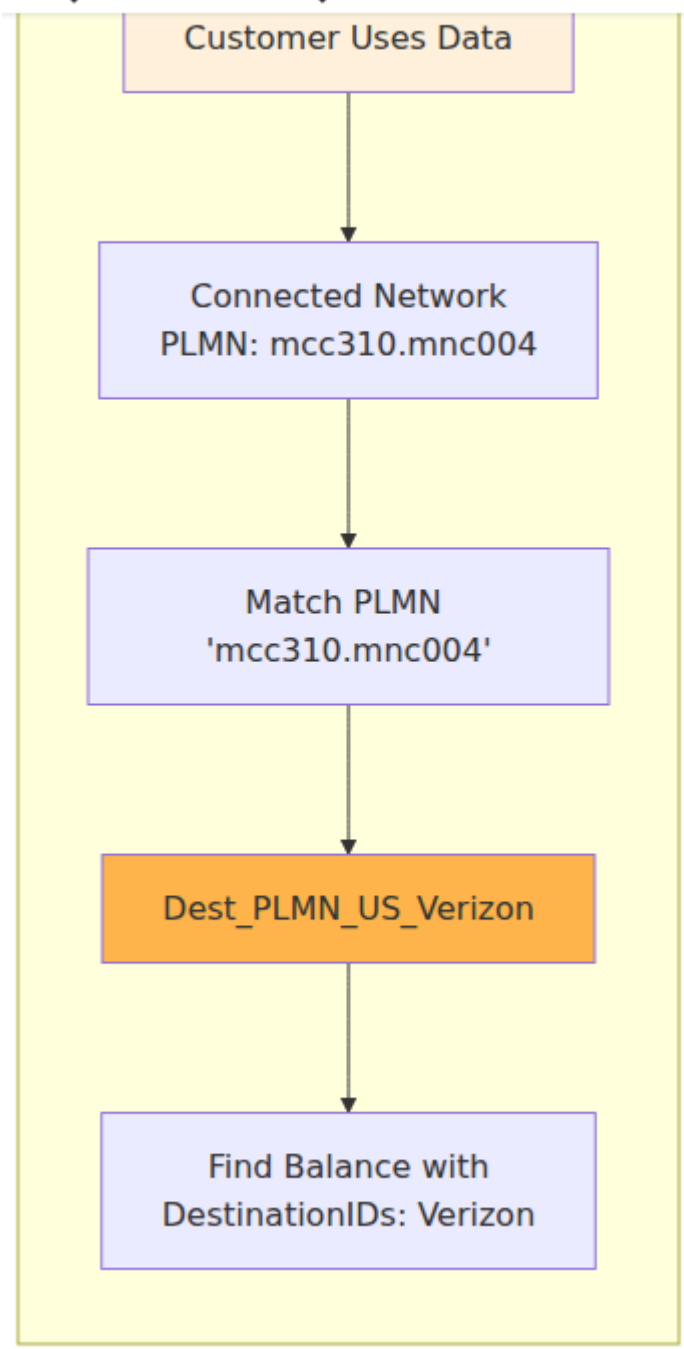
CGRateS     

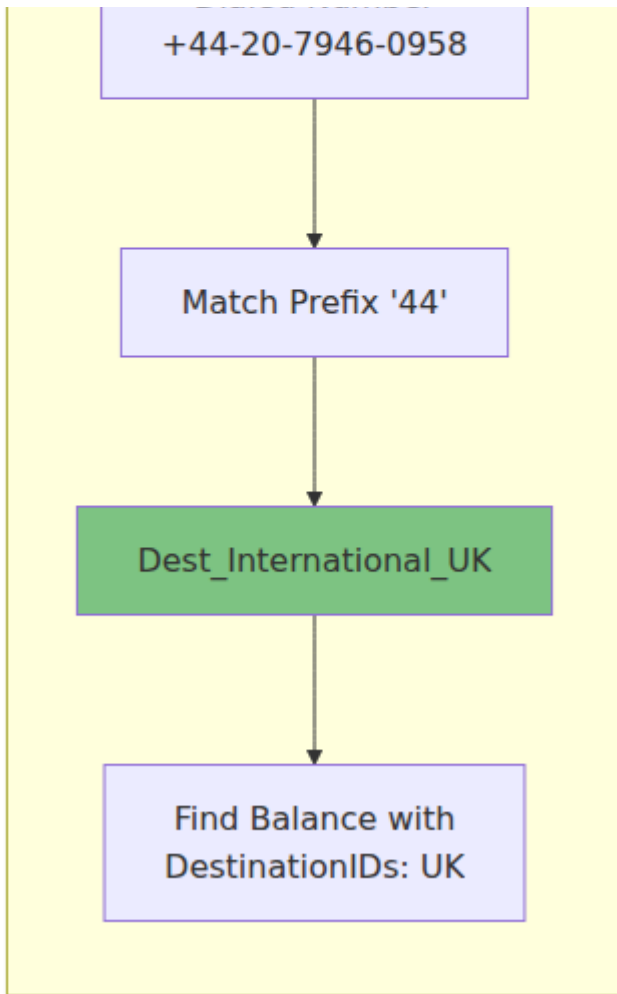
-  -  /SMS 
- **PLMN**  -  Verizon  Vodafone UK 

 CGRateS 

-  /**SMS**  → 
-  PLMN  → PLMN 

OmniCharge OmniRAN Downloads





□□□□

- □□/SMS □□ → □□□□□□□□□□□□□□
- □□□□ → □□ PLMN □□□□□□□□□□□□

□□□□

```

import cgrateshttpapi
import time

OCS_obj = cgrateshttpapi.CGRateS("ocs.example.com", "2080")
tenant = "your_tenant_name"
tpid = str(tenant) + "_" + str(int(time.time()))
  
```

□ **1** □□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□ SMS □□□□□□

□□□□□

```
#
=====
# DOMESTIC DESTINATIONS (US/Canada - NANP)
#
=====

# US/Canada Mobile & Fixed (share area codes)
OCS_Obj.SendData({
    'method': 'A pierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_Domestic_All",
        "Prefixes": ["1"] # NANP (US/Canada)
    }]
})

# US Toll-Free
OCS_Obj.SendData({
    'method': 'A pierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_Domestic_TollFree",
        "Prefixes": ["1800", "1888", "1877", "1866", "1855",
"1844", "1833"]
    }]
})

# US Premium Rate
OCS_Obj.SendData({
    'method': 'A pierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_Domestic_Premium",
        "Prefixes": ["1900"]
    }]
})
```

□□□□□

```
#
=====
# INTERNATIONAL DESTINATIONS (By Country)
#
=====

# United Kingdom
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_International_UK",
        "Prefixes": ["44"]
    }]
})

# China
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_International_China",
        "Prefixes": ["86"]
    }]
})

# Australia - Mobile
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_AU_Mobile",
        "Prefixes": ["614"] # Australian mobiles (04xx dialed as
614xx)
    }]
})

# Australia - Fixed Line
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
```

```

        "TPid": tpid,
        "ID": "Dest_AU_Fixed",
        "Prefixes": [
            "612", # NSW (Sydney)
            "613", # VIC (Melbourne)
            "617", # TAS (Hobart)
            "618" # SA, WA, NT (Adelaide, Perth, Darwin)
        ]
    }
}

# Australia - Toll-Free
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_AU_TollFree",
        "Prefixes": [
            "611800", # Toll-free
            "611300" # Local rate
        ]
    }]
})

# Australia - Premium Rate
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_AU_Premium",
        "Prefixes": ["6119"] # Premium rate services
    }]
})

# Europe (grouped)
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_International_Europe",
        "Prefixes": [
            "33", # France
            "49", # Germany
            "39", # Italy
        ]
    }]
})

```


- `mcc310.mnc004` `mcc234.mnc015` `mcc505.mnc057`

`XXXXXXXXXX`

`XXXXXXXXXXXX SIM XXXXXXXXXXXX`

```
# On-Net (Your Home Network)
# This is YOUR operator's PLMN - the network where YOUR customers'
SIMs are active
# All domestic on-net data balances should use this destination

OCS_Obj.SendData({
  'method': 'ApierV2.SetTPDestination',
  'params': [{
    "TPid": tpid,
    "ID": "Dest_PLMN_OnNet",
    "Prefixes": ["mcc505.mnc057"] # Example: MCC 505, MNC 57
    if this is your home network
  }]
})
# Replace mcc505.mnc057 with YOUR actual operator's PLMN code
# All domestic on-net data balances should use the prefixes set
out here
# Otherwise they're considered roaming
```

PLMN

```
#
=====
# US ROAMING PLMN DESTINATIONS
#
=====

# 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"
        ]
    }]
})

# All US PLMNs (catch-all)
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_US_All",
        "Prefixes": ["mcc310", "mcc311", "mcc312", "mcc313",
            "mcc316"]
    }]
})
# Note: This matches ANY network with these MCCs (310, 311, etc.)
# Customer on mcc310.mnc004 (Verizon), mcc310.mnc410 (AT&T),
# mcc311.mnc580 (US Cellular)
# would ALL match this destination
```

PLMN

```
#
=====
# UK ROAMING PLMN DESTINATIONS
#
=====

# Vodafone UK
OCS_Obj.SendData({
  'method': 'ApierV2.SetTPDestination',
  'params': [{
    "TPid": tpid,
    "ID": "Dest_PLMN_UK_Vodafone",
    "Prefixes": ["mcc234.mnc015"]
  }]
})

# EE (Everything Everywhere)
OCS_Obj.SendData({
  'method': 'ApierV2.SetTPDestination',
  'params': [{
    "TPid": tpid,
    "ID": "Dest_PLMN_UK_EE",
    "Prefixes": ["mcc234.mnc030", "mcc234.mnc033"]
  }]
})

# All UK PLMNs
OCS_Obj.SendData({
  'method': 'ApierV2.SetTPDestination',
  'params': [{
    "TPid": tpid,
    "ID": "Dest_PLMN_UK_All",
    "Prefixes": ["mcc234"]
  }]
})
```

PLMN

PLMN "PLMN" "PLMN"

```

#
=====
# ROAMING ZONES (Multi-Country Groups)
#
=====

# Zone 1: North America
OCS_Obj.SendData({
    'method': 'A pierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_Zone_NorthAmerica",
        "Prefixes": [
            "mcc310", "mcc311", "mcc312", "mcc313", "mcc316", #
USA
            "mcc302", "mcc334" # Canada
        ]
    }]
})
# Using MCC-only prefixes provides coverage on ANY network in these
countries
# Great for "North America roaming" packages that work on all
carriers

# Zone 2: Europe
OCS_Obj.SendData({
    'method': 'A pierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_Zone_Europe",
        "Prefixes": [
            "mcc234", # United Kingdom
            "mcc208", # France
            "mcc262", # Germany
            "mcc222", # Italy
            "mcc214", # Spain
            "mcc228" # Switzerland
        ]
    }]
})

# Zone 3: Asia Pacific
OCS_Obj.SendData({

```

```

'method': 'ApierV2.SetTPDestination',
'params': [{
  "TPid": tpid,
  "ID": "Dest_PLMN_Zone_AsiaPacific",
  "Prefixes": [
    "mcc505", # Australia
    "mcc460", # China
    "mcc454", # Hong Kong
    "mcc440", # Japan
    "mcc520", # Thailand
    "mcc525", # Singapore
    "mcc530" # New Zealand
  ]
}]
})

# Catch-all for any roaming (use with caution)
OCS_Obj.SendData({
  'method': 'ApierV2.SetTPDestination',
  'params': [{
    "TPid": tpid,
    "ID": "Dest_Roaming_All",
    "Prefixes": ["mcc"] # Matches ANY PLMN (very broad)
  }]
})

```

□□□

- □□□□□□□□□□ "□□ Verizon □□□ 5GB" → Dest_PLMN_US_Verizon
- □□□□□□□□□□□□□□□□□□□□ "□□□□□□□□□□□□□□ 10GB" → Dest_PLMN_US_All
- □□□□□□□□□□□□□□□□□□□□ "□□□□□□□□□□□□□□□□□□□□ 15GB" → Dest_PLMN_Zone_Europe
- □□□□□□□□□□□□□□□□□□□□ "□□□□□□□□□□□□□□□□□□□□ 1GB" → Dest_Roaming_All

Table 1

Column 1	Column 2	Column 3	Column 4
Row 1 Col 1	Row 1 Col 2	“100 Row 1 Col 3”	Row 1 Col 4
Row 2 Col 1	Row 2 Col 2	“100GB Row 2 Col 3”	Row 2 Col 4
Row 3 Col 1	Row 3 Col 2	“Row 3 Col 3 Verizon Row 3 Col 4 5GB”	Row 3 Col 4
Row 4 Col 1	Row 4 Col 2	“Row 4 Col 3 5GB”	Row 4 Col 4
Row 5 Col 1	Row 5 Col 2	“10GB Row 5 Col 3”	Row 5 Col 4

Table 2

Table 2.1

Table 2.1 CGRateS Table 2.1 ID Table 2.1 PLMN Table 2.1 TarifPlan Table 2.1

Table 2.2

- Table 2.2.1
 - Table 2.2.2
 - △ Table 2.2.3 TarifPlan Table 2.2.3 - Table 2.2.3
- LoadTariffPlanFromStorDb Table 2.2.4

Table 3

Table 3.1

```
# Initial configuration - Only 3 countries
OCS_Obj.SendData({
  'method': 'ApierV2.SetTPDestination',
  'params': [{
    "TPid": tpid,
    "ID": "Dest_PLMN_Zone_Europe",
    "Prefixes": [
      "mcc262", # Germany
      "mcc208", # France
      "mcc222"  # Italy
    ]
  }]
})
```

□□□□□□ □□ Dest_PLMN_Zone_Europe □□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□

```
# Updated configuration - Now 5 countries
OCS_Obj.SendData({
  'method': 'ApierV2.SetTPDestination',
  'params': [{
    "TPid": tpid,
    "ID": "Dest_PLMN_Zone_Europe",
    "Prefixes": [
      "mcc262", # Germany
      "mcc208", # France
      "mcc222", # Italy
      "mcc214", # Spain (NEW)
      "mcc234"  # UK (NEW)
    ]
  }]
})

# Load the tariff plan to make changes active
OCS_Obj.SendData({
  'method': 'ApierV1.LoadTariffPlanFromStorDb',
  'params': [{
    "TPid": tpid,
    "DryRun": False,
    "Validate": True
  }]
})

# Reload the destination cache
OCS_Obj.SendData({
  'method': 'CacheSv1.ReloadCache',
  'params': [{
    "DestinationIDs": ["Dest_PLMN_Zone_Europe"]
  }]
})
```

Dest_PLMN_Zone_Europe -

2

???


```
# Add Canada to "Domestic" calling
OCS_Obj.SendData({
  'method': 'ApierV2.SetTPDestination',
  'params': [{
    "TPid": tpid,
    "ID": "Dest_Domestic_All",
    "Prefixes": [
      "1", # Already includes USA (and previously Canada)
      # No change needed - "1" prefix already covers both
    ]
  }]
})

# Or create separate North America destination
OCS_Obj.SendData({
  'method': 'ApierV2.SetTPDestination',
  'params': [{
    "TPid": tpid,
    "ID": "Dest_North_America",
    "Prefixes": [
      "1" # USA + Canada
    ]
  }]
})
```

DestinationIDs: "Dest_Domestic_All" "1"

/SMS/

1

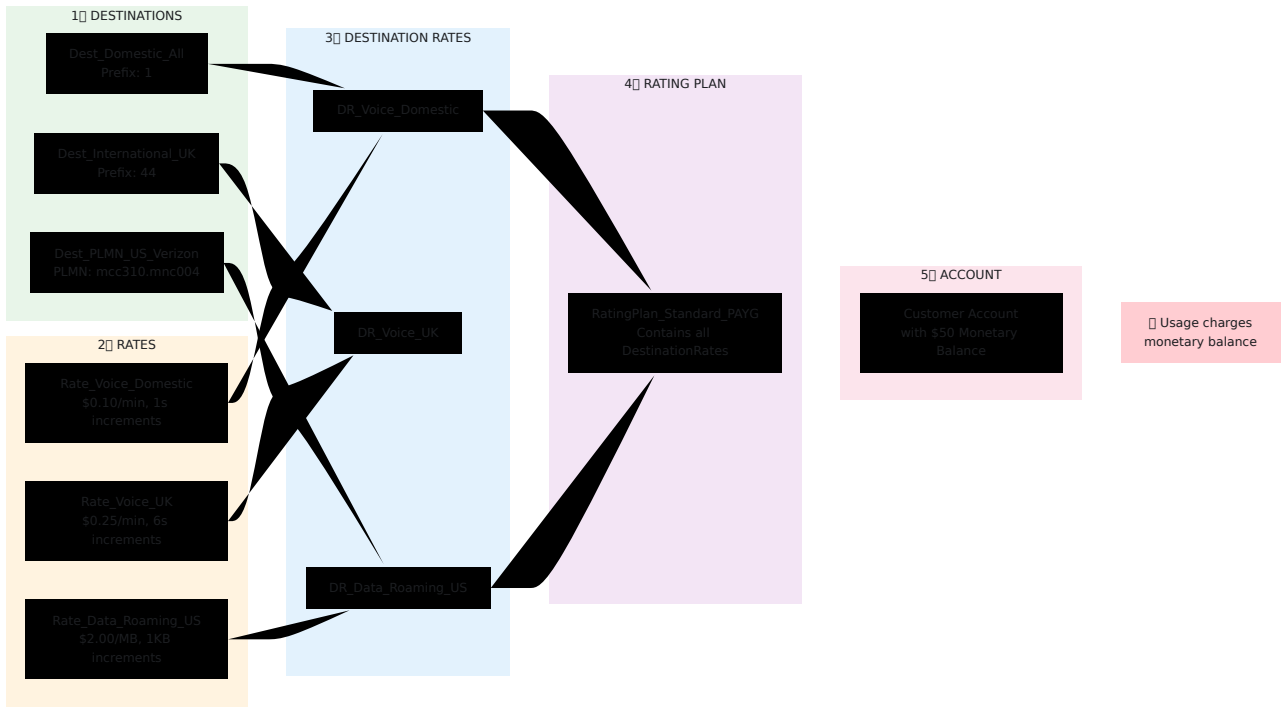
DestinationIDs: "*any" DestinationIDs: "*any"

3

```
# Customer has:  
# Balance 1: UK minutes  
{  
  "BalanceId": "UK_Voice_100min",  
  "BalanceType": "*voice",  
  "DestinationIDs": "Dest_International_UK",  
  "Units": 100 * 60 * 1000000000,  
  "BalanceWeight": 1200  
}  
  
# Balance 2: Monetary with wildcard  
{  
  "BalanceId": "PAYG_Credit",  
  "BalanceType": "*monetary",  
  "DestinationIDs": "*any", # Matches ANYTHING  
  "Units": 5000, # $50  
  "BalanceWeight": 1000 # Lower priority  
}  
  
# Customer calls Australia (+61)
```

-
- "61"
- PAYG "*any"
- PAYG

- PLMN
- /
-



XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXSMS XXXXXXXX

```

#
=====
# VOICE RATES (Per Minute)
#
=====

# Domestic voice: $0.10/minute
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPRate',
    'params': [{
        "TPid": tpid,
        "ID": "Rate_Voice_Domestic",
        "RateSlots": [{
            "ConnectFee": 0,
            "Rate": 0.10,
            "RateUnit": "60s",          # 1 minute
            "RateIncrement": "1s",     # Per-second billing
            "GroupIntervalStart": "0s"
        }
    ]
}]
})

# UK voice: $0.25/minute
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPRate',
    'params': [{
        "TPid": tpid,
        "ID": "Rate_Voice_UK",
        "RateSlots": [{
            "ConnectFee": 0.05,        # $0.05 connection fee
            "Rate": 0.25,
            "RateUnit": "60s",
            "RateIncrement": "6s",     # 6-second blocks
            "GroupIntervalStart": "0s"
        }
    ]
}]
})

#
=====
# SMS RATES (Per Message)
#
=====

```

```

# Domestic SMS: $0.05/message
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",    # Per message
            "GroupIntervalStart": "0s"
        }
    ]
}]
})

#
=====
# DATA RATES (Per MB) - For Roaming
#
=====

# Roaming on US Premium Networks (Verizon, AT&T): $2.00/MB
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPRate',
    'params': [{
        "TPid": tpid,
        "ID": "Rate_Data_Roaming_US_Premium",
        "RateSlots": [{
            "ConnectFee": 0,
            "Rate": 2.00,
            "RateUnit": "1048576",  # 1 MB in bytes
            "RateIncrement": "1024", # Per KB
            "GroupIntervalStart": "0s"
        }
    ]
}]
})

```

□□□□□□

- **ConnectFee** - □□□□□□□□□□
- **Rate** - □□ RateUnit □□□

- **RateUnit** - 60s 1MB 1
- **RateIncrement** -
- **GroupIntervalStart** -

DestinationRates

```

# Link Domestic destination to Domestic voice rate
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
    }]
  }]
})

# Link UK destination to UK voice rate
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
    }]
  }]
})

# Link Verizon PLMN to roaming data rate
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
    }]
  }]
})

```

```
}]
})
```

□□□□□□

□□□ DestinationRates □□□□□□□□□□

```
OCS_Obj.SendData({
  'method': 'ApierV2.SetTPRatingPlan',
  'params': [{
    "TPid": tpid,
    "ID": "RatingPlan_Standard_PAYG",
    "RatingPlanBindings": [
      # Voice rates (higher weight = more specific)
      {"DestinationRatesId": "DR_Voice_UK", "TimingId":
"*any", "Weight": 40},
      {"DestinationRatesId": "DR_Voice_Domestic",
"TimingId": "*any", "Weight": 20},

      # SMS rates
      {"DestinationRatesId": "DR_SMS_Domestic", "TimingId":
"*any", "Weight": 20},

      # Data roaming rates
      {"DestinationRatesId": "DR_Data_Roaming_US_Premium",
"TimingId": "*any", "Weight": 50}
    ]
  }]
})
```

□□□□□□

□□□□□□□□□□ CGRateS □□

```

# Load tariff plan
OCS_Obj.SendData({
  'method': 'ApierV1.LoadTariffPlanFromStorDb',
  'params': [{
    "TPid": tpid,
    "DryRun": False,
    "Validate": True
  }]
})

# Reload cache
OCS_Obj.SendData({
  'method': 'CacheSv1.ReloadCache',
  'params': [{
    "Tenant": tenant,
    "DestinationIDs": ["*all"]
  }]
})

```

□□□□

- □□□□ - □□□□□□□□□□
- CGRateS □□□□□□ - □□□□□□□□□□

□□□□

1. □□□□□□□□

```

# Good - specific home network
"DestinationIDs": "Dest_PLMN_OnNet", # mcc505.mnc057

# Bad - too broad, matches competitors
"DestinationIDs": "Dest_PLMN_AU_All", # mcc505 (all AU operators)

```

2. □□□□□□□□□□

```
# Good - self-documenting
Dest_PLMN_US_Verizon
Dest_International_UK
Dest_PLMN_Zone_Europe

# Bad - unclear
Dest_Data_1
Dest_Voice_Package
```

3. □□□□□□□□□□

```
# Good - logical grouping
Dest_Domestic_All
Dest_Domestic_TollFree
Dest_Domestic_Premium

# Bad - too granular
Dest_Area_Code_212
Dest_Area_Code_213
Dest_Area_Code_214
```

4. □□ **PLMN** □□

□□ PLMN □□□□□□□□□□ MCC/MNC □□□□□□

```

# Verizon Wireless
# Source: https://www.mcc-mnc.com/ (verified 2024-12)
OCS_Obj.SendData({
  'method': 'ApierV2.SetTPDestination',
  'params': [{
    "TPid": tpid,
    "ID": "Dest_PLMN_US_Verizon",
    "Prefixes": [
      "mcc310.mnc004", # Verizon - Nationwide
      "mcc310.mnc010", # Verizon - East
      ...
    ]
  }]
})

```

□□□□

□□□□□□□□□□□□

□□□ □□□□□□□□□□□□□□□□

□□□

```

# 1. □□□□□□□□□□
OCS_Obj.SendData({
  'method': 'ApierV2.GetTPDestination',
  'params': [{
    "TPid": tpid,
    "ID": "Dest_PLMN_US_Verizon"
  }]
})

# 2. □□□□ DestinationIDs □□
# □□□□□□□□□□ "*any"

# 3. □□ PLMN□□□□□□□□□□□□□□□□
# □□ Diameter CCR □□□□ Visited-PLMN-Id AVP

```

PLMN OnNet

PLMN OnNet

PLMN OnNet `Dest_PLMN_OnNet`

PLMN

```
# PLMN Diameter CCR
# PLMN

OCS_Obj.SendData({
  'method': 'ApierV2.GetTPDestination',
  'params': [{
    "TPid": tpid,
    "ID": "Dest_PLMN_OnNet"
  }]
})
# PLMN
# ["mcc505.mnc057"] SIM
```

PLMN OnNet

PLMN OnNet

PLMN

- PLMN OnNet
 - +44-20-xxxx → "44"
- DestinationIDs**
 - "DestinationIDs": "Dest_International_UK" | "Dest_International_All"
- ExpiryTime**
- Units > 0**

PLMN

- CGRateS** - PLMN

Ansible Playbooks:

OmniCRM **Ansible**

 SIM Card Provisioning <concepts_sim_provisioning> Ansible
 SIM eSIM

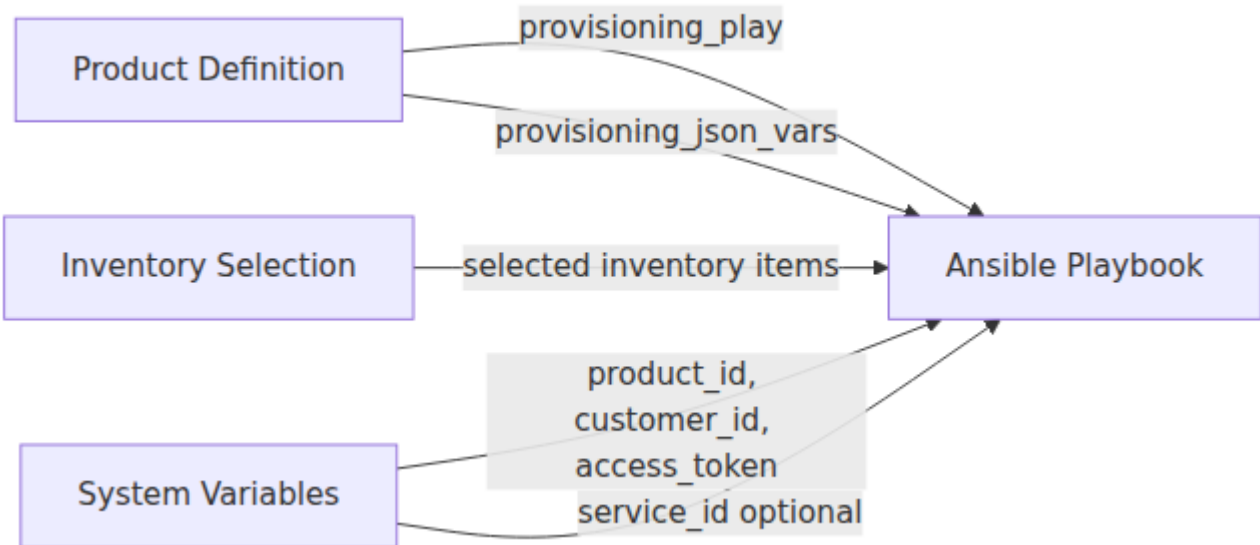
Playbooks

 Playbooks OmniCRM playbook
 -

Playbooks

 OmniCRM

- playbook provisioning_play
- playbook provisioning_json_vars
- playbook
- playbook



Playbooks 目錄

目錄 playbook 目錄

目錄

目錄 playbook 目錄

- 目錄
- IPTV 目錄
- VoIP 目錄
- 目錄

目錄

目錄 playbooks 目錄

- 目錄 CPE 目錄 playbook
- 目錄 playbook
- 目錄 playbook

目錄

目錄

- 目錄
- 目錄
- 目錄

目錄

目錄 playbooks

- 目錄
- 目錄
- 目錄

目錄

目錄 playbooks 目錄

- 目錄
- 目錄

- `provisioning_play`

OmniCRM Playbook

```
# 1. SIM provisioning 1
# provisioning_play: play_simple_service
- CRM
- OCS
- SIM
-

# 2. Bundle provisioning 3
# provisioning_play: play_bundle_internet_tv_voice
-
- IPTV
- VoIP
-
-

# 3. Topup no charge 0
# provisioning_play: play_topup_no_charge
- service_id
- OCS
-
-

# 4. CPE provisioning 0
# provisioning_play: play_prov_cpe_mikrotik
-
-
-
-
```

OmniCRM `playbook` tasks

Plays Tasks

OmniCRM Plays Tasks tasks OmniCRM playbooks

Playbook

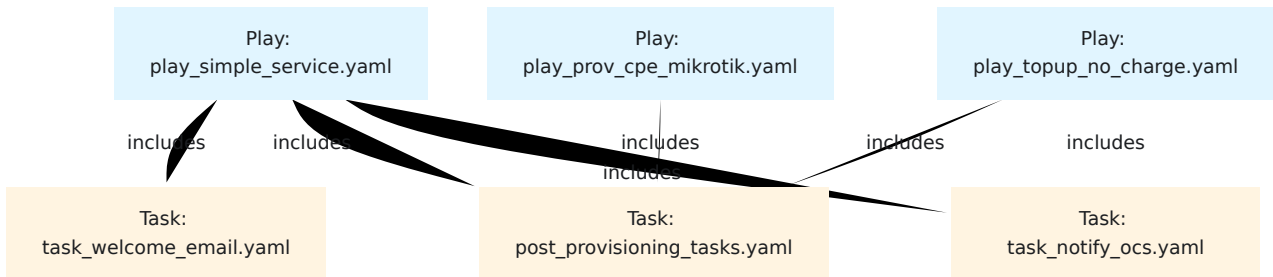
Plays `OmniCRM-API/Provisioners/plays/` playbooks

- `play_simple_service.yaml` -
- `play_topup_no_charge.yaml` -
- `play_prov_cpe_mikrotik.yaml` -

Task

plays Tasks `task_`

- `task_welcome_email.yaml` -
- `task_activate_olt.yaml` - OLT
- `task_notify_ocs.yaml` -



```

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

        - name: Configure billing
          uri: ...

    # Include reusable task
    - include_tasks: task_welcome_email.yml

    # Include post-provisioning tasks
    - include_tasks: post_provisioning_tasks.yml

```

Playbook `tasks`

OmniCRM playbooks `tasks` playbooks `tasks`

`tasks`

playbook `tasks`

```

- name: Descriptive Name of the Playbook
  hosts: localhost           # localhost
  gather_facts: no          # no
  become: False             # False

  tasks:
    - name: Main block
      block:
        # tasks

      rescue:
        # tasks

```

□□□□

name

□□□□□ UI □□□□□□□□□□□□□□□□□□□□ □ `playbook_description` □

hosts: localhost

□□ OmniCRM playbooks □□ localhost □□□□□□□□□□ API □□□□□□□□□□□□□□ SSH□

gather_facts: no

Ansible □□□□□□□□□□□□

- □□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□

become: False

□□□□□□□□ API □□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

□□ playbook □□□□□□□□□□□□

```
tasks:  
  - name: Include vars of crm_config  
    ansible.builtin.include_vars:  
      file: "../..../crm_config.yaml"  
      name: crm_config
```

□□□□□□□□□□ `crm_config.ocs.cgrates` □ `crm_config.crm.base_url` □□

`crm_config.yaml` □□□□□□

```
ocs:  
  cgrates: "10.0.1.100:2080"  
  ocsTenant: "default_tenant"  
crm:  
  base_url: "https://crm.example.com"
```

□□□□□□

□□□□□□□□□□

□□□□□□

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

□□□□□□

```
- 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]"
```

□□ **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 }}"
```

📖 **Playbook** 📖

📄📄📄📄📄

📄📄📄📄📄📄📄📄📄📄

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

tasks:
  - name: Main block
    block:

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

      # 2. ████████
      - 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. ████████
      - 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. ████████████████
      - 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. 生成UUID
```

- name: Generate UUID
set_fact:
 uuid: "{{ 99999999 | random | to_uuid }}"
- name: Generate Service UUID
set_fact:
 service_uuid: "Service_{{ uuid[0:8] }}"

```
# 6. 创建OCS/CGRateS
```

- 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. 余额增加
- 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. CRM 当前日期
- 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.
- 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.
- include_tasks: post_provisioning_tasks.yml

```

```
rescue:

# 00/0000
- 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"
```



```

- 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. サービス情報取得
    - 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. サービス情報取得
    - 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. サービス情報取得
    - 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. 测试添加信用的操作
- 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. 测试重置操作
- 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. 测试计算到期日期
- 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. 送信メッセージ
- 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

```

CPE 送信

送信メッセージ送信メッセージONT

```

- 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. 設定 CPE の ID
  - 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. API から CPE のデータを取得
  - 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. API から Site の情報を取得
  - 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. CPE の位置情報を更新
  - 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. 更新 CPE

```
- 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. 创建 WiFi 网络

```
- 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. 配置文件
- 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. 读取配置文件
- name: Read config file
  ansible.builtin.slurp:
    src: "{{ config_dest }}"
  register: config_content

# 9. 打补丁
- 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. ██████████
- 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

```

████████

██ CGRateS ActionPlans ██████████

```

# 创建 playbook 文件

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

# 2. 创建 Action 文件
- 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. 创建 ActionPlan 文件
- 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. `ActionPlan` `配置`

- 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. 定期删除 ActionPlan
- 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
```

部署

部署 playbook 到 playbooks 目录

部署

`task_welcome_email.yaml` - 部署

```

# 000000000000 play 000
# - api_response_customer (000000)
# - package_name (0000)
# - monthly_cost (0000)
# - setup_cost (00000)

- 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
```

□□□□□

post_provisioning_tasks.yaml - □□□□□□□□□□□□□□□□

```
# □□□□□□□□□□ playbooks □□□□□
# □□□□□□□□□□□□

- include_tasks: task_notify_ocs.yaml
```

task_notify_ocs.yaml □□□□□

```
- 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
```

□□□□

□□□□□

□□□□□□□□

```

- 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 }}"

```

□□□□□□□□□□

```

- 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

```

日期时间

获取日期/时间

- **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') }}"`

计算日期

- **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`

生成UUID

UUID 生成

- **name:** Generate UUID
set_fact:
uuid: `"{{ 99999999 | random | to_uuid }}"`
- **name:** Generate service identifier
set_fact:
service_uuid: `"SVC_{{ uuid[0:8] }}"`

生成服务ID

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

□□□□□□□□

```
- 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 }}"
```

□ CGRateS/OCS □□

□□□□□

```
- 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
```

□□□□□

```
- 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
```

□□□□□

```
- 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
```

□□□□□□

```
- 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
```

□□□□□□□□□□

```

- 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

```

□□□□

□□□□□□□□

```

- 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

```

□□□□

- `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`

□□□□

- `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`

□□□□□

□□□□

```

- 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 }

```

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 }}"

```

ignore_errors

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

□□□□□□□□

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

□□□□□□□□

```
- 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
```

□□□□

□□□□

□□□□□□□□□□

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

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

□□□□□□□□

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

□□

□□□□□□□□□□

- 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

□□

□□□□□□ API □□□

```
- 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 }}"
```

□□□

□□□□□□□□□□□□□□

```
# □□□□□□□□□□
- name: Check if account exists
  uri:
    url: "{{ ocs_endpoint }}/get_account"
    method: POST
    body: {"Account": "{{ service_uuid }}" }
  register: account_check
  failed_when: false

# □□□□□□□□□□
- name: Create account
  uri:
    url: "{{ ocs_endpoint }}/create_account"
    method: POST
    body: {...}
  when: account_check.status == 404
```

□□□

□□□□□□□□□□

```
# □  
mailjet_api_key: "abc123def456"  
  
# □ - □□□□□□  
mailjet_api_key: "{{ lookup('env', 'MAILJET_API_KEY') }}"  
  
# □ - □□□□□□  
mailjet_api_key: "{{ crm_config.email.api_key }}"
```

□□□□ HTTPS □□□□□□

```
- name: Call external API  
  uri:  
    url: "https://api.example.com/endpoint"  
    method: POST  
    headers:  
      Authorization: "Bearer {{ access_token }}"  
  validate_certs: yes
```

□□

□□□□□□□□

```

# 0000000000000000
# 00000 15 000000000 1 00
# 0000000000 50% 000
- 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) }}"

```

00 Playbooks

0000

1. 000000000000000000000000
2. 00000000000000000000000000000000
3. 0000000000000000 API 00
4. 000000000000000000000000000000
5. 0000000000 `action: "deprovision"` 00000000

0000 playbook

```

- 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

```

□□□□

□□□□□□

```

# □□ - □□□□□□
customer_id: "{{ customer_id }}"

# □□ - □□□□□□
customer_id: {{ customer_id | int }}

```

□□□□□□□□□□

```

# task - create_account
service_uuid: "{{ service_uuid }}"

# task - register
service_uuid: "{{ service_uuid | default('') }}"

```

task

```

# task - create_account
- name: Create account
  uri: ...
  register: response

# task - verify
- name: Create account
  uri: ...
  register: response

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

```

task

task Omnitouch task

1. task
2. task Ansible playbooks task
3. task playbooks
4. task

task

Ansible 簡介

Ansible playbooks 是什麼

簡介

OmniCRM 安裝與配置

簡介

安裝與配置 SIM 卡 IP 地址

簡介

OmniCRM 安裝

- `product_id` - 產品 ID
- `customer_id` - 客戶 ID
- `service_id` - 服務 ID / 類型
- `access_token` - API 訪問的 JWT

安裝

Ansible Playbooks 安裝 `rescue` 處理錯誤

- 安裝 Ansible
- 配置 Ansible
- 安裝 Playbooks
- 執行 Playbooks

配置

配置 Ansible

1. 安裝 Ansible

安裝 Ansible 的步驟

2. 配置 Ansible

在 playbook 中配置 `action: "deprovision"` 操作

□□□□□

□□ OmniCRM playbooks □□□□□□□□□□

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

tasks:
  - name: Main block
    block:
      # □□□□□□□□□□
      - name: Create OCS account
        uri: ...

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

      - name: Add transaction
        uri: ...

rescue:
  # □□□/□□□□□□□□□□
  - name: Print all vars for debugging
    debug:
      var: hostvars[inventory_hostname]

  # □□□□□□□□□□□□□□
  - name: Remove account in OCS
    uri: ...

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

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

□□□□□□

□□ play_simple_service.yaml □□□□□

```

rescue:
  # 1. []
  - 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. []
  - 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. [] service_uuid
  - 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. [] 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. CRM サービス削除
- 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. 测试手动取消部署
- 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"
```

测试部署

ignore_errors: True

测试部署 ignore_errors: True 测试部署失败时继续执行后续任务

测试

测试 when 测试部署失败时继续执行后续任务

```
when: service_creation_response is defined
```

测试

测试部署失败时继续执行后续任务

1. 测试部署失败时继续执行后续任务
2. 测试部署
3. 测试 OCS/测试部署
4. 测试部署失败时继续执行后续任务

测试

测试部署

- 测试部署 (action == "deprovision") → 测试
- 测试部署失败时继续执行后续任务 → 测试

测试部署

OCS/CGRateS 测试

```
- 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
```

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
```

□□□□□□□□□□

```

- 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

```

ActionPlans

```

- 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

```

□□□□□

```

- 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

```

□□□□□□□□

□□□□□□□□

1. □□□□□□□□
2. □□□□□□ playbook□□□ action: "deprovision"
3. Playbook □□□□□□□□
4. □□□□□□□□
5. □□□□□□□□

□□ **API** □□□

```

POST /crm/provision/run_playbook
{
  "product_id": 123,
  "customer_id": 456,
  "service_id": 789,
  "action": "deprovision"
}

```

□□□□□□

□□ 1 OCS □□□□□□□□□□

- OCS □□□□
- □□□□□□
- □□□□□ OCS □□
- □□□□□□□□□□□□

□□ 2 □□□□□□□□□□

- OCS □□□□
- □□□□□□
- □□□□□
- □□□□□ OCS □□□□□
- □□□□□□□□□□□□

□□ 3 □□□□□□□□□□

- OCS □□□□
- □□□□□□
- □□□□□□□□□□
- □□□□
 - □□□□□□□□□□□□
 - □□□□□□
 - □□ OCS □□

□□□□□□□

1. □□□□ **ignore_errors**

□□□□□ - □□□□□□□□□□□□□□

2. □□ **when** □□□□□□□□□□

□□□□□□□□□□□□□□

```
when: service_creation_response is defined
```

3. 環境構築

環境構築の準備

```
- name: Print all vars for debugging
  debug:
    var: hostvars[inventory_hostname]
```

4. 環境構築

環境構築の準備

- 環境構築
- OCS のインストール
- 環境構築

5. 環境構築

環境構築の準備

6. 環境構築

OCS のインストール `ReloadScheduler: true` CGRateS のインストール

環境構築

環境構築の準備

- 環境構築
- 環境構築
- 環境構築
- 環境構築
- 環境構築

環境構築の準備

- CRM のインストール
- OCS のインストール
- 環境構築
- 環境構築
- 環境構築

Ansible

Ansible `block/rescue` blocks allow you to group tasks into blocks and handle errors gracefully. This is useful for ensuring that your playbook can recover from failures and continue to execute other tasks.

Example

```
- 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:
    # 1. Log the error
    # 2. Clean up
    # 3. Playbook continues
    - name: Cleanup partial changes
      uri: ...
```

For more information, see the `concepts_provisioning` page.

OmniCRM API

OmniCRM API - UI

OmniCRM

API RESTful API JWT API IP

API Swagger API

API URL

<<https://yourcrm/crm/docs/>>



OmniCRM

- 1. **JWT Bearer** - Web UI

2. **API** -

3. **IP** -

JWT Bearer

JWT

- Web UI
-
-

JSON POST `/crm/auth/login`

```
{  
  "email": "youruser@yourdomain.com",  
  "password": "yourpassword"  
}
```

API `token` JSON

`refresh_token`

Swagger `/auth/login` `Try it out`

Swagger Authorize Value
Bearer Authorize

API 32

API 32

API 32

API

- API
- API
- API
- API cron

API 32

API 32 `crm_config.yaml` API 32 32
API 32 `X-API-KEY` API 32

API 32

API 32 `crm_config.yaml` API 32

```
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"
```

API

X-API-KEY API

```
curl -X GET "https://yourcrm.com/crm/customers" \
  -H "X-API-KEY: your-secure-api-key-here-minimum-32-chars"
```

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
}

#
response = requests.get(crm_url + '/crm/customers',
headers=headers)
for customer in response.json()['data']:
    print(customer)
```

- API (openssl rand -base64 48)

- `description` API
- `description` API
- API
- API
- API

IP

IP API

-
- VLAN
- Ansible

IP

IP `crm_config.yaml`

```
ip_whitelist:
  - 192.168.1.100
  - 10.0.0.0/24
  - 172.16.50.10
```

- IP
- IP
- IP
- IP
- IP

Python API

JWT

```

import requests

crm_url = 'https://yourcrm.com'
session = requests.Session()

print("Provisioning data to server: " + str(crm_url))

headers = {
    "Content-Type": "application/json"
}

# 🚀
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")

# 📊
response = session.get(crm_url + '/crm/customers',
headers=headers)
for customer in response.json()['data']:
    print(customer)

```

API 📊

API 📊 API 📊
[Monitoring & Metrics](#)

📊 Prometheus 📊 `/crm/metrics` 📊

□□□□

- **System Architecture** - □□□□□□□□□□
- **Monitoring & Metrics** - □□□□□□□□□□ API□□□□□□□□□□
- **Authentication Flows** - □□□□□□□□□□
- **RBAC** - □□□□□□□□□□

□□□□□□□□□□

□□□□□□□□□□□□□□□□□□——□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

Introduction

Introduction to the course

OmniCRM Introduction

Introduction to the course

Introduction to the course "Introduction" "IP" Introduction to the course

Introduction

Introduction to the course

OmniCRM Introduction "Introduction" Introduction to the course

Introduction to the course

Introduction

Introduction to the course

Introduction to the course

- Introduction to the course CRM Introduction
- Introduction to the course Introduction to the course
- Introduction to the course SIM Introduction to the course SIM Introduction

['SIM Card', 'Phone Number'] CRM Introduction <administration_inventory>

- Ansible Playbook Play <concepts_ansible> Ansible service_id ICCID

MSISDN
UI

MSISDN

UI

UI

category UI

- **standalone** - 獨立安裝包
- **addon** - 附加安裝包
- **bundle** - 集成安裝包
- **promo** - 促銷包

Ansible playbook 安裝包

standalone 獨立安裝包 - addon 附加安裝包 bundle 集成安裝包 - 安裝包
 playbook 安裝包 - 安裝包

UI 安裝包/安裝包

安裝包

service_type 安裝包

安裝包

- **mobile** - 移動安裝包
- **fixed** - 固定安裝包
- **fixed-voice** - 固定安裝包VoIP安裝包
- **hotspot** - 熱點安裝包
- **dongle** - USB 安裝包
- **voice** - 語音安裝包
- **data** - 數據安裝包

安裝包

- 安裝包
- 安裝包
- 安裝包
- 安裝包

mobile 安裝包 fixed 安裝包

3. ⚙️ 📁 - 📁📁📁📁📁📁📁📁
4. 📁 📁 - Ansible playbook 📁📁📁📁
5. 📁 📁📁 - 📁📁📁📁📁📁

📁📁📁📁📁

📁📁📁📁📁📁📁📁📁📁

- 📁📁📁 - 📁📁📁📁📁📁📁
- 📁📁 - 📁📁📁📁📁📁📁
- 📁 - 📁📁📁📁📁

📁📁📁📁

- 📁📁📁 - 📁📁📁📁📁📁 FontAwesome 📁
- 📁📁📁📁 - 📁📁📁📁📁📁📁
- 📁/📁📁📁 - 📁📁📁📁📁
- 📁📁📁 - 📁📁📁📁📁 \$ 📁
- 📁📁📁 - 📁📁📁📁📁📁📁📁

Font Awesome

Font Awesome 4.7.0 - The complete set of 600 icons in a single file, ready to use. Includes the Font Awesome CSS files, the Font Awesome icons, and the Font Awesome logo.

Font Awesome 4.7.0 - The complete set of 600 icons in a single file, ready to use. Includes the Font Awesome CSS files, the Font Awesome icons, and the Font Awesome logo.

Font Awesome

Font Awesome 4.7.0 - The complete set of 600 icons in a single file, ready to use. Includes the Font Awesome CSS files, the Font Awesome icons, and the Font Awesome logo.

Ansible

- Ansible
 - Ansible - 使用 Python 实现
 - Ansible - 使用 YAML 语法
 - Ansible - 使用 SSH 协议
- Ansible
 - Ansible - 使用 Playbook
 - Ansible - 使用 Roles
- Ansible
 - Ansible - 使用 Inventory
 - Ansible - 使用 Connection Plugins


Ansible

Ansible 是 IT 运维领域的利器

□□□□□

- □□□□□
 - □□□□□□□□□□/□□
 - □□□□□□□□
 - □□□□□□□□□□
- □□□□□
 - □□□□□□□□□□□□/□□
 - □□□□□□□□□□
 - □□□□□□□□□□□□
- □□□□□□□□□□
 - □□ - □□□□□□□□□□□□
 - □□□□ - □□□□□□□□□□
 - □□□□□□□□

□□□□□

- □□□□□
 -  - □□□□
 - □□□□ - □□□□□□□□“_clone”□□□□□□
 - □□□□ - □□□□□□□□
- □□/□□□□□
 - □□ - □□□□□□□□
 - □□□□ - □□□□□□□□□□□□□□□□□□□

쿠폰

쿠폰은 할인율을 지정하고, 할인 대상 금액을 지정할 수 있다.

- 할인율 10%인 쿠폰을 \$50.00 → 할인액 \$5.00
- 할인율 0%인 쿠폰 → 할인액 \$0.00
- 할인 대상 금액 → 할인 대상 금액이 없으면

쿠폰 조건

- **enabled** - 쿠폰이 활성화된 상태인지 `false` 인 경우
- **residential** - 주거용 고객에게만 적용되는 쿠폰
- **business** - 상업용 고객에게만 적용되는 쿠폰
- **customer_can_purchase** - 고객이 쿠폰을 구매할 수 있는지 `true` 또는 `false`
- **available_from** - 쿠폰이 유효한 시작 날짜/시간
- **available_until** - 쿠폰이 유효한 종료 날짜/시간

쿠폰 설정

- **contract_days** - 쿠폰이 유효한 기간 30 일 또는 365 일 또는 0 일
- **auto_renew** - 자동 갱신
 - `prompt` - 고객에게 알림
 - `true` - 자동 갱신
 - `false` - 수동 갱신

- **allow_auto_renew** - 是否允许自动续费 false

其他属性

- **terms** - 服务条款
- **features_list** - 功能列表 Python 列表 ['Feature 1', 'Feature 2']

其他属性

- **provisioning_play** - 关联的 Ansible playbook 名称 .yaml 文件路径 `OmniCRM-API/Provisioners/plays/`
- **provisioning_json_vars** - 关联的 JSON 文件 Ansible playbook 需要的变量 `customer_id` `product_id` `service_id` `access_token`
- **inventory_items_list** - 关联的库存项列表 ['SIM Card', 'Mobile Number'] 每个项包含 ID 和名称
- **relies_on_list** - 依赖的 ID 列表

其他属性

- **created** - 创建时间
- **last_modified** - 最后修改时间

□□□□□□

□□□□□□ **SIM**□

```
{
  "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"
}
```

XXXXXXXXXXXXXXXXXXXXSIM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXX

```
{
  "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"
}
```

XX/XXXX

□□□□□□□□□□

```
{
  "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"
}
```

□□□□□□□□□□ playbook □□□□□□□□□□ + IPTV + □□□□□□□□□□□□□□□□□□□□□□□□□□

4. ID inventory_items_list

```
"provisioning_json_vars": "{\"monthly_cost\": 50, \"data_gb\": 100}"
```

API

```
{
  "product_id": 10,
  "customer_id": 456,
  "monthly_cost": 45,
  "custom_param": "value"
}
```

Ansible extra_vars

```
{
  "monthly_cost": 45,           #
  "data_gb": 100,             # provisioning_json_vars
  "product_id": 10,           #
  "customer_id": 456,         #
  "custom_param": "value",    #
  "access_token": "eyJ...",   #
}
```

- access_token - API CRM JWT IP/API
- refresh_token

- `initiating_user` - 启动用户 ID
- 产品 ID `product_id` 客户 ID `customer_id` 服务 ID `service_id`

示例

```
inventory_items_list: ["SIM Card", "Mobile Number"]
```

1. **UI/API** 交互 - 启动用户 ID
2. 产品 ID - 产品 ID
3. **Playbook** 产品 ID - 产品 ID 通过 CRM API 交互

示例 - SIM 产品 ID: 789 - 客户 ID: 101

示例 playbook 交互 - `SIM Card: 789 - Mobile Number: 101`

Playbook 交互 ID 通过 CRM API 交互 `ICCID` `IMSI` `MSISDN` 交互

Ansible 交互

交互 `extravars` 通过 Ansible playbook 交互

```
ansible-playbook --extra-vars '{"product_id": 789, "customer_id": 101, "monthly_cost": 100}'
```

交互

交互

- `product_id` - 产品 ID
- `customer_id` - 客户 ID
- `service_id` - 服务 ID
- `access_token` - API 令牌
- `provisioning_json_vars` 交互
- API 交互

API Parameters

Example: "£5" service_id: 123 playbook

- product_id: 45
- customer_id: 456
- service_id: 123
- access_token:
- provisioning_json_vars

Playbook

- CRM API service_id
- UUID
- OCS UUID
- CRM

API Parameters

- product_id -
- customer_id -
- ID SIM Card Mobile Number
- access_token - API

- product_id -
- customer_id -
- service_id - ID
- access_token - API

service_id - playbook

Playbook

playbook 需知

- product_id - 产品ID
- customer_id - 客户ID
- service_id - 服务ID
- Modem Router ID 需知
- access_token - API 鉴权

playbook 需知 play_seniors_package IPTV 需知

需知

需知

OCS 需知 OCS CGRateS 需知 ActionPlans ThresholdS 需知

需知

需知

需知

UI 需知

- 需知 - 需知
 - residential = true 需知 business = false 需知
 - business = true 需知
- 需知 - 需知
 - category = standalone 需知 bundle 需知
 - category = addon 需知
- 需知 - 需知

- `enabled = true` - 可購
- 開始日期 `available_from` 及 結束日期 `available_until` 日期 - 可購
- `customer_can_purchase = true` 可購 - 可購

API

API 查詢API 查詢

- 查詢 (/crm/product/) - 查詢
- 查詢 (/crm/product/paginated) - 查詢

查詢 `include_disabled=true` 查詢

UI 查詢

- 查詢 - 查詢
- 查詢 - 查詢
- 查詢 - 查詢
- 查詢 - 查詢

查詢

查詢 查詢

1. 查詢 - 查詢 `service_type` 查詢
 - 查詢 `service_type = "mobile"` 查詢 `service_type = "mobile"` 查詢
 - 查詢
2. 查詢 - 查詢 `relies_on_list`
 - 查詢/查詢
 - 查詢
3. 查詢 - 查詢 `residential` 及 `business` 查詢

查詢

查詢 `service_type = "mobile"` 查詢

- 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇/business = true category != "addon"
- 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇/business = true category = "addon" service_type = "mobile"
- 〇〇

〇〇〇〇

〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇

〇〇〇〇〇〇

- **service_id** - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- **customer_id** - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- **product_id** - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- **service_name** - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- **service_type** - 〇〇〇〇〇〇〇〇〇〇〇〇〇VoIP〇IPTV〇〇〇〇〇〇〇〇〇〇
- **service_uuid** - 〇〇 OCS/CGRateS 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- **icon** - 〇〇〇〇〇〇〇〇〇〇〇〇 FontAwesome 〇〇〇〇〇〇〇〇〇〇

〇〇〇〇〇〇〇

- **service_status** - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- **service_provisioned_date** - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- **service_active_date** - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- **service_deactivate_date** - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- **contract_end_date** - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇

〇〇〇〇〇

- **retail_cost** - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- **wholesale_cost** - 〇〇〇〇〇〇〇〇〇〇〇〇
- **service_billed** - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇true
- **service_taxable** - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇true
- **invoiced** - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- **promo_code** - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇

Attributes

- **service_visible_to_customer** - `boolean` (default: true)
- **service_usage_visible_to_customer** - `boolean` (default: true)

Playbooks

- **provisioning_play** - Ansible playbook
- **provisioning_json_vars** - Ansible playbook variables
- **deprovisioning_play** - Ansible playbook
- **deprovisioning_json_vars** - Ansible playbook variables

Fields

- **bundled_parent** - `string` (service_id)
- **site_id** - `string` (required)

Metadata

- **service_notes** - `string`
- **created** - `datetime`
- **last_modified** - `datetime`

API/UI

API/UI

Endpoint: `PATCH /crm/service/{service_id}`

- 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

Product and Service

Product and Service Data

- service_id, customer_id, product_id
- service_uuid
- service_provisioned_date
- provisioning_play, provisioning_json_vars
- deprovisioning_play, deprovisioning_json_vars
- bundled_parent
- invoiced
- created, last_modified

API and Ansible

API and Ansible playbooks are used for provisioning and deprovisioning services. Web UI, API, and Ansible playbooks.

API and Ansible

1. API - Ansible playbooks
2. API - "relies_on_list"
3. API - available_from/available_until, enabled, customer_can_purchase
4. API - inventory_items_list
5. API - SIM
6. API - API

API and Ansible

API and Ansible

Step 1: Overview

API Endpoints

- `provisioning_play` - Ansible playbook (parameter: `play_psim_only`)
- `provisioning_json_vars` - Provisioning JSON variables
- `customer_id` - Customer ID
- `product_id` - Product ID
- `service_id` - Service ID
- `access_token` - Access Token ID

Step 2: Configuration

Configuration Parameters

1. Parameter: `provisioning_json_vars`
2. Parameter: `access_token`
3. Parameters:
 - `access_token` - API CRM JWT
 - `initiating_user` - Initiating User ID
 - `customer_id`, `product_id`, `service_id`
4. Parameter: `{inventory_type: inventory_id}`

Example JSON

```
{
  "customer_id": 123,
  "product_id": 456,
  "service_id": 789,           # Service ID
  "SIM Card": 1001,          # SIM Card ID
  "Mobile Number": 1002,     # Mobile Number
  "monthly_cost": 30,        # From provisioning_json_vars
  "data_gb": 50,             # From provisioning_json_vars
  "access_token": "eyJ...",  # API Access Token
  "initiating_user": 5       # Initiating User ID
}
```

Step 3: Execution

Provisioner Provisioner

- `provision_id` - Provisioner ID
- `provisioning_play` - playbook name
- `provisioning_json_vars` - Provisioner JSON vars
- `task_count` - playbook task count in YAML
- `provisioning_status` - Provisioner status 1 = Success 0 = Failure 2 = Pending 3 = Error
- `product_id` `customer_id` `service_id` - Provisioner metadata

4. Ansible playbook

API Provisioner

1. `OmniCRM-API/Provisioners/plays/{playbook_name}.yaml` Ansible playbook YAML
2. `ansible_runner.run()` parameters
 - `playbook` - Ansible playbook name
 - `extravars` - Provisioner Ansible vars
 - `inventory` - `'localhost,'` inventory
 - `event_handler` - Provisioner event handler
3. Provisioner playbook

5. ProvisioningEventHandler

Ansible Provisioner Provision_Event

- `event_name` - Ansible playbook name
- `event_number` - Ansible event number
- `provisioning_status` - Provisioner status
 - **0** = Success - Success
 - **1** = Failure - Failure
 - **2** = Pending - Pending
 - **3** = Error - Error (with `ignore_errors: true` in playbook)
- `provisioning_result_json` - Provisioner result JSON

Provisioner Provisioner

06 Ansible playbook Provisioners/plays/*.yaml

Ansible playbook

1. `access_token` - `/crm/product/product_id/{{ product_id }}` GET
2. `/crm/customer/customer_id/{{ customer_id }}` GET
3. `/crm/inventory/inventory_id/{{ inventory_id }}` GET
ICCID, MSISDN
4. API
 - HSS
 - IMS/IP
 - CGRateS/OCS
 - ENUM
 -
5. `/crm/transaction/` POST
6. OCS/CGRateS POST `retail_setup_cost`
7. **OCS** POST UUID
8. OCS/CGRateS Actions, ActionPlans
9. `/crm/service/` PUT/POST 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. `PATCH /crm/inventory/inventory_id/{{ inventory_id }}` PATCH
 "status": "Active"

11. `Provision.provisioning_status` 0

7

playbook

- `Provision.provisioning_status` 0
- `Provision.provisioning_status` 2
`crm_config.provisioning.failure_list`
- `ignore_errors: true` 3

CRM

`category: standalone`

- `customer_id` `product_id`
- SIM
- API PUT `/crm/service/`
- HSS OCS
- SIM

category: addon

- customer_id product_id service_id
-
- OCS
- OCS
-
-

category: bundle

-
-
- + +
-
- CRM
- + IPTV + VoIP

playbook

playbook

1. OmniCRM-API/Provisioners/plays/{playbook_name}.yaml
2. Ansible extravars {{ variable_name }}
3. Authorization: Bearer {{ access_token }} API
4. rescue ignore_errors
- 5.
- 6.
7. fail

Playbooks

playbook

- customer_id -
- product_id -

- `service_id` - 服務 ID/產品 ID
- `access_token` - CRM API 的 JWT 令牌
- `initiating_user` - 發起用戶
- 產品 ID `{{ inventory_type }}: inventory_id`
- 提供 `provisioning_json_vars` 參數
- 其他參數

Playbooks 查詢

- 查詢 `GET /crm/product/product_id/{{ product_id }}`
- 查詢 `GET /crm/customer/customer_id/{{ customer_id }}`
- 查詢 `GET /crm/inventory/inventory_id/{{ SIM_Card }}`
- 查詢 `POST /crm/transaction/`
- 查詢 `PUT /crm/service/`
- 查詢 `PATCH /crm/service/{{ service_id }}`
- 查詢 `PATCH /crm/inventory/inventory_id/{{ inventory_id }}`

playbook 查詢

查詢

1. Playbook 查詢 `customer_id` `product_id` `service_id` `access_token`
2. 查詢 `GET /crm/service/{{ service_id }}` 獲取 `service_uuid`
3. 查詢 `GET /crm/product/product_id/{{ product_id }}` 獲取
4. OCS 查詢 `POST to CGRateS` 獲取 `retail_cost`
5. OCS 查詢 `POST to CGRateS` 獲取
6. CRM 查詢 `POST /crm/transaction/` 獲取
7. 查詢

查詢 `Provision` 查詢 `Provision_Event` 查詢

OCS 查詢

OCS 查詢 `CGRateS` 查詢 CRM 查詢 OCS 查詢

- **Phone Number** - Phone Number DID
- **Phone Number** - Phone Number MB
- **Phone Number** - Phone Number GB
- **Phone Number** - Phone Number \$30 10GB
- **Phone Number** - Phone Number

CRM OCS

OCS/CGRateS

UI API

1. **API** - GET /crm/service/{service_id} UI

2. **API** - service_uuid

3. **CGRateS API** - CGRateS

i. **Get_Balance(service_uuid)** - BalanceMap

-
- ID ID
- custom_Name_hr custom_Expiration custom_Description_String

ii. **Get_ActionPlans(service_uuid)** -

4. **CGRateS**

```

{
  "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. UI -

- **ServiceUsage.js** - 3
- **UsageCard.js** -
- **UsageProgress.js** - /
-

BalanceMap

CGRateS

- **ID** - "DATA_10GB_2025_01"
- **Value** -
 - 5368709120 = 5 GB
 - 3600 = 1 小時
 - 100 = 100 分鐘
 - 25.50 = \$25.50
- **ExpirationDate** - ISO 8601 格式
- **Weight** - 權重
- **DestinationIDs** - ["AU", "INTERNATIONAL"]

CRM

- **custom_Name_hr** - ID
- **custom_Expiration** - "2025 年 1 月 15 日 11 時"
- **custom_Description_String** -
 - "5 GB 10 GB"
 - "60 分鐘"
 - "50 分鐘"
 - "\$25.50"

- **service_visible_to_customer** - false
- **service_usage_visible_to_customer** - false

- /
-
-

API Spec

Web UI

- 查詢 3 個月 ServiceUsage
- 查詢 GET /crm/service/{service_id} 的 CGRateS
- 查詢

查詢

API Spec / Action

查詢 DID 的 OCS Actions
Action_ServiceUUID_ServiceName_WhatitDoes

\$60 GPON 1TB Action

Action_kj49-adsf-1234-9742_60_GPON_1TB_MonthlyExpiry

1. \$0
2. CRM HTTP POST
3. 1TB 1

MRC ActionPlan_{service_uuid}_Monthly_Charge ActionPlan ActionPlan

MRC 12

Actions ActionPlans

查詢

API Spec

Action 查詢

[[ActionPlans]] [[Ansible]] Action [[ExecuteAction]]

[[]]

[[\$10 PAYG]]

[[\$0]]

[[/]]

[[ThresholdS]]

[[PAYG / \$0]]

[[CRM]] [[OCS]]

[[CRM API]] [[OCS]] [[Balances]] [[ActionPlans]] [[Actions]]

[[CRM API]] [[ActionPlans]] [[Ansible Playbooks]] [[CRM]]

[[OCS]] [[ActionPlans]]

[[ThresholdS]]

[[CRM]] [[ActionPlans]]

[[ActionPlans]] [[CRM]]

[[ActionPlans]]

[[CRM]] [[ActionPlans]]

1. API - GET /crm/service/{service_id} API

-
- `service_uuid` OCS
- CGRateS API UUID ActionPlans
- `Get_ActionPlans(service_uuid)` CGRateS

2. ActionPlan - ActionPlan

```
{
  "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` -
- `PlanName` - playbook
- `NextExecTime` - ActionPlan ISO
- `custom_NextExecTime_hr` - "11 days" "2025 2 1"
- `ActionPlanId_split_dict` - ActionPlanId

3. UI - ActionPlans ActionPlansTable

- ActionPlanId ProductID
- retail_cost
- custom_NextExecTime_hr
-

- `DELETE /crm/oam/remove_action_plan/{action_plan_id}` API
- `ocs.Remove_ActionPlan()` API

API ActionPlans

- `DELETE /crm/oam/remove_action_plan/{action_plan_id}`
- `ocs.Remove_ActionPlan()`

API ActionPlans

API ActionPlans

API ActionPlan

1. API ActionPlansTable `DELETE /crm/oam/remove_action_plan/{action_plan_id}`
2. `ocs.Remove_ActionPlan()` API
3. `DELETE /crm/oam/remove_action_plan/{action_plan_id}`
4. API `ocs.Remove_ActionPlan()` CGRateS ActionPlan
5. `{service_id}` ActionPlan `{ActionPlanId}`
6. ActionPlan

API ActionPlan

1. API ActionPlansTable `DELETE /crm/oam/remove_action_plan/{action_plan_id}`
2. `ocs.Remove_ActionPlan()` API
3. `ocs.Remove_ActionPlan()`
 - `ocs.Remove_ActionPlan()` `product_id`
 - `ocs.Remove_ActionPlan()`
 - `ocs.Remove_ActionPlan()` `playbook`
 - `ocs.Remove_ActionPlan()`
4. `ocs.Remove_ActionPlan()`
5. `ocs.Remove_ActionPlan()`

API ActionPlan

API `auto_renew` ActionPlans

- `auto_renew = "true"` ActionPlans

- `auto_renew = "prompt"` 选项 - 自动续订
- `auto_renew = "false"` 选项 - 不续订 ActionPlans

在 playbook 中 CGRateS 选项 ActionPlan

- 选项 ID 选项 ActionPlanId
- 选项
- 选项
- 选项

ActionPlan 选项

ActionPlans 选项

选项

```
ServiceID_{service_uuid}__ProductID_{product_id}__CustomerID_{customer_id}
```

选项

```
ServiceID_abc-123-def__ProductID_456__CustomerID_789__MonthlyRenewal
```

选项 CRM

- 选项 ActionPlan 选项
- 选项
- 选项
- 选项

CRM 选项 `ActionPlanId_split_dict` 选项

选项 ActionPlans

在 CRM 选项 ActionPlans 选项

ServiceView.js

- ActionPlans
-
-

-
-
-

- ActionPlans“”
-

ActionPlans

CRM

GET /crm/service/expiring?threshold=7

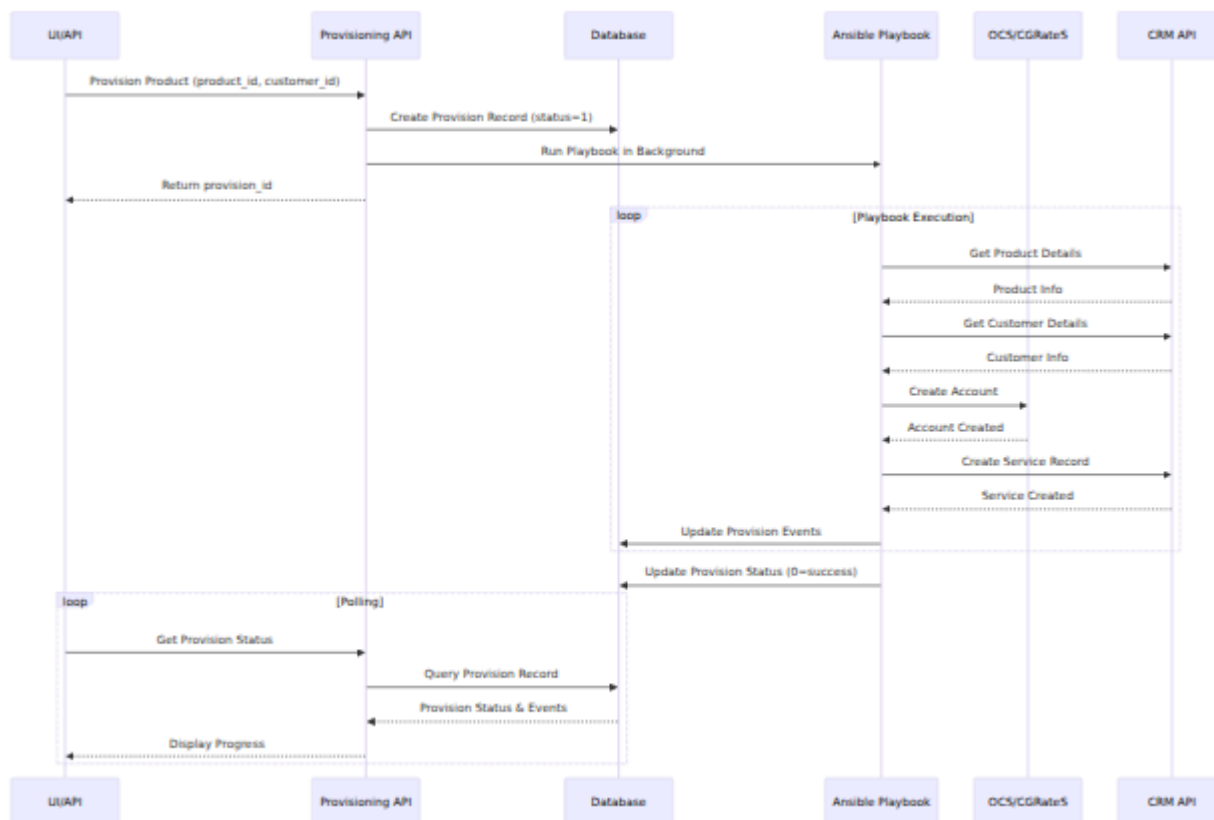
-
- ActionPlans
- ActionPlans

-
-
-
-



OmniCRM **Ansible**

SIM <concepts_sim_provisioning> SIM eSIM



- `provisioning_json_vars` - Ansible JSON
- `inventory_items_list` - `['SIM Card', 'Mobile Number']`
- `product_id` `product_name` -

```
{
  "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
}
```

1. `provisioning_play`

`OmniCRM-API/Provisioners/plays/play_psim_only.yaml`

2. `extra_vars`

- i. `provisioning_json_vars` `{"iccid": "", "msisdn": ""}`

- ii. /API

- iii. `product_id` `customer_id`

- iv. `access_token` `refresh_token`

3. `inventory_items_list`

UI/API

- **SIM** ID - SIM Card ID
- **Mobile Number** - Mobile Number

Inventory ID `extra_vars` provisioning

```
extra_vars = {
  "product_id": 1,
  "customer_id": 456,
  "SIM Card": 789,           # SIM Card inventory_id
  "Mobile Number": 101,     # Mobile Number inventory_id
  "iccid": "",              # provisioning_json_vars
  "msisdn": "",             # provisioning_json_vars
  "access_token": "eyJ..."
}
```

4. `hostvars[inventory_hostname]` Ansible

Inventory

```
- 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]"
```

Inventory

Inventory ID API provisioning

```

- 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 }}"

```

□□□□□□□□□□□□□□□□

- □ HSS □□□ SIM □□ IMSI
- □□□□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□□□

□□□□□□□□□□ **SIM** □□

□ `play_psim_only.yaml` □□□□□□□□□□□□□□□□□□□□

- 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
```

□□□□□□□□□□

1. □□□□□□ `provisioning_play: "play_psim_only"`
2. □□□□ `inventory_items_list: ['SIM Card', 'Mobile Number']`
3. □□□□□□□□□□□□□□□□
4. □□ ID □□ `extra_vars` □□□□□□
5. □□□ API □□□□□□□□□□□□
6. □□□□□□□□□□□□□□□□
7. □□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□ Ansible □ `block` □ `rescue` □□□□□□□□□□□□□□□□□□□□□□

□□□□□

□□ `play_psim_only.yaml` □

```

- name: OmniCore Service Provisioning 2024
  hosts: localhost
  gather_facts: no
  become: False

tasks:
  - name: Main block
    block:
      # ---  [] [] [] [] ---
      - name: Get Product information
        uri: ...

      - name: Create account in OCS
        uri: ...

      - name: Provision subscriber on HSS
        uri: ...

      - name: Create service record
        uri: ...

      # ...  [] [] [] [] [] [] [] [] ...

rescue:
  # ---  [] [] [] [] ---
  # [] [] [] [] [] [] [] [] [] [] [] [] [] []
  # 1. [] [] [] [] [] [] [] [] [] [] [] [] [] []
  # 2. action == "deprovision" [] [] [] [] []

  - name: Get Inventory items linkedd 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:] }}", # []
        "ue_ambr_dl": 9999999, # []
        "ue_ambr_ul": 9999999
      }
    loop: "{{ hss_subscriber_data | dict2items }}"
    when:
      - deprovision_subscriber | default(false) | bool ==
false

# []
- name: Set status to "Success" if Manual deprovision /
Fail if failed provision
  assert:
    that:
      - action == "deprovision"

```

[][][][][][][][]

1. []

[][][][][][][][]

- [] [] block [] rescue []
- [] action: "deprovision" [] rescue

2. []

rescue

- OCS
-
-
- HSS
-

“”

-
-
-
-

3. ignore_errors

ignore_errors: True

-
-
-

“ OCS ”

- OCS
-
-
- HSS

4.

rescue

```

- name: Set status to "Success" if Manual deprovision / Fail if
  failed provision
  assert:
    that:
      - action == "deprovision"

```

□□□□□

- □□ `action == "deprovision"` □□□□□□□□□□□□□□ 0□
- □□ `action` □□□□ `!= "deprovision"` □□□□□□□□□□□□□□ 2□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

5. □□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```

- 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

```

□□□□□□□□□□

- □□□□□□ SIM □□□□□□□□ `deprovision_subscriber: true`□
- □□□□□□ SIM □□□□□□□□ HSS □□□□ `deprovision_subscriber: false`□

□□□□□□□□

□□□□□□

```
{
  "product_id": 1,
  "customer_id": 456,
  "provisioning_play": "play_psim_only"
}
```

rescue

```
{
  "service_id": 123,
  "service_uuid": "Service_abc123",
  "action": "deprovision",
  "provisioning_play": "play_psim_only"
}
```

rescue

action: "deprovision"

provisioning_json_vars

```
{
  "provisioning_json_vars": "{\"monthly_cost\": 50,
  \"data_limit_gb\": 100}"
}
```

```
{
  "product_id": 1,
  "customer_id": 456,
  "monthly_cost": 45,
  "data_limit_gb": 150
}
```

extra_vars

-
-
-

```
{
  "product_id": 10,
  "product_name": "Extra 10GB Data",
  "provisioning_play": "play_local_data_addon",
  "provisioning_json_vars": "{\"data_gb\": 10}",
  "inventory_items_list": "[]"
}
```

```
extra_vars = {
  "product_id": 10,
  "customer_id": 456,
  "service_id": 123, #
  "data_gb": 10,
  "access_token": "eyJ..."
}
```

API 호출

API 호출 방법

- UI 호출
- REST API 호출
- API 호출 `PUT /crm/provision/`

UI 호출

UI 호출 방법

1. UI 호출

UI 호출 방법

- `provisioning_play` - Ansible 플레이북
- `inventory_items_list` - 인벤토리 항목 목록 ['SIM Card', 'Mobile Number']
- `provisioning_json_vars` - JSON 변수

2. 인벤토리 항목 목록

`inventory_items_list` 호출 시 인벤토리 항목 목록을 반환합니다.

3. JSON 변수

JavaScript 호출 `PUT /crm/provision/` 호출

```
PUT /crm/provision/
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "product_id": 42,
  "customer_id": 123,
  "SIM Card": 5001,
  "Mobile Number": 5002
}
```

4. API 設定

設定

- Bearer API IP 設定
- CREATE_PROVISION 設定
- initiating_user
- 設定
- OmniCRM-API/Provisioners/plays/play_psim_only.yaml

5. 設定

設定

```
# 設定
product_vars = json.loads(product['provisioning_json_vars'])
# 設定
request_vars = request_body_json
# 設定
system_vars = {
    'product_id': 42,
    'customer_id': 123,
    'access_token': auth_token, # 設定
    'initiating_user': 7
}
# 設定
extra_vars = {**product_vars, **request_vars, **system_vars}
```

6. 設定

設定 1

```
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, # OK
    'task_count': 85,
    'initiating_user': 7,
    'created': '2025-01-10T14:30:00Z'
}
```

7. 테스트

테스트할 데이터

- play_psim_only.yaml
- extra_vars
- ID 456
- 테스트 데이터

8. API 테스트

UI에서 provision_id

```
{
  "provision_id": 456,
  "provisioning_status": 1,
  "message": "Provisioning job created"
}
```

9. UI 테스트

UI에서 3번 GET /crm/provision/provision_id/456

```

{
  "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. Ansible 部署

部署环境

- 部署环境部署环境 Provision_Event 部署
- 部署环境部署环境0=部署2=部署3=部署部署部署部署 JSON
- UI 部署部署部署部署部署部署部署部署部署部署
- 部署部署部署部署部署部署部署部署部署部署

UI 部署

部署部署部署 1部署部署部署部署

- 部署部署部署 - 部署部署部署部署部署/部署/部署
- 部署 - 部署部署部署部署部署部署
- 部署部署 - 部署部署部署部署部署部署/部署部署部署部署

部署部署

部署部署部署部署部署12 部署部署部署 8 部署

✓ CRM API ✓ ✓ SIM ICCID:
8991101200003204510 ✓ 555-0123 ☞ OCS/CGRateS
... ..

11.

- provisioning_status: 0 -
- provisioning_status: 2 -

UI

-
- X
- provisioning.failure_list

- initiating_user JWT ID
- API ID
- IP ID

- CREATE_PROVISION
- VIEW_OWN_PROVISION

Ansible CRM API

Ansible API CRM
Bearer

`access_token` API

1. Bearer

Web UI

1. `POST /crm/auth/login`
2. JWT `access_token` 15-30 `refresh_token`
3. Bearer
4. API `Authorization: Bearer ...`
- 5.
6. `access_token` Ansible

Authorization Bearer API

2. API X-API-KEY

API

1. `PUT /crm/provision/` `X-API-KEY: your-api-key...`
2. API `crm_config.yaml`
3. JWT
- 4.
5. Ansible

API JWT API JWT

- API
- `admin` JWT
- ID JWT
- API

API JWT

3IP

1. IP 192.168.1.100
2. API IP `crm_config.yaml` `ip_whitelist`
3. JWT
- 4.
5. Ansible

IP JWT

API

```

- 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"

```

access_token 15-30
access_token refresh_token

```
refresh_token = request.cookies.get('refresh_token')  
run_playbook(playbook_path, extra_vars, provision_id,  
refresh_token=refresh_token)
```

access_token

1. 401
2. refresh_token POST /crm/auth/refresh
3. access_token
- 4.

API /IP 1-2

- 1.

- o
- o Ansible
- o Provision
- o
- o API

- 2.

Ansible CRM API

- o **Bearer** JWT refresh_token
- o **API** /IP access_token

- 3.

API

- Provision_Event
-
-
- UI GET /crm/provision/provision_id/<id>

4.

Ansible

- API
- API
- SIM IP
- OCS/OCS
-
- CRM API
-
- /SMS

5.

Ansible block rescue

- rescue
- 3 ignore_errors: true
- YAML

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

```

○○○○○○○○○○○○

1. ○ CRM API ○○○○○○○○
2. ○○○○○○ UUID
3. ○ OCS ○○○○○○
4. ○○ CRM API ○○○○○○
5. ○○○○○○○○
6. ○○○○○○○○ rescue ○○○○ OCS ○○

○○○○○○○○○○

○○○○◆◆◆ ○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○ OCS ○○ ActionPlans ○○○○○○

○○○○○○○

OmniCRM ○○○○○○○○○○○

- **POST**
/crm/provision/simple_provision_addon/service_id/<id>/product_id/<id>

□□□□□□□□□□□□□□□□□□□□□□

- **POST**
/crm/provision/simple_provision_addon_recharge/service_id/<id>/product_id/<id>

□□□□□□□□□□□□□□

□□□□□□□□□□

□□□□□□□□ **IP** □□□ □ **API** □□ □□□□□□□□

- □□□□ IP □□□□□□□ □□ `crm_config.yaml` □□ `ip_whitelist` □□
- □□□□□□□□□ □□ `crm_config.yaml` □□ `api_keys` □ API □□
- □□□□□□□□□□□□□□

□□□ **OCS ActionPlan** □□

OCS □□□□□□□□□□□□□□□□□□□□□□□□

```
{
  "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"
      }
    ]
  }]
}
```

□□□□□□ HTTP POST □□

□□□□□□□□□□□□□□□□ play_topup_no_charge.yaml □□□□□□□□□□□□□□□□/□□□□

□□□□□□□□□□□□

□□ play_topup_monetary.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",
        "destination_msisdn": "{{ customer_phone }}",
        "message_body": "Thanks for topping up {{ monthly_cost
        }}"
      }
    status_code: 201
    ignore_errors: True

```

□□□□

1. □ API □□□□□□□□□□□□
2. □□□□□□□□□□
3. □□ Stripe API □□□□□
4. □ OCS □□□□□□
5. □ CRM □□□□□
6. □□□□ SMS□□□ □ignore_errors: True□□□□□□□□□□□□□□

000

00000000000000000000OmniCRM 00 000000000000000000000000000000

0000000000000000

1. 0000000000000000000000
2. IPTV 0000000000 1 0 service_id
3. 00 IP 0000000000 1 0 service_id

0000000000

- 00000000000000000000 service_id
- 0000000000 extra_vars
- 00000000000000 Provision 00

0000000000

000000000000000000000000000000

00000000

0000000000 **2**

000000000000000000

- API 000000000000
- 0000000000 assert: that: response.status == 200
- 0000000000
- 0000000000
- 0000000000
- OCS/OCS 00

0000000000 **3**

00000000000000000000

- 發送 SMS/email 通知
- 配置 `ignore_errors: True`
- 配置通知

通知

配置通知

- 配置 YAML 通知
- 配置通知
- 配置通知
- 配置 Ansible 通知

配置通知

- Ansible 通知
- 配置 `stdout` 通知
- 配置 `stderr` 通知
- 配置通知
- 配置通知

配置通知

配置通知 2 配置通知 `crm_config.yaml` 配置 `provisioning.failure_list` 配置通知

配置通知

- 配置
- 配置/配置
- 配置
 - 配置
 - 配置
 - 配置
- 配置/配置
- 配置 Ansible 配置



OmniCRM  


API



```
GET /crm/provision/provision_id/<id>  
Authorization: Bearer <token>
```



```
{  
  "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": "{...}"  
    }  
  ]  
}
```


□□□□

- □□□□□□□□□□ customer_id□service_id □□□□□□
- □□□□□□□□□□□□□□□□□□□□□□
- □□□□□ Provision_Event □□□□□□□□□□/□□□□

□□

- □□□□□□□□□□ customer_id□product_id□service_id
- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□/□□
- **IP** □□□□□□□□□□□□□□□□□□

□□

- □□□□□□□□□□ API □□□□□□
- □□□□□□□□□□ 2-5 □□□□□□□□
- □□□□□□□□ Ansible □□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

- `concepts_ansible` - □□ Ansible □□□□
- `concepts_sim_provisioning` - SIM □□□□□□□□ eSIM□
- `concepts_api` - CRM API □□□□□□□□
- `concepts_products_and_services` - □□□□□□□□
- `administration_inventory` - □□□□□□□□
- `□□□□□□` - □□□□□□□□□□

Step 2: Data Retrieval

Retrieve data from the database using a query.

Example:

```
SELECT * FROM users WHERE name = 'John';
```

```
SELECT * FROM users WHERE name = 'John Smith' OR name = 'John Doe' OR name = 'Johnson Enterprises';
```

Output:

- Name
- ID
- Email
- Address

Example:

```
SELECT * FROM users;
```

Output:

Step 3: Data Manipulation

Manipulate data using SQL commands like INSERT, UPDATE, and DELETE.

□□□□□□□□

□□□□□□

SIM□ *

□□SIM□□ • SIM-00123 - ICCID□8944...0001□□□ • SIM-00124 - ICCID□
8944...0002□□□ • SIM-00125 - ICCID□8944...0003□□□

□□□□ *

□□□□□ • +44 7700 900123□□□□ • +44 7700 900124□□□□ • +44 7700
900125□□□□

□□□□□□□□

- □□□*□□□□□□□□
- □□□□□□□□□□□□“□□”□“□”□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□

□□□□□□

- □□□□□□□□□□□□□□□□

- 00000000000000000000
- 000000"00"0000"0000"

00 4000000000000000

000000000000000000000000

00000000000000000000

00

[0] [0]

00000000

- 00000000000000000000
- 000000000000000000000000

000000 00000000000000"0"0000000000000000000000000000000000"0"0

00 50000000

000000000000000000000000

000John SmithID01230 000000 - 20GB00

000 • SIM00SIM-001230ICCID08944...00010 • 000000+44 7700 900123

000 • 00000£0.00 • 00000£15.00

00000000

[00] [000000]

00**"000000"**-0000000000

00 60000000

0000000000000000

✓ 000000 ✓ SIM ICCID 8944...0001 ✓ 000000+44 7700 900123 ↻
OCS 000000... 0 000000 0 00000000

00060000030000

00000.200000API00000000

000000

0000

↻ 00000

0 00000000

0000000000

00 7000

000

000000John Smith00

00ID0789 00000000 - +44 7700 900123 000000

[00000] [00]

00**“0000”**00000000000000

000

00000000

00000000OCS

000000000000000000000000000000

00ID0456

[00000] [00] [00]

- 000000000000000000000000
- 0000000000
- 000000000000000000000000???

□□□□

□□□□□□□□□□

□□□□□□□□“addon”□□□□□□□□□□□□□□

1. □□□□□□

- □□□□□□□□□□
- □service_type□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□

2. □□□□□□

- □□□□□□□service_id□□□□
- □□□□□□□OCS□□
- □□□□□□□□□□□□

3. □□□□□

- □□□□□□□□□□
- □□□□□□□□□□□□□□□□

□□□

□□□□□□ • □□ - +44 7700 900123□ID□789□ • □□ - +44 7700 900456□ID□790□

□□□□□□□□□□□□□□ [□□ - +44 7700 900123 ▼]

[□□] [□□]

□□□□□□□

□□□□□□□□□□□

- □□□□ - □□□□□□□□□□□□
- □□□□□ - □□□□□/□□□□□□
- □□□□□ - □□□□□□□□□□□□□□□□

□□□□□□□

□□□□□□[□□□□□□ - 123 Main St ▼]

□□□□□□[Jane Doe - <jane@acme.com> ▼]

□□□□□□[Bob Smith - <bob@acme.com> ▼]

□□□□□□[PO-2025-001234_____]

[□□] [□□]

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□

1. □□CSV□□□□□□
2. □□□□□□□□□□
3. □□□□□□□□□□
4. □□□□□□□□□□

□□□□□□

□□□□ **1**□□□□□□

1. □□□□□□□□□□
2. □□□□□□ → □□
3. □□□□□□□□
4. □□**“□□ - 20GB□□”**
5. □□**“□□□□□□”**
6. □□□□□□□□“+1234567890”
7. □□□□□□**John Smith**
8. □□□□□□□□SIM□
9. □□□□□□□□
10. □□□□□□
11. □□□□□□
12. □□□□□□□□

13. SIM卡

2

1. John Smith
- 2.
3. **
- 4.
5. ** - 100Mbps**
6. /
7. CPE
- 8.
- 9.
- 10.

3

1. “”
- 2.
- 3.
4. “”
5. “5GB”
- 6.
- 7.
- 8.

“”

-
- -
 -

- 認證碼

“認證碼”

- 認證碼
- 認證碼
 - 認證碼
 - 認證碼“*”“*”
 - 認證碼

“認證碼”

- 認證碼
- 認證碼
 - 認證碼
 - 認證碼ID
 - 認證碼

認證碼

- 認證碼
- 認證碼
 - 認證碼5
 - 認證碼
 - 認證碼OCS
 - 認證碼

認證碼

- 認證碼OCS
- 認證碼
 - 認證碼OCS
 - 認證碼SIM/HLR/HSS
 - 認證碼RADIUS/DPI
 - 認證碼

□□□□

□□□□□

- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□
- □□□□□□□□□□□□

□□□□□

- □□□□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□

□□□□□

- □□□□□□□□□□□□□□□□
- □□□□□□□□□□“□□”
- □□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□

□□□□□□

- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□□□□□□□

□□□□

- `csa_add_service` - □□□□
- `concepts_products_and_services` - □□□□□□

- `administration_inventory` - □□□□
- `concepts_provisioning` - □□□◆◆□□□
- `guide_product_lifecycle` - □□□□□□□□□□□□



□□□□□ Self-Care Portal <self_care_portal> □□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□



□□□□□□□□□□□□□□□□ □□ □□□□□□□□

□□□□□□

- □□□□
- □□□□□□□□□□□□□□□□
- □□□□
- □□□□□□

- 5G 网络
- 千兆

网络服务

网络服务

网络服务

- 5G - 5GB 流量
- 千兆 - 千兆
- WiFi 6 - WiFi 6
- IP - IP

网络服务

网络服务

1. 网络服务
2. “网络服务” 或 “网络服务”
3. 网络服务

网络服务

网络服务

- 网络服务 - 网络服务 VoIP
- 网络服务 - 网络服务
- 网络服务 - 网络服务
- 网络 - 网络

网络服务

网络服务

网络服务

□□□□□□

□□□□□□

- □□□□
- □□□□
- □□□□□□□□□□
- □□□□
- □□/□□□□
- □□□□□□□□
- “□□□□□□”□□

□□□□□□□□

□□ **1**□□□□□□□□

□□□□□□□□□□□□□□ “□□□□□□”

□□ **2**□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

Modem-12345 - TP-Link AX1800 • Modem-12346 - TP-Link AX1800 • Modem-12347 - Netgear RAX40

3

[] []

- -
- -

4

“”

✓ ✓ ↻ ...

- 1.
2. Ansible (provisioning_play)
- 3.
- 4.
5. OCS
- 6.

Complete Product Lifecycle Guide - Adding Addons
<guide_product_lifecycle>

概要

本報告は、最新の市場調査結果を基に、“スマート”製品に関する

市場動向を

5GB 製品

2025年1月10日 から 2025年1月17日 の期間に 約£5.00

WiFi 6 製品

2024年12月1日 から Modem-12345 の期間に 約£10.00

結論

市場調査結果を要約すると

- 市場規模は拡大している
- “スマート”製品が注目されている
- 競争相手は“ABC”と“DEF”である
- 今後の見通し

詳細情報は

- 製品A - 市場シェア率
- 製品B - 成長率
- 製品C - 顧客満足度
- 製品D - 価格競争力

お問い合わせ

お問い合わせ先: CPE 製品開発部 (連絡先情報)

お問い合わせ

お問い合わせ先: **1** (連絡先情報)

- お問い合わせ先: “お問い合わせ先”
- お問い合わせ先

3. 如何“**🔗🔗🔗**”
4. 如何“5GB 限制”
5. 限制
6. 限制

如何 2 限制 IP

1. 限制 VPN 限制 IP
2. 限制
3. 限制“IP 限制”
4. 限制 IP
5. 限制
6. 限制 IP 限制

如何 3 限制

1. 限制
2. 限制
3. 限制“WiFi 6 限制”限制
4. 限制
5. 限制
6. 限制 £10.00 限制

限制

“限制”

- 限制
- 限制 `service_type` 为 `residential/business` 限制

限制

- 限制
- 限制
 - 限制
 - 限制

- `inventory_items_list`

Inventory Items

- `inventory_items_list`
- `inventory_items`
 - `inventory_items`
 - `inventory_items`
 - `inventory_items` `inventory_items_list`

Inventory

- `inventory`
- `inventory`
 - `inventory` `auto_renew: true`
 - `inventory`
 - `inventory`

Inventory

- `guide_product_lifecycle` - `guide_product_lifecycle`
- `csa_add_service` - `csa_add_service`
- `concepts_products_and_services` - `concepts_products_and_services`
- `administration_inventory` - `administration_inventory`

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□ □□□□ <csa_add_service> □□□□□□□□ <csa_service_usage> □□□□□□□□
<csa_modify> □□□□□□□□ <concepts_products_and_services> □

□□□□□□

□□□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□

□□□□□□□□□□□□□□□□

□□

- □□ **ID** - □□□□□□
- □□□□ - □□□□□□□□□□“□□ - +44 7700 900123”□

- 網路 - 網路服務
- 網路 - IPTV/VoIP
- 網路 - 網路服務
- 網路 - 網路服務
- 網路 - 網路服務/網路服務
- 網路 - 網路服務
- 網路 - 網路服務

網路

網路

網路

- 網路 - 網路服務
- 網路 - 網路服務
- 網路 - 網路服務
- 網路 - 網路服務

網路

網路

網路

網路

網路

- 網路 - SIM 網路
- 網路 - DSL 網路
- **IPTV** - 網路
- **VoIP** - 網路

□□□□□

□□□□□□□□□□

□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□

□□□ □□□□□□□□□□□□□□

□□

□□□□□□□□□□□□□□□□

□□□□


- □□□□
- □□ ID
- □□□□
- □□□□□□□□□□□□□□
- □□ UUID

□□□ □□“0770”□□□□□□□□□□□□□□□□“0770”□□□□□

□□□□□

□□□□□□□□□□□□□□□□

□□□□□□

- □□ ID□□□□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□

□□□□

- □□□□□□□□A-Z□□□□-□□□□□□-□□□□
- □□□□□□□□Z-A□□□□-□□□□□□-□□□□
- □□□□□□□ □ ▲ □ ▼ □□□□

□□□□□□□

□□□□□□□□□□□□□□□□□□□□

□□□□□□□

□□□□□□□□□□□□□□□□□□□□

□□□□□□□

1. □□ - □□□□□□□□□□□□□□□□
2. □□ - □□□□□□□□SIM □□□□□□□□□□
3. □□ - □□□□□□□□□□
4. □□□□ - □□/□□/□□□□□□□□□□
5. □□□□ - □□□□□□□□□□□□□□
6. □□□□ - □□□□□□□□

□□□□□

□□□□□□□□□□□□□□□□□□□□□□ : □□□□□□□□□□

- □□ - □□□□□□□□
- □□ - □□□□□□□□
- □□□□ - □□□□□□□□□□□□□□
- □□□□ - □□□□□□□□□□□□□□
- □□ - □□/□□□□□□

□□□□

□□□□□□□□□□□□□□□□

□□□□

□□□□

□□□□□□□□□□□□□□

□□□

□□□□□□□□□□□□□□□

| □□ 1 □□ 2 □□ 3

□□□□□

□□□□□□□□□□□□

□□□□

□□□□□□□□□□□□□□

□□□□□□

1. □□□□□□□□
2. □□ “□□□□” □□
3. □□□□□□□□

□□

□□□□□□□□□□□□

- □□□□□□□□
- □□□□□□□□
- □□□□□□□□
- □□□□□□□□

◆◆ □□□□

□□□□□□□□

1. □□□□□□
2. □□ “□□” □□
3. □□ “□□□□” □□□□
4. □□ “□□”

□□□□□□

- □□ → □□□□ □□□□□□□□□□□□□□
- □□ → □□□ □ OCS/□□□□□□□□□□□□□□
- □□ → □□□ □□□□□□□□□□□□
- □□□ → □□□ □□□□□□□□□□□□□□

□□□□□□□□

□□ “□□□□” □□□□□□□□□□□□□□

□□□□□□

- □□ - □□□□□□□□
- □□□□ - □□□□□□□□□□
- □□□□ - □□□□□□□□□□

- **SMS** -
- - /

- -
- -
- -

- -
- -
- - /
- -
- -
- -
- -
- -

API Integration

Retrieve scheduled actions for a CGRateS account

- Endpoint - `CGRateS` API `ActionPlan`
- Method - `GET`
- Auth - API Key
- Params - Account ID

Example

Request: `CGRateS` API `ApierV1.GetScheduledActions` CRM Account ID

Response

JSON Response

- Account ID (string)
- Actions (array)

```
Account: Acme Corp, ID: FixedWireless_75628fa5, Plan: Mega, Price: $89.99
```

```
Timestamp: 2025-11-11 10:45
```

```
Meta: {}
```

- Account Name "Acme Corp"

4. `CGRateS` 的初始化

5. `CGRateS` 的更新

配置项

- `CGRateS` 的初始化
- `CGRateS` 的更新
- `CGRateS` 的更新
- `CGRateS` 的更新
- `CGRateS` 的更新

配置项

- `CGRateS` 的更新
- `CGRateS` 的更新
- `CGRateS` 的更新
- `CGRateS` 的更新

配置项

配置项

1. `CGRateS` 的更新 (配置项) 的更新

2. `CGRateS` 的更新

`CGRateS` 的更新

`Acme Corp` `FixedWireless_75628fa5` 配置项 2025-11-11 10:45

`CGRateS` 的更新

[配置] [配置项]

3. `CGRateS` 的更新

4. `ActionPlan` 的更新 `CGRateS` 的更新

- 000000000000
- 00000000000000000000

0000

- 00 CRM 0000000000
- 0000000000
- 00000000000000000000 "N/A"

000000

- 0000 00000000
- 0000000000000000 "3000" "200"
- 00000000 CGRateS 000000

0000

- 00 CRM 0000000000
- 000000 - 000000
- 000000 - 000000
- 000000 - 000000000000000000

0000

00000/000000 "N/A"

- 000 CGRateS 0000 UUID 0 CRM 00000 service_uuid 000
- 000
 - 0000000000000000
 - 00 service_uuid 00000000 ServiceType_UUID 00000
FixedWireless_75628fa5 0
 - 0000000 CRM 00000000 CGRateS 000

0000 "N/A"

- 000 CGRateS 0000 ID 0 CRM 00000000
- 000
 - 0000000000000000

- 查詢 ActionPlan 查詢 product_id 查詢 CRM 查詢
- 查詢

“查詢”查詢

- 查詢 CGRateS 查詢 NextRunTime
- 查詢
 - 查詢 CGRateS ActionPlan 查詢
 - 查詢 ActionTiming 查詢
 - 查詢 CGRateS 查詢

“查詢”查詢

- 查詢 查詢
- 查詢
 - 查詢
 - 查詢 Ansible 查詢
 - 查詢 OCS 查詢
 - 查詢

查詢

- 查詢 查詢 CGRateS 查詢
- 查詢
 - 查詢 “查詢” 查詢
 - 查詢 ActionPlan 查詢 CGRateS 查詢
 - 查詢 CGRateS API 查詢

查詢

查詢

◀ 查詢 1 [2] 3 4 5 查詢 ▶

查詢 [10 ▼]

- 10

CGRateS 设置

设置 CGRateS 的数据库连接信息

设置 ID 为 ATTR_ACCOUNT_{service_uuid}

设置 UUID 为 Mobile_SIM_c2880638 设置 ID 为

ATTR_ACCOUNT_Mobile_SIM_c2880638

设置

1. 设置数据库连接信息
2. 设置 “CGRateS 设置” 选项
3. 设置 “设置” 选项
4. 设置 / 设置 / 设置
5. 设置 “设置”

□□□□

- □□ - □□□□□□□□□□ *req.Account □ *req.IMSI □
- □□ - □□□□□□

- `*constant` - 常數
- `*variable` - 由 RSRParser 定義的變數
- `*composed` - 組成的
- `*usage_difference` - 使用差值
- `*sum` - 總和
- `*value_exponent` - 數值指數
- 其他 - 其他規則

規則

```
{
  "Path": "*req.Account",
  "Type": "*constant",
  "Value": [{"Rules": "Mobile_SIM_474a380a"}]
}
```

規則

- 規則
- IMSI/MSISDN 規則
- MaxBitrateDL/UL
- PcefPolicyName
- 規則

規則

規則

CGRateS 規則

規則

規則 ID 規則 `FLTR_ACCOUNT_{service_uuid}`

規則 規則 UUID 規則 `Mobile_SIM_c2880638` 規則 ID 規則 `FLTR_ACCOUNT_Mobile_SIM_c2880638`

□□□□□

1. □□□□□□□□
2. □□ **"CGRateS □□"** □□
3. □□ **"□□□□"** □□
4. □□/□□/□□□□□□
5. □□ **"□□□□"**

□□□□□□

- □□ - □□□□□□□□□□ `~*req.Account`□□ `~*req.Destination`□□
- □□ - □□□□□
 - `*string` - □□□□□□□□
 - `*prefix` - □□□□□□□
 - `*suffix` - □□□□□□□
 - `*empty` - □□□□□
 - `*exists` - □□□□□
 - `*notexists` - □□□□□□
 - `*timings` - □□□□/□□□□
 - `*destinations` - □□□□□□□□
 - `*rsr` - RSR □□□□□
 - `*gt / *gte / *lt / *lte` - □□□□□
- □ - □□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□

```
{
  "Element": "~*req.Account",
  "Type": "*string",
  "Values": ["Mobile_SIM_474a380a"]
}
```

□□□□□

- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□

□□□□□□□□

- □□□□ - □□□□□□
- □□□□ - □□□□□□□□□□□□□□
- □□□ - □□□□□□□□□□

□□□□□□□□

□□“□□□□”□□□□□□□□□□□□

- 項目
- CGRID項目 ID
- 項目 UUID
- 項目
- 項目/項目
- 項目
- 項目
- 項目
- 項目 JSON
- 項目 CGRateS 項目
- 項目
- 項目 JSON 項目

□□□

- □□□□□□□□□□
- □□□□□□□□

- 認證
- 授權
- 審計

認證

認證信息保存在“/usr/local/omnicrm/conf/auth.json”文件中

如下

```
{
  "uid": "UID",
  "password": "string:~*req.Account:{service_uuid}"
}
```

CGRateS API 參數

CGRateS 與 OmniCRM API 的參數

POST /crm/ocs/proxy

參數

- `method` - CGRateS API 方法 `APIerSv1.GetAttributeProfile`
- `params` - 參數
- `customer_id` - 客戶 ID
- `service_id` - 服務 ID

參數

- `tenant` - CGRateS 租戶

參數

```
{
  "method": "APIerSv1.GetAttributeProfile",
  "params": [{"ID": "ATTR_ACCOUNT_Mobile_SIM_c2880638"}],
  "customer_id": 123,
  "service_id": 456
}
```

00000

000000 OmniCRM 000000 `crm_config.yaml` 000 `ocs.ocsTenant` 000000000
CGRateS 00000000000000000000000000000000

000000

00 `cgrates_api_access` 0000000000000000 `admin` 000

000???

000 GET 0 CGRateS API 00000000000000000000000000000000

- 000 API 00
- 000000
- 000000000
- 00000000 ID
- 000000000
- 000

0000 CGRateS 0000000000000000

0000 CGRateS 00

00 CGRateS 00000“000000”

- 000 0000 `cgrates_api_access` 00
- 000 00000000000000000000000000000000

00000000000

- 000 CGRateS 0000000000000000
- 000
 - 0000000 CGRateS 00000000
 - 000000000000
 - 0000000000000 API 00
 - 000000000000000000000000

API

- API CGRateS API
- API
 - API
 - API JSON
 - API
 - API CGRateS

API

- API
- API
 - API
 - API
 - API UUID API CGRateS

API

- API
- API “API”

API CGRateS

- API GET
- API - API/API

API

API 1

1. API
2. API
3. API
4. API
5. API

□□□□ 2□□□□□□□□□□

1. □□ “□□” □□
2. □ “□□□□” □□□□□□
3. □□□□□□□□□□
4. □□□□□□□□


□□□□ 3□□□□□□□□

1. □□□□□□□□“□□”
2. □□□□□□□□□□□□□□
3. □□□□□□□□
4. □□ “□□□□”
5. □□□□

□□□□ 4□□□□□□□□□

1. □□□□□□□□
2. □□□□□□
3. □□□□□□□□
4. □□□□□□□□“□□”
5. □□□□

□□□□ 5□□□□□□□□□□□□

1. □ □□□□□□□□ □□
2. □□□□□□□□□□
3. □□ “□□□□” □□
4. □□□□/□□/□□□□ □□
5. □□□□□□□□□□□□

□□□□

□□□□□□□□□□

- 0000 000000000000
- 0000 00“0000”0000000000

000000000000

- 0000 000000000000
- 0000
 - 000000000000“0770”0000000000
 - 0000000000
 - 000000000000

000000000000

- 0000 0000000000
- 0000 0000000000000000000000

000000

- 0000 00000000
- 0000 00000 ▲▼ 000000000000

00000000

- 0000 000000000000
- 0000
 - 0000000000000000
 - 000000000000
 - 000000000000

000000

- `csa_add_service` - 000000
- `csa_modify` - 0000000000000000
- `concepts_products_and_services` - 0000000000
- `basics_customers` - 000000

□□□□□□

□□□□□□□□□□□□□□□□□□□□

□□□

□□□□□□□□□□□□□□□□

□□□□

- 0000 - 00000000000000000000
- 0000 - 000000000000
- 0000 - 00000000/00
- 000000 - 0000000000

0000

00000000000000000000000000000000

0000000

- 000000 - 000000000000
- 000 - 000000000000
- 000 - 0000000000000000“12.5GB / 20GB”
- 000000
 - 000>50% 00
 - 00020-50% 00
 - 000<20% 00
- 000 - 000000000000

00000

00000

000000000000

000 0000 (GB) 0 000 (MB)

00000

| 00012.5GB / 20GB (62%)

00000

- 000000 - 5GB 10GB 20GB 0000
- 0000 - 000000000000500GB 1TB

45 / 250 (18%)

Options

- 160 options
- [unclear]
- [unclear]

Options

[unclear]

Price (£, \$, €, ¥)

Options

£15.50 / £20.00 (77%)

Options

- [unclear]
- [unclear]
- [unclear]
- [unclear]

???

"[unclear]" [unclear]

Options

20GB

12.5 GB 7.5 GB 2025 1 25 15 10

5GB

5.0 GB 0 GB 2025 1 31 21 20

□ □□□ 3GB

□□□0 GB □□□3.0 GB □□□2025□1□18□□□□□ □□□30

□□□□17.5 GB

□□□□□

- **ID/□□** - □□□□□□□
- □□ - □□□□□□□□□□
- □□□ - □□□□□□□□□□
- □□□□ - □□□□□□□□□□
- □□ - □□□□□□□□□□□□□□□□

□□□□□

□□□□□□□□□□□□□□□□

- □□□ = □□□□
- □□□ = □□□□

□□□□□

- □□□□□□□□ 30□□□□□□□□□□□□□□□□
- □□□□□□□□ 20□□□□□□□□
- □□□□□□□□ 10□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□ CGRateS □□□□□□□□□□□□□□□□□□□□□□□□□□□□ CGRateS Actions and Topup Behaviors□

□□□□□

□□□□□ **3** □ □□□□□□□□□

□□□□□

- □□□□□□□

- 10000
- 10000
- 100000000

10000

- 100000000
- 100000000
- 100000000
- 100000000000000

1000

- 100000000000000
- 100000000000000000
- 10000000000

100000000000000000

10000

100000000000

- 100 (GB)
- 100 (min)
- 100 (msgs)
- 100 (min)

1000

1000 12.5GB 100 1000 125 min 100 1000 45 msg 100 1000 £15.50 100

10000/10000

1000000

- 容量 (GB ~ TB)
- 接続速度

料金

450GB / 500GB 月額 £45.00

特徴

主な特徴

- 柔軟なプラン
- 高速通信

詳細

`dongle=true` 設定

お問い合わせ

お問い合わせ **0 / 0**

- 料金体系 OCS 料金
- 接続
◦ 柔軟なプラン OCS 料金
◦ OCS API 料金
◦ 接続 UUID 料金

お問い合わせ

- 料金体系 OCS 料金
- 接続
◦ 柔軟なプラン
◦ 柔軟なプラン
◦ OCS API 料金

お問い合わせ

- OCS
- - OCS
 -
 - OCS

-
- -
 - OCS

-
- -
 -
 -

OCS/CGRateS

OCS CGRateS [CGRateS Actions and Topup Behaviors](#)

- 1.
2. OmniCRM GET /crm/service/{service_id}
3. API UUID OCS
4. OCS

```

{
  "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. UI 00000000000000000000 → GB0000 → 0000

6. 0000000000

7. 0 3 000000

OCS 00000000

OCS 配置と UI 表示

- `*data` → 通信 表示
- `*voice` → 音声 表示
- `*sms` → SMS 表示
- `*monetary` → 課金 表示

課金項目 + 単位 + 表示形式 (例: 1024³)

単位

- 通信 単位 → GB 単位 1024³
- 音声 単位 → 分単位 60×10⁹
- SMS 単位
- 課金 単位 → 円単位 100

表示形式

OCS 設定と UI 表示に関する情報は [CGRateS Actions and Topup Behaviors](#) を参照してください。

設定

CGRateS 設定項目

- `auto_renew = true`
- `...`
- `...`
- `...`

1. 設定

- `auto_renew = true`
- OCS 設定
- 課金項目設定

2. 設定項目

設定項目

- 設定 ID - 設定 UUID
- 設定 - 設定項目
- 設定 - 設定項目 1 日 00:00 UTC
- 設定 - 設定項目

3. 設定項目

- OCS 設定項目
- 設定項目 20GB 設定 500 設定項目
- 設定項目 30 設定項目
- 設定項目

4. 設定項目

- 設定 OCS 設定項目
- 設定項目
- 設定項目

設定項目

```
{
  "ActionPlanId":
  "ProductID_MonthlyPlan_ProductName_20GB_Mobile__ActionPlan_Monthly_F
  "NextExecTime": "2025-02-01T00:00:00+00:00",
  "ActionName_hr": "設定",
  "PlanName": "20GB 設定",
  "ActionFrequency_hr": "設定",
  "custom_NextExecTime_hr": "日 22 設定"
}
```

設定項目

- 設定 - 設定項目
- 設定 - 設定 OCS 設定項目
- 設定 - 設定項目

□□□□

- CGRateS Actions and Topup Behaviors - □□□□□□□□□□□□□□□□□□□□□□□□□□□□
- CGRateS Destinations - □□□ PLMN □□□□□□□□□□□□
- features_topup_recharge - □□□□□□□□□□
- csa_service_management - □□□□□
- csa_modify - □□□□□□□□□□□□
- concepts_products_and_services - □□□□□

□□□□□□

□□ **Web UI** □

1. □□□□ — □ CRM □□□□
2. □□□□ — □□□□□□□□
3. □□“□□□□□□” — □□□□□□□□□□□□□□□□
4. □□□□ — □□□□□□□□
5. □□□□ — □□□□□□□□□□

□□ **API** □

□□□□□□

□□□ `POST /auth/impersonate`

□□□□ `can_impersonate` □ `admin`

□□□

```
{  
  "user_id": 42  
}
```

JSON

```
{
  "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"
}
```

JSON 格式

JSON 格式

JSON 格式

- 成功 — 成功返回 JSON 格式
 - 成功
 - 成功
 - 成功
 - 成功
- 参数 — 参数
 - `impersonating_user_id` — 用户 ID
 - `target_user_id` — 目标用户 ID
 - `impersonation_start` — 开始时间
- 消息 — UI 消息
 - "成功 [消息] 成功"
 - "成功"
- 日志 — 日志
 - 成功
 - 成功
 - 在 `ImpersonationLog` 中记录

API

Web UI

1. `POST /auth/impersonate` — 開始
2. `POST /auth/stop_impersonation` — 停止
3. `POST /auth/refresh` — 刷新

API

`POST /auth/stop_impersonation`

成功

成功

```
{
  "success": true,
  "impersonation_end": "2025-01-04T15:45:00Z",
  "duration_seconds": 900,
  "access_token": "your_original_token",
  "refresh_token": "your_original_refresh_token"
}
```

成功

API

`ImpersonationLog`

- **impersonating_user_id** — 冒充者 ID
- **target_user_id** — 目標用戶 ID
- **impersonation_start** — 開始時間
- **impersonation_end** — 結束時間
- **impersonation_duration** — 持續時間

成功

- `impersonation_logs` — `impersonation_logs`
- `impersonation_logs` — `impersonation_logs`
- `impersonation_logs` — `impersonation_logs`
- `impersonation_logs` — `impersonation_logs`

`impersonation_logs`

`impersonation_logs` API `impersonation_logs`

```
GET /auth/impersonation_logs?user_id={user_id}&start_date={date}&end_date={date}
```

`impersonation_logs`

- `impersonation_logs`
- `impersonation_logs`
- `impersonation_logs`
- `impersonation_logs`

`impersonation_logs`

1. `impersonation_logs` — `impersonation_logs`
2. `impersonation_logs` — `impersonation_logs`
3. `impersonation_logs` — `impersonation_logs`
4. `impersonation_logs` — `impersonation_logs`
5. `impersonation_logs` — `impersonation_logs`
6. `impersonation_logs` — `impersonation_logs` `can_impersonate`
7. `impersonation_logs` — `impersonation_logs`

`impersonation_logs`

- `impersonation_logs` — `impersonation_logs`
- `impersonation_logs` — `impersonation_logs`
- `impersonation_logs` — `impersonation_logs`

- 0000 — 000000000000000000000000
- 0000 — 0000000000000000
- **MFA** 00 — 0000 2FA 0000000000000000

00

- 00000000 — 0000000000000000
- 0000 — 000000000000000000000000
- 0000 — 000000000000000000
- 0000 — 000000000000

0000000000

000“0000000”000000

- 000000000000 `can_impersonate` 0 `admin` 00
- 000000000000000000000000

0000000000000000

- 0000000000000000000000000000
- 0000000000000000000000

0000000000

- 00000000 API 00000000 `POST /auth/stop_impersonation`
- 000000000000 cookies 000000000000

000000000000

- 0000000000000000000000000000
- 000000000000 - 0000000000 ID

📖📖📖📖 CGRateS 📖📖

📖📖📖📖📖 OmniCRM 📖📖📖📖📖📖 CGRateS 📖📖📖📖

📖📖📖 CGRateS 📖📖📖 Python 📖📖📖📖📖📖📖📖📖 CRM 📖 OCS📖📖📖📖📖📖

📖📖

📖📖📖📖📖📖📖📖📖📖📖

1. **CGRateS/OCS** - 📖📖📖📖📖📖📖📖📖📖📖
2. **OmniCRM** - 📖📖📖📖📖📖📖📖📖📖📖

📖📖📖📖📖📖📖📖📖📖📖 Ansible Play 📖 CGRateS 📖📖📖📖

📖📖📖📖CGRateS 📖📖

CGRateS 📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖 OmniCRM 📖📖📖 CGRateS 📖📖📖📖📖📖

- 📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖
- 📖📖📖📖📖📖 MO 📖 MT 📖/SMS 📖📖📖📖📖📖📖📖
- 📖📖📖📖📖📖 2G/3G 📖📖 CAMEL📖MO-MT SMS📖USSD📖📖📖📖📖📖📖📖📖 LTE/5G 📖 Diameter
- 📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖
- 📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖 LCR📖📖📖📖📖📖

📖📖 **CRM** 📖📖 - 📖📖📖📖 OmniCRM 📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖 CGRateS 📖📖📖📖📖

[CGRateS 📖📖](#) 📖📖 [CGRateS 📖📖📖📖📖📖](#) 📖📖📖📖📖📖📖📖📖📖

Python Requests

Part 1: CRM API Authentication

```
import requests
import json

crm_url = 'https://crm.example.com/'
session = requests.Session()

headers = {
    "Content-Type": "application/json"
}

# Authentication
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("CRM Authentication Success")
```

Part 2: Making API Calls

Using the authenticated session to make API calls to the CRM system.

```

inventory_list_new = []

# 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"
})

# 
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"
})

# 
for inventory_template in inventory_list_new:
    # 
    inventory_list_existing = session.get(
        crm_url + '/crm/inventory/item_template/',
        headers=headers
    )

    # 
    if inventory_template['item'] in [x['item'] for x in
inventory_list_existing.json()]:
        # 
        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"👉👉👉👉: {inventory_template['item']}")
else:
    # 🐞🐞🐞🐞
    response = session.put(
        crm_url + '/crm/inventory/item_template',
        json=inventory_template,
        headers=headers
    )
    assert response.status_code == 200
    print(f"👉👉👉👉: {inventory_template['item']}")

```

👉👉👉👉

- **item**👉👉👉 - 🐞🐞🐞
- **itemtext1_label** - 🐞🐞🐞🐞🐞🐞🐞🐞
- **itemtext2_label** - 🐞🐞🐞🐞🐞🐞🐞🐞
- **wholesale_cost** - 🐞🐞🐞
- **retail_cost** - 🐞🐞🐞🐞🐞
- **allow_dropdown_staff** - 🐞🐞🐞🐞🐞🐞🐞🐞
- **allow_dropdown_customer** - 🐞🐞🐞🐞🐞🐞🐞🐞
- **icon** - Font Awesome 🐞🐞

🐞 3🐞🐞🐞 **CGRateS**

```

import cgrateshttpapi
import time

OCS_Obj = cgrateshttpapi.CGRateS("ocs.example.com", "2080")

tenant = "your_tenant_name"
tpid = str(tenant) + "_" + str(int(time.time()))

```

4 CGRateS

1. - /SMS Dest_AU_Mobile Dest_International_UK
2. **PLMN** - Dest_PLMN_OnNet Dest_PLMN_US_Verizon

- /SMS →
- → PLMN

-
- PLMN
- PLMN mccXXX.mncYYY
-

CGRateS -

5 CGRateS

CRM CGRateS

CGRateS

- Python OCS_Obj.SendData
- *topup *topup_reset
- SMS
- Balanceld BalanceType DestinationIDs Units ExpiryTime Weight
-
-


```

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", # 套餐 Action_au-premium-
plan-1
    "product_name": "AU 套餐 1",
    "comment": "套餐 100GB 通话3000 短信3000 6GB 流量",
    "provisioning_json_vars": "",
    "terms": "套餐 30 天",
    "service_type": "mobile",
    "residential": True,
    "business": True,
    "enabled": True,
    "icon": "fa-solid fa-mobile",
    "features_list": "['套餐 100GB 通话', '套餐 3000 短信', '套餐 3000 短信 SMS', '6GB 流量']",
    "wholesale_cost": 45.00,
    "wholesale_setup_cost": 0,
    "auto_renew": "True",
    "customer_can_purchase": True
})

```

套餐

```

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", # 中国 Action_au-data-
addon-20gb
    "product_name": "AU 中国 20GB",
    "comment": "中国 20GB 中国",
    "provisioning_json_vars": "",
    "terms": "中国",
    "service_type": "mobile",
    "residential": True,
    "business": True,
    "enabled": True,
    "icon": "fa-solid fa-mobile",
    "features_list": "['中国 20GB 中国']",
    "wholesale_cost": 22,
    "wholesale_setup_cost": 0,
    "auto_renew": "False",
    "customer_can_purchase": True
})

```

中国

- **category**中国 - 中国
 - standalone - 中国
 - addon - 中国
 - bundle - 中国
- **provisioning_play**中国 - 中国 Ansible playbook中国
 - play_topup_charge_then_action 中国 play_simple_service 中国
- **relies_on_list** - 中国 product_slugs 中国
- **contract_days** - 中国0 = 中国30 = 中国
- **retail_cost**中国 - 中国/中国
- **retail_setup_cost** - 中国
- **product_slug**中国 - 中国 **CGRates** 中国 Action_{product_slug} 中国
- **product_name**中国 - 中国

- **comment** - 説明
- **provisioning_json_vars** - ansible playbook の JSON 変数
- **terms** - 利用規約
- **service_type** - サービスタイプ
- **residential** - 住宅用
- **business** - ビジネス用
- **enabled** - 有効/無効
- **inventory_items_list** - ansible Python 変数
- **icon** - Font Awesome アイコン
- **features_list** - ansible Python 変数
- **wholesale_cost** - 小売価格
- **wholesale_setup_cost** - 小売設置価格
- **auto_renew** - 自動更新 "True" か "False" 指定
- **customer_can_purchase** - 顧客が購入可能かどうか

ansible-playbook

7 API

```
#
product_list_existing = session.get(crm_url + '/crm/product',
headers=headers)
existing_products = product_list_existing.json()

#
for product in product_list_new:
    print(f": {product['product_slug']}")

    #
    if product['product_slug'] in [x['product_slug'] for x in
existing_products]:
        print(f": {product['product_slug']}")

        # product_id
        product_id = [
            x['product_id']
            for x in existing_products
            if x['product_slug'] == product['product_slug']
        ][0]

        product['product_id'] = product_id

    #
    response = session.patch(
        crm_url + '/crm/product/' + str(product_id),
        json=product,
        headers=headers
    )
    print(f": {response.status_code}")
    assert response.status_code == 200
    print(f": {product['product_slug']}")
else:
    #
    print(f": {product['product_slug']}")
    response = session.put(
        crm_url + '/crm/product',
        json=product,
        headers=headers
    )
    print(f": {response.status_code}")
```

```
assert response.status_code == 200
print(f"response: {product['product_slug']}")
```

API Calls

CRM API CGRateS API

```
# CGRateS API
ActionsId = "Action_au-premium-plan-1"

# CRM API
product_slug = "au-premium-plan-1"

# playbook API
cgr_action_name = "Action_" + product_slug
# ID: "Action_au-premium-plan-1"
```

Playbooks

ExecuteAction API SetBalance API
CGRateS API OmniCRM-
API/Provisioners/plays/play_topup_charge_then_action.yaml
ExecuteAction API Playbooks

playbook product_slug

```

# play_topup_charge_then_action.yaml
- name: cgr_action_name
  set_fact:
    cgr_action_name: "Action_{{
api_response_product.json.product_slug }}"

#
- name: 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 }}"
        }]
      }

```

CGRateS " " "

Ansible Playbooks

"50GB " "

1. CGRateS

5 " " 50GB " "

- **ActionsId:** "Action_50gb-data-pack" product_slug
- **BalanceType:** "*data"
- **DestinationIDs:** "Dest_PLMN_OnNet"
- **Units:** 50 * 1024 * 1024 * 1024 50GB

- **Identifier:** `"*topup"` □□□□□□

2. □ 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", # □□□ ActionsId □□□□□
    "Action_"
    "product_name": "50GB □□□",
    "comment": "□□□ 30 □□ 50GB □□□□",
    "provisioning_json_vars": "",
    "terms": "□□□ 30 □□□□□□□□□□□□□□□□□□□□□□□□□□□□",
    "service_type": "mobile",
    "residential": True,
    "business": True,
    "enabled": True,
    "icon": "fa-solid fa-database",
    "features_list": "['50GB □□□□', '30 □□□□', '□□□□']",
    "wholesale_cost": 20.00,
    "wholesale_setup_cost": 0,
    "auto_renew": "False",
    "customer_can_purchase": True
}

# □ CRM □□□
response = session.put(
    crm_url + '/crm/product',
    json=product_50gb_data,
    headers=headers
)
assert response.status_code == 200
print("□□□□□□: 50gb-data-pack")
```

3. □□□□

□□□□□□□□□□

1. 创建 "50GB 套餐" product_slug: 50gb-data-pack
2. Playbook play_topup_charge_then_action
3. Playbook 创建 Action_50gb-data-pack
4. Playbook 调用 APIerSv1.ExecuteAction ActionsId Action_50gb-data-pack
5. CGRateS 创建
6. 创建 50GB 套餐

创建 Playbooks

创建

创建 CRM 创建 CGRateS 创建 CGRateS

1

SMS

CRM

```
{
  "product_slug": "premium-monthly-plan", # Action_premium-monthly-plan
  "product_name": "",
  "category": "addon",
  "retail_cost": 59.00,
  "auto_renew": "True",
  "contract_days": 30,
  "provisioning_play": "play_topup_charge_then_action",
  "features_list": "['100GB ', '5000 ', '5000 SMS', '']"
}
```

*topup_reset Python

2. 10GB 附加服務

10GB 附加服務

CRM 設定

```
{
  "product_slug": "10gb-addon", # 10GB 附加服務
  "product_name": "10GB 附加服務",
  "category": "addon",
  "retail_cost": 15.00,
  "auto_renew": "False", # 不續訂
  "contract_days": 0,
  "provisioning_play": "play_topup_charge_then_action",
  "features_list": ["10GB 附加服務", "30 天免費試用"]
}
```

10GB 附加服務 *topup 10GB 附加服務 CGRateS 10GB - 10GB

3. 50GB 固定月費

50GB 固定月費

CRM 設定

```
{
  "product_slug": "fixed-50gb-monthly", # 50GB 固定月費
  "product_name": "50GB 固定月費",
  "category": "addon",
  "retail_cost": 35.00,
  "auto_renew": "True",
  "contract_days": 30,
  "provisioning_play": "play_topup_charge_then_action",
  "features_list": ["50GB 固定月費", "30 天免費試用", "10GB 附加服務"]
}
```

50GB 固定月費 *topup_reset 50GB 固定月費 CGRateS 50GB - 50GB

□□□□□□□□□□□□□□□□

```
# 1. □□□□□□□□□□□□
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. □□□□□□
OCS_Obj.SendData({
  "method": "ApierV2.SetTPRate",
  "params": [{
    "TPid": tpid,
    "ID": "RT_DATA_$0_10_PER_MB",
    "RateSlots": [{
      "ConnectFee": 0,           # □□□□□
      "Rate": 0.10,             # □□□ $0.10
      "RateUnit": "1048576",    # 1 MB□1024 * 1024 □□□
      "RateIncrement": "1048576", # □ MB □□
      "GroupIntervalStart": "0s"
    }]
  }]
})

# 3. □□□□□□□□□□□□
OCS_Obj.SendData({
  "method": "ApierV2.SetTPRatingPlan",
  "params": [{
    "TPid": tpid,
    "ID": "RP_PAYG_DATA",
    "RatingPlanBindings": [{
      "DestinationRatesId": "DR_DATA_PAYG",
```

```

        "TimingId": "*any",
        "Weight": 10
    }
}
})

# 4. 設定TPRatingProfile
OCS_Obj.SendData({
    "method": "ApierV2.SetTPRatingProfile",
    "params": [{
        "TPid": tpid,
        "Tenant": tenant,
        "Category": "data",
        "Subject": "*any", # 設定Subject
        "RatingPlanActivations": [{
            "ActivationTime": "2024-01-01T00:00:00Z",
            "RatingPlanId": "RP_PAYG_DATA",
            "FallbackSubjects": ""
        }]
    }]
})

# 5. 設定APIOptions
OCS_Obj.SendData({
    "method": "APIerSv1.LoadTariffPlanFromStorDb",
    "params": [{
        "TPid": tpid,
        "DryRun": False,
        "Validate": True,
        "APIOpts": {}
    }]
})

```

注意事項

- 設定TPRatingProfile時，TPRatingProfile的Weight值必須為 **\$0.10** 或 **MB**
- 設定TPRatingProfile時，Subject值必須為 `*monetary`
- 設定TPRatingProfile時，RatingPlanId值必須為 `1 MB`

□□□□□□□□

```
# □□□□□□□□□□
OCS_Obj.SendData({
  "method": "ApierV2.SetTPRate",
  "params": [{
    "TPid": tpid,
    "ID": "RT_VOICE_DOMESTIC_$0_30_PER_MIN",
    "RateSlots": [{
      "ConnectFee": 0.15,           # $0.15 □□□□
      "Rate": 0.30,               # □□□ $0.30
      "RateUnit": "60s",         # □□□□□
      "RateIncrement": "60s",    # □□□□□□□□
      "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 □□□□
      "Rate": 1.50,              # □□□ $1.50
      "RateUnit": "60s",
      "RateIncrement": "1s",     # □□□□□□□□□
      "GroupIntervalStart": "0s"
    }
  ]
}]
})
```

□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□


```
{
  "Identifier": "*cdrlog",
  "BalanceType": "*generic",
  "ExtraParameters": "
  {\"Category\\":\"^activation\\",\"Destination\\":\"Product Name\\"}",
  "Weight": 80
}
```

4. 平衡 ID

```
# 100GB
"BalanceId": "Premium_Data_100GB"
"BalanceId": "AU_Voice_Domestic__" + str(units)

# 3000
"BalanceId": "data1"
"BalanceId": "balance"
```

5. 單位換算

```
# 100GB - 單位換算
Units = 100 * 1024 * 1024 * 1024 # 100GB 單位換算
Units = 3000 * 60 * 1000000000 # 3000 單位換算

# 107374182400
Units = 107374182400
```

6. 單位換算

單位換算 = 單位換算

```

# 重置账户 = 700
{"Identifier": "*reset_account", "Weight": 700}
{"Identifier": "*topup_reset", "Weight": 90}
{"Identifier": "*cdrlog", "Weight": 80}

# 余额权重 = 1200
"BalanceWeight": 1200 # 100/1000
"BalanceWeight": 1100 # 1000
"BalanceWeight": 1000 # 1000

```

7. 计费策略

计费策略配置

```

# 100MB
Units = 100 * 1024 * 1024 # 100MB 计费

# 100GB
Units = 100 * 1024 * 1024 * 1024 # 100GB

```

计费策略

计费策略配置

计费策略 Playbook 配置 "计费策略" 的 "ActionsId 配置"

计费策略 CGRateS 配置

计费策略

1. 计费策略 CGRateS 配置
2. 计费策略 `ActionsId = "Action_" + product_slug`
3. 计费策略

“ ”

DestinationIDs

1. CGRateS
2. *any
3. ID

1. *topup *topup_reset
2. 0
- 3.

*topup_reset *topup

- Identifier *topup_reset *topup
- Balanceld

CGRateS



OmniCRM
AMBER

SMS

OmniTouch OWL

OmniTouch OWL

- CBE - Web
- CBC - 2G/3G/4G/5G

OWL Web Chrome/Firefox/Safari/Edge



PWS

-
-
- 500
-
- CBC
- TOTP
-

SMS 10

500

"

-
-
-
-
-

- -
- -

2

OWL

- -
- -
- -

“”

3

TOTP

- RSA SecurID
- Google Authenticator, Authy, Microsoft Authenticator
TOTP

2FA

2FA

1. Google Authenticator, Authy, Microsoft Authenticator
2. OWL QR
- 3.
- 4.
- 5.

2FA <2fa>

OWL → 2FA

-
-

- 品牌 - 品牌名称
- 品牌 - 品牌名称
- 品牌 - 品牌名称

品牌

品牌CBC品牌名称

- 品牌 - ISO品牌名称en es fr zh
- 品牌 - 品牌名称1395品牌

品牌名称品牌名称品牌名称品牌名称

品牌

品牌名称

- 品牌 - 品牌名称ID品牌名称
- 品牌 - 品牌名称MCC-MNC
- **RAT** - 品牌名称LTE 5G UMTS GSM

品牌名称品牌名称品牌名称品牌名称

品牌名称品牌名称

品牌Web UI

1. 品牌名称 - 品牌名称CBC品牌名称
2. 品牌“品牌名称” - 品牌名称
3. 品牌名称
 - 品牌名称品牌名称4370
 - 品牌名称品牌名称
 - 品牌名称品牌名称5-60
 - 品牌名称999品牌名称
 - 品牌名称品牌名称品牌名称
 - 品牌名称品牌名称“品牌”

4. 國際化

- 雙“引號”
- 國際化
- 國際化GSM7 1395 Unicode
- 國際化

5. 國際化

- 雙“引號”
- 國際化
- 國際化MCC-MNC
- 國際化RAT LTE 5G
- 國際化

6. □□□□ - □□□□□□□□“□□□□”

□□**API**□

□□□ `PUT /crm/cbc/`

□□□□ `CREATE_CBC_MESSAGE`

□□□□

```
{
  "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"
    }
  ]
}
```

□□□

OmniCRM CBC

OmniCRM CBC OWL CBC

OWL CBC

CBC URL `crm_config.yaml`

```
cbc_url: "http://cbc.example.com:8080"
```

OmniCRM

OmniCRM CBC

1. OmniCRM CBE
2. OmniCRM CBC API
3. `{cbc_url}/alerts/send` HTTP POST
4. CBC CBSP SBc-AP N50
5. CBC
6. OmniCRM

OmniCRM

OmniCRM CBE CBC

OmniCRM

OmniCRM 2G/3G/4G/5G / OmniCRM

OWL 2G 3G 4G 5G

CBSP - 2G/3G - BSC

OmniCRM CBSP CBC 2G GSM BSC

- 2G 3G /
- BSC

- 3GPP RNC/BSC OWL CBC
- CBSP

3GPP RNC/BSC SABP RNC/BSC CBSP RNC

SBc-AP - 4G/5G - MME/IWF

SBc-AP OWL CBC 4G/5G eNodeB/gNodeBs MME

- 4G LTE
- 5G 2025
- MME OWL CBC
- SBc-AP

N50 - 5G - AMF

5G N50 OWL CBC 5G gNodeBs AMF

- OWL CBC
- 2025 5G SA AMF
- 5G SA

MNO

OWL CBC

- CBC MNO
-
- MME RNC BSC

OWL CBC

OWL  

- **NetAct**
- **U2000 / U2020**
- **NetNumen / ZXPOS**
- **ENM**

OmniTouch

RBAC

OWL RBAC



RBAC

- - OWL
- -
- -
- -

RBAC

1. -
2. -
3. -

CRUD

- -
- -
- -  

- `role` - `role`

CBC

- `CREATE_CBC_MESSAGE` - `role`
- `VIEW_CBC_MESSAGE` - `role`
- `UPDATE_CBC_MESSAGE` - `role`
- `DELETE_CBC_MESSAGE` - `role`

`role`

`role` `<rbac>`

role

`role`

`role`

1. `role` URL
2. `role`
3. `role` 2FA `role` 2FA

`role`

`role` `<authentication_flows>`

`role`

`role`

1. `role` "role"
2. `role`
3. `role`

`role`

`role` → `role`

1. `role` → `role`

OWL

→

OWL

-
-
-
-
-
-

-
- NDMO
-
-

□□□□

□□□□ VIEW_CBC_MESSAGE □ CREATE_CBC_MESSAGE

□□□□

□□□□

□□□□

- □□□□ - □□□□□□□□□□
- □□□□□ - □□□□□□□□□□
- □□□□ - □□□□□□□□
- □□□□ - □ cellSites.txt □□□□

□□□□

- □□□□□ - □□□□□□□□
- □□□□ - □□□□□□□□
- □□□□ - □□□□□□□□

□□□□

- □□□□ - □□□□□□□□□□
- □□□□ - □□□□□□□□□□
- □□□□ - □□□□□□□□□□

□□□□

□□1□□□□□□□

□□□□□□□□□□□□□□□□ /cellSites.txt □□□□□□□

- **lat/lng** - □□□□
- **site_name** - □□□
- **sectors** - □□□□□□□□□□

□□2□□□□□□□

□□□□□□

1. □□□□□□□□□□

- □□□□
- □□□□
- □□□□□□□□

2. □□□□□□□□□□

- □□□□
- □□□□□□
- □□□□□□□□□□

3. □□□□□□□□□□

- □□□□□□
- □□□□□□
- □□□□□□□□

□□**3**□□□□□□□□

□□□□□□

- □□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□
- □□□□□□□□

□□□□□□

□□**4**□□□□□□

□□□□□□□□

- □□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□
- □□□□□□□□□□

□□□□□□

□□□□□□□□□□□□

□□□□□

1. □□**CBC**□□□□□□□□□□□□
2. □□□□□□□□□□□□□□
3. □□□□□□□□□□□□□□
4. □□□□□□□□□□□□□□□□
5. □□**CBC**□□□□□□□□□□□□

□□□□□□□□□□

□□□□□

□□□□□

□□□□□□□□□□□□□□□□□□

- □□□□ - □□□□/□□□□□□
- □□□□ - □□□□□□□□
- □□□□ - □□□□□□□□□□**AMBER**□□
- □□□□□□□□ - □□□□/□□□□□□

□□□□□□□□

□□□□□ □`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}
        ]
      }
    ]
  }
]
```

□□□□□□□□

1. □□□□□□□□□□
2. □□□□□□□□□□
3. □□□□□□□□□□□□
4. □□□□□□□□□□

□□□□

Google Maps API

□□□□□□□□□□Google Maps API□□□

```
REACT_APP_GOOGLE_API_KEY=your_google_maps_api_key_here
```

□□□□□

□□□□□□□□□□□□public/cellSites.txt□

```
# [lat,lng,name,sector1,sector2,sector3  
-33.8688,151.2093,SYD_CENTRAL,0,120,240  
-33.8650,151.2070,SYD_HARBOUR,45,165,285
```

□□□□□□□□□□

- □□□□□□
- □□□□□□
- □□GPS□□□□□□
- □□□□□□□□RNC□□□□

□□□□□□

□□□□ `mapStyles.js` □□□□□□□□□□□□□□□□

- □□□□□□
- □□□□□□
- □□□□□□
- □□□□□□

□□

□□□□□□

- □□□□□□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□

□□□□□□

- □□□□□□□□□□□□
- □□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□

□□□□□□

- 繁體中文
- 簡體中文
- 繁體中文(台灣)
- 簡體中文(大陸)

繁體中文

繁體中文(台灣) Omnitouch 繁體中文(大陸) OWL 繁體中文

繁體中文

- 繁體中文(台灣) 360 繁體中文
- 繁體中文(大陸)
- 繁體中文(台灣) “繁體中文” “繁體中文”
- 繁體中文
- 繁體中文(台灣)

繁體中文

- 繁體中文(台灣)
- 繁體中文(大陸)
- 繁體中文(台灣)
- 繁體中文(台灣) Unicode 繁體中文

繁體中文

- 繁體中文(台灣)
- 繁體中文(大陸)
- 繁體中文(台灣)
- 繁體中文(台灣)

繁體中文

- 繁體中文(台灣) 4379 4380-4381 繁體中文
- 繁體中文(大陸)
- 繁體中文(台灣) CBC 繁體中文

- 0000000000000000
- 0000000000000000

0000000000000000 Omnitouch 0000000000000000

000000

- 0000000000000000
- 0000000000000000
- 0MNO 00000000
- 0000000000000000

000000

00 Omnitouch 0000000000000000

000000

000000000000

- 000 **K8s** - Kubernetes 0000000000
- 000 - VMware 0 Proxmox 0 HyperV
- 000 - 00000000
- 000 - AWS 0 GCP
- 000 - 00000000

000000

0000000000

- 000000000000 OWL CBE 00000000000000000000
- 000000/000000000000000000000000
- 000000000000000000000000 NDMO 000000000000000000000000

00000000000000000000 BSC/MME 0000000000

CBE 0 CBC 00

MNO CBC

CBE CBC

- CBE CBC
-
- CBE CBC TCP 443 TLS
- MNO NDMO CBE

CBE Citrix VPN

-
-
-

CBE NDMO /

3

- **2** CBE /
- **1**

- 50GB
- **CPU** 2 CPU
- **RAM** 8GB
- Omnitouch
- TCP 443 CBC TLS

CBC MNO

3

- 2 CBC
- 1

- 50GB
- CPU 2 CPU
- RAM 8GB
- Omnitouch
- MME RNC BSC
- CBC TCP 443 CBC

OWL

1. -
2. -
3. CBC - CBC
4. / -
5. -
6. IP - NDMO MNO IP
7. VPN - Omnitouch
8. CBC NDMO -
9. CBC MNO -
10. - CBC
11. CBC CBC -
12. CBC - BSC/MME/AMF
13. API NMS -
14. -
15. / -

16. 000000 - 0000000000

0000

00000000000000000000

GSM 70000000000000000

- 00093000
- 000150 × 93 = 001395000

Unicode UCS-20000000000000000

- 00041000
- 000150 × 41 = 00615000

OWL000

- 0000000500000
- Web UI00000000000000000000

000000

0000000000000000

- 00000000000000
- 00000000
- 0000000000
- CBC API000000
- 00000000

0000000000000000

```
SELECT * FROM cbc
WHERE created >= '2025-01-01'
ORDER BY created DESC;
```

□□□□□□□□

□□□□□□□□□□□□□□□□

□□□□□

- □□□2015□□□□□□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□/SMS□□□□
- □□□□□□□□□□SMS□□□□□□□□

□□□□□

- □□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□
- □□□□□/□□□□□□□□□□□□□□

□□□□□□□□

□□□□□□□□□□□□□□□□

1. □□□□□□□□□□4379□4380-4381□
2. □□□□□□□□□□□□□□
3. □□□□□□□□□□□□□□□□
4. □□□□□□□□□□□□□□□□

□□□□□

OWL□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□/SMS

□□□□□□□□SMS□□□□□□□□□□

- 臺灣電力公司
- 臺灣電力公司
- 臺灣電力公司

臺灣電力公司

臺灣電力公司

- 臺灣電力公司
- 臺灣電力公司
- 臺灣電力公司
- 臺灣電力公司

臺灣電力公司

臺灣電力公司

- 臺灣電力公司
- 臺灣電力公司
- 臺灣電力公司

臺灣電力公司/公司

臺灣電力公司

- 臺灣電力公司
- 臺灣電力公司
- 臺灣電力公司
- 臺灣電力公司

臺灣電力公司

OWL CBC 臺灣電力公司

- 臺灣電力公司
- API

- 000000
- 0000
- 00000000
- 0000

0000000000000000

00000000

0000

0000000000000000

- 00000000000000
- 0000000000000000000000000000000000
- 00000000
- 00000000000000

000000

- 000000437904380-43810000000000
- 0000000000000000000000000000000000
- 0000000000000000000000000000000000
- 00000000000000

00000000

◆◆ 0000000000000000000000000000000000/0000000000 Omnitouch 0000000000000000000000000000000000

000000

OWL 00000000000000

- 000 NetAct
- 00 U2000 / U2020
- 00 NetNumen / ZXPOS

- ENM

OmniTouch

NDMO

GIS Google Earth OmniTouch

NDMO

-
-
-
-

OmniTouch

??

-
-
-

- 2FA 2FA

2FA / 2FA

OmniTouch 2FA

- 2FA
- / /
-

1. 2FA
2. / /
3. / /
4. / /

API

API

API

- / -
- -
- -
- -

API

API

CBC

□□□□

PUT /crm/cbc/

□□□□□□

GET /crm/cbc/

□□□□

PATCH /crm/cbc/{cbc_message_id}

□□□□

DELETE /crm/cbc/{cbc_message_id}

□□□□API□□□□□Swagger□□/crm/docs/□

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□ OmniCRM □□□□□□□□□□□□□□□□□□□□□□□□

□□□ Customers <basics_customers>□ Inventory
<administration_inventory>□ Service Management
<csa_service_management>□

□□□□□□

□ CRM □□□□□□

□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□□□□□□□□□□□□

□□□□□

1. □□ - □□□□
2. □□□ - □□□□□□□□□□□□□□□□
3. □□ - □□□□
4. □□ - □□□□ICCID□□□□□□itemtext1□itemtext2□
5. □□ - □□□□□□□□□□□□ UID

□□□□□

- □□□□ - □□□□□□□□□□□□□□□□□□□□□□□□“Smith”□□“John Smith”□□“Smithson”□
- □□□□□□ - “john”□□“John”□□“JOHN”□□“john”
- □□□□ - □□□□□□□□□□□□□□□□
- □□□□ - □□□□□□□□□□□□□□ 10 □□□□

□□□□

□□□□

1. □□□□□□□□□□
2. □□**"□□"**-□□□□

□□□□□□

- □□□□ "Acme Corp"
- □□□□ "+1234567890" □ "1234567890"
- □□□□ "john@example.com" □ "john"
- □□□□ "ICCID8944" □□ "8944"
- □□ UUID □ "123e4567-e89b"

□□□□□□□□

□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□

1. □□**"□□□□□□□□"**-□□□□
2. □□□□**"□□"**-□□□□

□□□□□□

- □□□ customer_status = "Closed" □□□
- □□□□□□□□□□□□□□□□□□

□□□□□□□□□□

- □□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□

□□□□□□□□

□□□□□□□□

□□□□□□□□□□

John Smith □□ ID: 123 □□: customer

John Smith (□□□) □□ ID: 123 □□: contact

Mobile - +44 7700 900123 □□ ID: 123 □□: service

□□□□□□□□

- □~~□~~~~□~~~~□~~/□□ - □□□□□□□□□□□□
- □□ **ID** - □□□□□□□□□□
- □□ - □□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□ customer_id□□□□□□□□□□□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□

□□□□□□□□□□□□□□□□□□□□

□□□□

- /customers/{customer_id} - □□□□
- /customers/{customer_id}#4 - □□□□□□ 4□
- /customers/{customer_id}#2 - □□□□□□ 2□
- /customers/{customer_id}#8 - □□□□□□ 8□
- /customers/{customer_id}#3 - □□□□□□ 3□
- /inventory-items-list - □□□□□□

□□□#□□□□□□□□□□□□□□□□□□□□□□□□

□□

□□□□□ 10 □□□□□□□□

| □□□□ 11-20□□□ 47 □

□□□□□□□□□□□□

- □□□/□□□□□□
- □□ - □□□□□□□□
- □□ - □/□□□□□□□□□□□□

□□□□□□

□□ **1**□□□□□□□□

□□□□□□□□□□

□□□ • John Smith (□□□) - □□ ID: 123 • Mobile - 555-0123 (□□) - □□ ID: 123

□□□□□□□□□□□□

□□ **2**□□□ **SIM** □

□□□□□□□□□□□□□□ SIM□

□□□ • 8944538000000001234 (□□) - □□ ID: 456

□□□□□□ SIM □□□□□□□□□□

□□ **3**□□□□□□□□

□□□□□□ 6 □□□□□□□□□□

□□□□□□□□

□□□ • Acme Corporation (□□) - □□ ID: 789

□□ **4**□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□

□□□ • John Smith (□□□) - □□ ID: 123

□□ **5**□□□ **UUID** □□□□

□□□□□□□□ UUID□□□□□□□□□□

□□□ • Mobile - +44 7700 900123 (□□) - □□ ID: 456

□□□□

□□□□□□□□

- 姓名 - “Smith” “John Smith”
- 名字 - “John”
- 姓氏 - “Smith”
- 姓氏 - “Smith”
- 唯一标识符 **UUID** -

姓名

名字

- 姓氏

姓氏

- 名字
- 姓氏
- 姓氏
- 姓氏

姓氏

- 姓氏

姓氏

- itemtext1 ICCID MAC
- itemtext2 IMSI
- *itemtext3-20*

姓氏

- 姓氏
- 唯一标识符 **UUID**

姓氏

- 姓氏
- 姓氏
- 姓氏

- 検索条件
- ページ
- ページサイズ
- 閉じたレコードを返すかどうか itemtext1/2

API 例

API 例

```
GET /utilities/search_everything?  
search=Smith&page=1&per_page=10&search_closed_records=false  
Authorization: Bearer <token>
```

パラメータ

- search - 検索条件
- page - ページ番号 (デフォルトは1)
- per_page - ページサイズ (デフォルトは10)
- search_closed_records - 閉じたレコードを返すかどうか (デフォルトはfalse)

レスポンス

```
{
  "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
  }
}
```

□□□□□□□□

□□□□□□□□□□ SQL UNION□

```

-- Customers
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

-- Contacts
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

-- Sites
SELECT site_id AS id,
       site_name AS name,
       customer_id,
       'site' AS type
FROM customer_site
WHERE site_name LIKE '%Smith%'

UNION ALL

-- Inventory
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

```
-- Services
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%'
```

□□□□□□□□□□

□□□□

□□□□

- □□□□□□□□ LIKE □□□ %term% □
- □□□□□□□□
- □□□□□ > 100k □□□□□□□□□□□□
- □□□□□□ 10 □□□□□□

□□□□

- □□□□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□

□□□□

□□□□□□□□□□

- □□□ □□□□□□□□□□□□
- □□□
 - □□□□□ "+44 7700 900123" □□□ "07700900123" □□□□
 - □□□□□□□ "john.smith@example.com" □□□ "john" □□□□

- 000000 00000000000000000000

0000

- 000 000000000000000000
- 00000
 - 0000000000
 - 00000000000000000000
 - 00000000000000

0000000000

- 000 0000/0000000000
- 00000 0000 ID 00

00000000

- 000 000“00000000”000
- 00000 0000000000

00000

- basics_customers - 0000
- basics_navigation - 0000
- administration_inventory - 0000

- 0000000000
- 000000

0000000000 IP 000 IMSI 0000000000

0000

000000 **4** 0000

1 0000000

00000000000000000000

000

- 0000 - 00 1 0 30 0
- 000000 - 0000000000
- 000000 - 000000000000
- 00000000 - 0000000000/0000

00000

00000

- 000000000000 `REACT_APP_TOPUP_PRICE_PER_DAY` 00
- 000000 \$10 00
- 0000 `REACT_APP_CURRENCY_CODE` 00

2 0000000

00000000000000

- 0
- 0
- 000000

00000000

- 信用卡
- 銀行存款
- 支票
- 電子支付

3 信用卡

Stripe Elements, PayPal SDK

信用卡

- Visa, Mastercard, Amex
- 銀行
- Apple Pay, Google Pay, PayPal

安全

- PCI 認證
- OmniCRM
- 3D Secure
- 加密

流程

1. 信用卡
2. 銀行
3. 支票
4. 電子支付
5. 其他

其他

OCS 認證

4 其他

其他

2025 年 1 月 17 日

客户 ID: TXN-123456

标题

内容

正文

正文

正文

正文

正文

1. 正文

正文

- 状态: `succeeded`
- 金额: `days × price_per_day`
- 备注

2. 正文

- API 请求: `POST /oam/topup_dongle`
- 参数: `service_uuid` 和 `IMSI`
- 参数: `OCS/CGRateS`
- 参数: `play_topup_hotspot`

3. 正文

正文

- **HotspotTopup** 参数 - 正文
 - `payment_intent_id`

- service_uuid
- imsi
- 手机号码
- topup_amount
- 手机号码/日期/时间
- 数据源 - 数据源
 - 数据“数据源 - 7 天”
 - 数据topup_amount数据源
 - 数据 service_id 和 customer_id
- 数据源 - 数据源
 - 数据源
 - 数据源
 - 数据源Stripe payment_intent_id
- 数据源 - 数据源
 - 数据“数据 [数据源]”
 - 数据topup_amount数据源 - 数据
 - 数据源数据源

4. 数据源

数据源数据源数据源 playbook play_topup_hotspot数据源

- 数据 days 数据
- 数据 OCS/CGRateS API
- 数据源数据源
- 数据源数据源

数据源 playbook 数据源

```

- 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 }}"

```

□□□□□

□□□□□□□□□□□□□□□□□□□□

SMS □□□

□ OCS □□□□□□ □ Action_Balance_Low □ Action_Balance_Out □
 Action_Balance_Expired □□

□□□□□□□□

□ OCS/CGRateS □□□□□□□□□□□□□□□□□□

□□□□□□□□

- Action_Balance_Low - □□□□□□□□□□□□□□□□ 2 □□
- Action_Balance_Out - □□□□□
- Action_Balance_Expired - □□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

□□□□

“□□□□□□□□”

- □□□ Stripe □□□□□□□□□□□□□□
- □□□
 - □□ REACT_APP_STRIPE_PUBLISHABLE_KEY □□□□□□
 - □□ Stripe □□□□□□□□□□
 - □□□□□□□□□□ JavaScript □□

“□□□□□□□□□□...”

- □□□ □□□□□□□□ OCS □□□□□□□□□□□□ □□□□
- □□□
 - □□□□□□□□ GET /crm/provision/provision_id/<id>
 - □□ OCS/CGRateS API □□□□□□
 - □□ playbook play_topup_hotspot.yaml □□□□□□
 - □□ Ansible □□

“□□□□□□□□”

- □□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

“□□□□□□□□□□”

- □□□ UI □□□□□□□□□□□□
- □□□
 - □□ REACT_APP_TOPUP_PRICE_PER_DAY □□□□□□□□□□□□□□ \$10□
 - □□□□□□□□□□□□
 - □□□□□□□□□□□□

□□□□

□□□□□□□□

HotspotTopup

```
SELECT
  hotspot_topup_id,
  service_uuid,
  days,
  topup_amount,
  status,
  payment_intent_id,
  created
FROM hotspot_topup
WHERE status = 'Failed'
ORDER BY created DESC;
```

```
GET /crm/provision/provision_id/<provision_id>
```

Stripe

Stripe <https://dashboard.stripe.com>

- Stripe PCI 1
- OmniCRM
-
-

-
- IP

- IMSI 查詢
- 查詢

查詢

- 查詢 - 查詢
- 查詢 IP 及 IMSI
- 查詢
- 查詢 CRM 查詢

查詢

查詢

1. 查詢 - 查詢
2. 查詢 - 查詢
3. 查詢 **playbook** 查詢 - 查詢 playbook 查詢
4. 查詢 **OCS** 查詢 - 查詢 OCS API 查詢
5. 查詢 - 查詢 `REACT_APP_TOPUP_PRICE_PER_DAY`

查詢

1. 查詢 **URL** - 查詢
2. 查詢 - SMS 查詢
3. 查詢 - 查詢
4. 查詢 - 查詢

查詢

1. 查詢 **Stripe webhooks** - 查詢 webhook 查詢
2. 查詢 - 查詢 `payment_intent_id`
3. 查詢 - 查詢
4. 查詢 - 查詢
5. 查詢 - 查詢 <5 查詢

□□□□

- `payments_process` - □□□□□
- `concepts_provisioning` - □□□□□
- `Payment System Guide <payment_system_guide>` - □□□□□□□□□□
- `payments_transaction` - □□□□
- `payments_invoices` - □□□□

Table of Contents

OmniCRM Ansible

Table of Contents

OmniCRM

1. Introduction - Overview of the project
2. Prerequisites - Required software and dependencies
3. Installation - Step-by-step instructions for setting up the environment
4. Usage - How to run the application and manage the system

Table of Contents

OmniCRM

Table of Contents

Table of Contents

- Introduction
- Prerequisites
- Installation
- Usage

Table of Contents

Table of Contents

Table of Contents

Table of Contents

- wholesale_setup_cost - 0.00
- retail_setup_cost - 0.00

{}

```
{
  "retail_cost": 15.00,
  "wholesale_cost": 5.00,
  "retail_setup_cost": 0.00,
  "wholesale_setup_cost": 1.00
}
```

1. 20GB SIM Card

Prepaid mobile service with 20GB data.

20GB SIM Card

Prepaid mobile service with 20GB data. SIM card included.

1. 20GB SIM Card

Product name: Prepaid mobile 20GB

2. 20GB SIM Card

```
{
  "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 20GB SIM card included"
}
```

{}

- `product_name` - 产品名称
- `product_slug` - 产品 API 接口 URL 标识符
- `category` - "standalone" 独立产品/服务
- `service_type` - 服务类型
- `enabled` - 是否启用
- `icon` - UI 图标 FontAwesome 图标
- `comment` - 备注

第 3 步

```
{
  "retail_cost": 15.00,
  "wholesale_cost": 5.00,
  "retail_setup_cost": 0.00,
  "wholesale_setup_cost": 1.00,
  "contract_days": 30
}
```

解释

- 零售成本 £15.00
- 批发成本 £5.00
- 零售设置成本 £10.00 200% 折扣 67% 折扣
- 批发设置成本 -£1.00
- 合同天数 30 天

第 4 步

```
{
  "residential": true,
  "business": false,
  "customer_can_purchase": true,
  "available_from": "2025-01-01T00:00:00Z",
  "available_until": null
}
```

- 住宅产品

- **Inventory Items**
- **Inventory**
- **2025 1 1 Inventory**
- **Inventory Items**

Step 5

```
{  
  "auto_renew": "prompt",  
  "allow_auto_renew": true  
}
```

- `"prompt"` - Inventory items
- `"true"` - Inventory
- `"false"` - Inventory items
- `allow_auto_renew: true` - Inventory/Inventory items

Step 6

Inventory items
`<administration_inventory>` Inventory items

```
{  
  "inventory_items_list": ["SIM Card", "Mobile Number"]  
}
```

Inventory items

Inventory items OmniCRM Inventory items

- **Inventory** - Inventory items "SIM Card" "Mobile Number"
- **Inventory** - Inventory MAC Inventory items
- **Inventory** - Inventory items
- **Inventory** - Inventory items

Inventory items

inventory_items_list Python `<administration_inventory>`

```
# SIM
inventory_items_list: ["SIM Card", "Mobile Number"]

#
inventory_items_list: ["Modem Router", "Static IP Address"]

#
inventory_items_list: []

# CPE
inventory_items_list: ["Fixed Wireless CPE", "IPv4 Address", "IPv6 Prefix"]
```

1.

inventory_items_list

2.

3.

- -
- item_state "New" "In Stock" "Assigned" "Damaged"
- service_id customer_id NULL
- - /

“SIM ”

- ID
- `itemtext1` - SIM ICCID
- `item_location`

4.

- “”
-
-

5. Ansible

“” ID Ansible

```
#
# - SIM inventory_id: 5001
# - inventory_id: 5002

# Ansible
{
  "product_id": 42,
  "customer_id": 123,
  "SIM Card": 5001,           # ID
  "Mobile Number": 5002,    # ID
  "access_token": "eyJ..."
}
```

`hostvars[inventory_hostname]['SIM Card']` ID

6.

Ansible ID

```

- name: SIM
  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: ICCID IMSI
  set_fact:
    iccid: "{{ api_response_sim.json.itemtext1 }}"
    imsi: "{{ api_response_sim.json.itemtext2 }}"

```

SIM ICCID IMSI HSS

7. “”

```

- name: SIM
  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"
    }

```

- “”
-
-

XXXXXXXXXX

XXXX"XX"XXXXXXXXXX

1. XXXX **ID**XXXXXXXXXXXX service_id XXX
2. XXXXXXXXXXXXXXXXXXXXXXXX OCS XXXXXXXXXXXXXXXXXXXXXXXX
3. XXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

```
rescue:
  - name: XXXXXXXX
    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 # XXXXXXXX
```

XXXXXXXXXXXXXXXXXXXXXXXXXXXX"XX"XX

XXXXXXXXXX

XX inventory_items_list: "[]"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

- XXXX - XXXXXXXXVPN XX
- XXXXXXXX - XXXXXXXXXXXXXXXX
- XXXX - XXXXXXXXXXXXXXXX

XX "5GB XXXX"XXXXXXXXXX inventory_items_list: "[]"XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XX

Inventory

`inventory_items_list` configuration

1. SIM → SIM → SIM
2. Add "SIM" type
3. Labels
 - `itemtext1_label`: "ICCID"
 - `itemtext2_label`: "IMSI"
 - `itemtext3_label`: "PUK Code"
4. Add "SIM" type

Configuration: `<administration_inventory>`

Configuration

`inventory_items_list` configuration: `['SIM Card', 'SIM Card']`

Configuration

1 Configuration

```
# SIM Configuration
inventory_items_list: ["Primary SIM Card", "Secondary SIM Card", "Mobile Number"]
```

Configuration: "Primary SIM Card", "Secondary SIM Card"

2 Configuration

```
# SIM Configuration
inventory_items_list: ["Dual SIM Kit", "Mobile Number"]
```

Configuration: "Dual SIM Kit" SIM `itemtext1`: ICCID `itemtext2`: ICCID

Configuration


```
inventory_items_list: ["ONT Device", "GPON Port", "IPv4 Address", "Fiber Drop Cable"]
```

- ONT 設備名稱
- GPON 設備名稱 OLT 設備名稱
- IPv4 設備名稱 IP
- 設備名稱

設備名稱

```
inventory_items_list: ["Rental Modem"]
```

- 設備名稱
- 設備名稱

設備名稱

1. 設備名稱

設備名稱

- 設備 SIM 設備名稱
- 設備 IP 設備名稱
- 設備名稱

設備名稱

2. 設備名稱

設備名稱

- 設備 SIM 設備名稱
- 設備名稱
- 設備名稱
- 設備名稱

3. 設備名稱

□□□□□□

- □ SIM □□□□□□□□□□
- □□□□□□□□□□
- □□ CPE □□□□□□□□□□
- □□ IP □□□□□□

4. □□□□

□□□□□□□□□□□□□□

- □□□□ SIM □□□□
- □□□□□□
- □□□□□□□□□□
- □□□ COGS□□□□□□

5. □□□□




□□□□□□□□□□□□□□

- □□□□□□SIM □□□□□□□□
- □□□□□□□□
- □□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□

□□□ “□□□□□□”□□□

□□□□

- □□□□□□□□□□□□□□□
- □□□□□□“□□”□□□□□□
- □□□□□□“□□”□□“□□”
- □□□□□□□□□□□□□□□□□□

□□□□□□

1. □□□□□□□□□□□□□□ → □□ → □□

2. ...
3. ... → ... → ...
4. ... "..."
5. ... service_id ... NULL

...

...

- inventory_items_list ... []
- inventory_items_list ... NULL
- ... "..."

...

- ... inventory_items_list: ["Type1", "Type2"]
- ...
- ... API ... inventory_items_list

... "..."

...

- ...
- ... ID ...
- ...

...

- ... hostvars[inventory_hostname]['SIM Card']
- ... {{ hostvars[inventory_hostname]['SIM Card'] | int }}
- ...

... <administration_inventory>

... **7**...

...

```
{
  "features_list": "20GB 存储空间可扩容至30 GB",
  "terms": "30 天无理由退换货"
}
```

关键词

标题 - 关键词

描述

- 标题 - 关键词
 - “20GB”“30GB” - 关键词
 - 关键词
- 标题 - 关键词
 - “关键词”关键词
 - “关键词”关键词
 - 关键词
- 标题 - 关键词
 - 关键词“关键词”→关键词❓❓❓
 - 关键词
 - 关键词
- 标题 - 关键词
 - 关键词
 - 关键词
 - 关键词
- SEO** 关键词 - 关键词
 - 关键词“关键词”→关键词
 - 关键词

标题 - 关键词

描述

- 标题 - 关键词
 - “30 天” - 31 天
 - “关键词” - 关键词

- วัตถุประสงค์
- 2. วัตถุประสงค์ - วัตถุประสงค์
 - “วัตถุประสงค์” - วัตถุประสงค์
 - “วัตถุประสงค์”วัตถุประสงค์ - วัตถุประสงค์
 - วัตถุประสงค์
- 3. วัตถุประสงค์ - วัตถุประสงค์
 - วัตถุประสงค์
 - วัตถุประสงค์
 - วัตถุประสงค์ GDPR/วัตถุประสงค์
- 4. วัตถุประสงค์ - วัตถุประสงค์
 - “วัตถุประสงค์” - วัตถุประสงค์
 - “วัตถุประสงค์” - วัตถุประสงค์
 - “วัตถุประสงค์” - วัตถุประสงค์
- 5. วัตถุประสงค์ - วัตถุประสงค์
 - วัตถุประสงค์
 - วัตถุประสงค์
 - วัตถุประสงค์

วัตถุประสงค์

วัตถุประสงค์“วัตถุประสงค์”วัตถุประสงค์ 10,000 วัตถุประสงค์

- วัตถุประสงค์
- วัตถุประสงค์“วัตถุประสงค์.....”
- วัตถุประสงค์“วัตถุประสงค์”
- วัตถุประสงค์

วัตถุประสงค์“วัตถุประสงค์”

- วัตถุประสงค์
- วัตถุประสงค์
- วัตถุประสงค์
- วัตถุประสงค์

วัตถุประสงค์

UI

features_list

1

UI

-
-
- UI
- JSON

2 JSON

```
"['20GB', ' ', ' ']"
```

UI JSON JSON JSON

- "100Mbps"
- API
-

Python

- UI
- JSON API/

UI

UI

1. 0000000000

0000000000000000000000000000000000

2. 00000000

00“000000”0000000000000000

- 00000000??
- 0000000000000000
- 00000000000000
- 000000000000
- 00000000

3. 0000

00000000000000000000000000

000 • 20GB 0000 • 00000000 • 0000000 • 000 • 30 000

000£15.00/ 000£0.00

[000] [000000]

4. 000000000000

0000000000000000000000000000

0000000000

terms 00000000000000000000

000 **UI** 00000000

1. 0000000000

00000000000000000000000000

2. 0000

000000000000000000000000

[]

3.

- 50
-
-

8 **Ansible**

```
{  
  "provisioning_play": "play_local_mobile_sim",  
  "provisioning_json_vars": "{  
    \"days\": 30,  
    \"data_gb\": 20,  
    \"voice_minutes\": \"unlimited\",  
    \"sms_count\": \"unlimited\"  
  }"  
}
```

- `provisioning_play` - Ansible `.yaml`
- `provisioning_json_vars` -
- `OmniCRM-API/Provisioners/plays/play_local_mobile_sim.yaml`

```
{
  "product_name": "SIM卡 20GB",
  "product_slug": "prepaid-mobile-20gb",
  "category": "standalone",
  "service_type": "mobile",
  "enabled": true,
  "icon": "fa-solid fa-sim-card",
  "comment": "SIM卡 20GB 预付费套餐",

  "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 流量',
    '预付费套餐',
    'SIM卡',
    '30 天'
  ]",
  "terms": "30 天预付费套餐",

  "provisioning_play": "play_local_mobile_sim",
  "provisioning_json_vars": "{
    \"days\": 30,
    \"data_gb\": 20,
    \"voice_minutes\": \"unlimited\",
    \"sms_count\": \"unlimited\"
  }
```

```
}"  
}
```

□□□□□□□□

□□□ □□□□□□□□□□□□

□□ **1**□□□□□□□□□□ **5GB** □□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□

```
{
  "product_name": "5GB 5GB",
  "product_slug": "5gb-data-boost",
  "category": "addon",
  "service_type": "mobile",
  "enabled": true,
  "icon": "fa-solid fa-plus",
  "comment": "5GB 5GB",

  "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 7 ",
  "terms": "7 ",

  "provisioning_play": "play_topup_charge_then_action",
  "provisioning_json_vars": "{
    \"data_gb\": 5,
    \"days\": 7
  }"
}
```

2

```

{
  "product_name": "WiFi 6 모뎀 렌탈",
  "product_slug": "wifi6-modem-rental",
  "category": "addon",
  "service_type": "internet",
  "enabled": true,
  "icon": "fa-solid fa-router",
  "comment": "WiFi 6 모뎀 렌탈 - 모뎀",

  "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) 2.4GHz + 5GHz 40MHz",
  "terms": "WiFi 6 모뎀 렌탈 £150",

  "provisioning_play": "play_addon_assign_modem",
  "provisioning_json_vars": "{
    \"device_type\": \"modem_router\",
    \"requires_configuration\": true
  }"
}

```

필드 설명

- `category: "addon"` - 모뎀 렌탈
- `contract_days: 0` - 30 - 모뎀
- `inventory_items_list: "[]"` - `['Rental Modem']` - 모뎀

- `auto_renew: "false"` 00000000 `"true"` 0000 - 0000
- `relies_on_list: ""` - 00000000000000 `service_type` 000

0000000000000000

0000000000 `inventory_items_list` 0000

1. 0000 - 0000000000000000
2. 0000 - 000000000000000000000000000000
3. 00 - 000000000000000000
4. 0000 - 0000000000000000
5. 00 - 00000000000000
6. 00 - 00000000000000

00000000000000

0000“WiFi 6 00000000”000000000000

1. 00000000 - 0000“00000000”
2. 00000000 - 0000000000
3. 0000 - 00 £10.00 0000
4. 00000000 - 00000000
 - `service_id` 0000000000
 - `customer_id` 00000000
 - `item_state` 0 "Assigned"
5. 0000 - 0000000000000000
6. 00 - 0000000000000000 ONT
7. 0000 - 0000 £10 00000000000000

00000000000000

0000000000000000

1. 0000 - 0000“00000000”
2. 00000000
 - 0000000000000000
 - 000000000000

- 14 日

3. 状態

- `item_state = "In Stock"`
- `item_state = "Damaged"`
- ...

4. 価格

- 14 日 £150
- `item_state = "Lost"`
- £450+ ...

...

...

- ...
- ...
- `contract_days` ...

2

... 20GB ... OmniCRM ... Ansible ...

...

... → ... → ... ↓ ↓ ... ← ... ← ... Ansible ... ↓ ↓ ... → OCS ...
 ... → ... → ...

...

1. ...

...

- ...
- ... 20GB
- ...

2. SIM

Requirements

- **SIM** - SIM card
◦ Label "SIM-00123 - ICCID: 8944..."
- **Phone** - Phone number
◦ Label "+44 7700 900123"

Implementation

3. Billing

Requirements

- Amount £0.00
- Amount £15.00
- Amount £15.00
- Duration 30

Implementation

30

4. API

API

- API "status=1" Provision
- API + +
- Ansible
- provision_id UI

5. UI

Requirements

UI

```
{
  "days": 30,
  "data_gb": 20,
  "voice_minutes": "unlimited",
  "sms_count": "unlimited"
}
```

□□□□□

```
{
  "product_id": 42,
  "customer_id": 123,
  "SIM Card": 5001,
  "Mobile Number": 5002
}
```

□□□□□

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "initiating_user": 7
}
```

□□□□□ **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. Ansible

play_local_mobile_sim.yaml

Ansible

SIM

OmniCRM-API/Provisioners/plays/play_local_mobile_sim.yaml

```
- name: SIM
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name:
      block:
        # 1.
        # 2. API
        # 3. API
        # 4. API
        # 5. OCS CGRateS
        # 6. OCS
        # 7. CRM
        # 8.
        # 9.
        # 10.

  rescue:
    #
    # - OCS
    # -
    # -
```

□□□□□□□□

□□ **1**□□□□□

```
- name: □□ crm_config □□□  
  ansible.builtin.include_vars:  
    file: "../..../crm_config.yaml"  
    name: crm_config
```

□□□□□□□□

- OCS/CGRateS URL □□□
- CRM □□ URL
- □□□□

□□ **2**□□□□□□□□

```
- name: □ 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
```

□□□□□□

- □□ GET /crm/product/product_id/42
- □□□□□□□□
- □□□ api_response_product □□□

□□□□ □□□□□□□□□□ provisioning_json_vars □□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□
- □□□□□□□□□□

- 000000000000

00 30000000

```
- name: 000000
  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
}}"
```

000000000000

00 4000000000

```
- name: 0 CRM API 00 SIM 00
  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: 0000000000 IMSI
  set_fact:
    imsi: "{{ api_response_sim.json.itemtext2 }}"
    iccid: "{{ api_response_sim.json.itemtext1 }}"
```

000000

- 00 SIM 000 ID 5001
- 00 SIM 00000
 - itemtext1 = ICCID0SIM 0000
 - itemtext2 = IMSI0000000
- 00

00000000

- IMSI 0000 HSS0000000000000000
- ICCID 0000000000000000
- 000000MSISDN000000000000

00 5000000 **UUID**

```

- name: 00 UUID 00
  set_fact:
    uuid: "{{ 99999999 | random | to_uuid }}"

- name: 0000 UUID
  set_fact:
    service_uuid: "Local_Mobile_SIM_{{ uuid[0:8] }}"

```

00000000

- 0000 UUID
- 0000 Local_Mobile_SIM_a3f2c1d8 0 service_uuid

0000

- 00 UUID 0 OCS/CGRateS 00000000
- 0000000000
- 0000000000000000

00 600 **OCS** 000000

```

- name: 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

```

Annotations

- CGRateS JSON-RPC API
- service_uuid
-
-

Variables

- OCS
-
-

7

```

- name: 20GB
  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"
          }
        }]
      }

```

20GB

- 20GB
- 21474836480 $20 * 1024 * 1024 * 1024$
- 720 30
- 10

8

```
- name: 000000
  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"
          }
        }]
      }
```

- 999,999,999 0000000000000000
- 30 0000

00 900 CRM 00000000

```

- name: 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

```

注意

- 服务UUID
- 服务UUID OCS
- 服务名称
- 服务名称“”
- 服务ID

10

```

- name: SIM
  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"
    }

```

11

- SIM
- service_id SIM
- "SIM"
-

11

- SIM
-
-

11

```

- name: 0000000000
  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 }} - 00",
        "description": "0000",
        "retail_cost": "{{ setup_cost }}",
        "wholesale_cost": "{{
api_response_product.json.wholesale_setup_cost }}"
      }

```

00000000

- 00 £0.00 0000000000000000
- 00 £1.00 000000
- 0000000000000000

00 **12**0000000000000000

```

rescue:
  - name: 删除OCS 套餐
    uri:
      url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
      method: POST
      body:
        {
          "method": "ApierV2.RemoveAccount",
          "params": [{
            "Account": "{{ service_uuid }}"
          }]
        }
  - name: 失败
    fail:
      msg: "删除OCS 套餐失败"

```

删除套餐

- 删除套餐
- 删除OCS 套餐
- 删除“套餐”
- 删除套餐

删除

- OCS 套餐
- 套餐
- 套餐

删除套餐

删除套餐

1. OCS 套餐CGRateS

- ID Local_Mobile_SIM_a3f2c1d8
- 套餐
 - 20GB 套餐 30 套餐

- 999M 30
- 999M 30

2. CRM

- ID 1234
- John Doe [customer_id: 123]
- 20GB [product_id: 42]
- "Mobile - +44 7700 900123"
- UUID [Local_Mobile_SIM_a3f2c1d8]
-
- £15.00 £5.00
- £10.00/

3.

- SIM 5001 1234 123
- 5002 1234 123

4.

-
-

5.

-
- 20GB
-
-
-

3

□□□□□□□□

□□□□□□ 20GB □□□□ 18GB□□□□□□“5GB □□□□”□□□□□

1. □□□□□□□

- □□“Mobile - +44 7700 900123”□□□□
- □□□□□□□□□□20GB □□ 18GB□90%□
- □□“□□□□□□”□“□□”

2. □□□□□□□□□□

□□□□□□□□□□

- `category = "addon"`
- `service_type = "mobile"` □□□□□□□□□□
- `residential = true` □□□□□□□□□□
- `enabled = true`

□□□□□“5GB □□□□ - £5.00”

3. □□□□□□□□

- □□“5GB □□□□”
- □□□□ £5.00
- □□□□□□□□

4. □□□□□□□□

□□□□ `play_topup_charge_then_action.yaml` □□□□□□□

```
{
  "product_id": 43, # 5GB □□□□□□
  "customer_id": 123,
  "service_id": 1234, # □□□□□
  "access_token": "eyJ...",
  "data_gb": 5, # □□ provisioning_json_vars
  "days": 7 # □□ provisioning_json_vars
}
```

□□□□□□□□□□

- □□ `service_id` □□□□□□□□
- □□□□□
- □□□□□□□□□□□□

□□□□□□□□□□□□

□□ **1**□□□□□□□□

```
- name: □ 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_fact:
    service_uuid: "{{ api_response_service.json.service_uuid }}"
    customer_id: "{{ api_response_service.json.customer_id }}"
```

□□□□

- □□ `service_uuid` □□□□□ OCS □□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□

□□ **2**□□□□□□

```

- name: 创建支付方法
  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: 设置支付方法 ID
  set_fact:
    payment_method_id: "{{ api_response_payment_method.json.data.payment_method_id }}"

- name: 创建支付
  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 存储空间",
          "service_id": "{{ service_id | int }}",
          "product_id": "{{ product_id | int }}",
          "invoice": true
        }
      }
  register: charge_response

- name: 断言支付成功
  assert:
    that:
      - charge_response.json.success == true

```

创建支付

- Stripe 5GB
- £5.00
- £1.50
- Stripe
- Stripe
- Stripe

Stripe

- Stripe
- Stripe
- Stripe

3 OCS

```

- name: Stripe 5GB
  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
          }
        }]
      }

```

Stripe

- Stripe 5GB 5368709120
- 168 7

- 20년 - 20년

20년

- 2GB 25
- 5GB 7
- 7GB
- 20년

4년

```
- name: 20년
  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",
        "description": "5GB 7",
        "retail_cost": 5.00,
        "wholesale_cost": 1.50
      }
}
```

20년

- £5.00
- £1.50
- 20년

20년

1. 20년
2. 20년
3. 20년 OCS

4. CRM

5. 7GB

- £15 £5
- £5 £1.50

```

{
  "product_name": "10GB",
  "category": "addon",
  "retail_cost": 10.00,
  "contract_days": 30,
  "auto_renew": "true",
  "provisioning_play": "play_topup_charge_then_action"
}
  
```

```

- name: []
  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
        }]
      }

```

[]

- [] OCS []
- [] 30 [] [] Action_{{ product_slug }}
- []
- []

[]

- []“[]2025 [] 2 [] 1 [] - £10.00”
- []“[]”[]
- []“[]”[]

Part 4

XX

XXXXXXXXXX

XX

1. XXXXX - XXXXX"XXXXX"
2. XXXXX - XXXXXXXXXXXXXXXXXX
3. XXXX - XXXXXXXXXXXXXXX
4. XXXXX - XXXXXXXXXXXXXXX

XXXXXXXXXX

1. XXXXXXXX

- XXXXXXX
- XX"XXXXX"
- XXXXXXX"XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
- XXXXX

2. XXXXXXXX

XX

- XXXXXXX"XXXX"
- 7-30 XXXXXXXXXXXXXXX
- XXXXXXXXXXXXXXXXXXXXXXX
- XXXXXXXXXXXXXXX

3. XXXXXXXX

XX

```
{
  "action": "deprovision",
  "service_id": 1234,
  "customer_id": 123,
  "service_uuid": "Local_Mobile_SIM_a3f2c1d8"
}
```

[[service.deprovisioning_play]]

4. Ansible []

```
- name: 设置OCS
hosts: localhost
tasks:
  - name: 设置OCS
    uri:
      url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
      method: POST
      body:
        {
          "method": "ApierV2.SetAccount",
          "params": [{
            "Account": "{{ service_uuid }}",
            "ExtraOptions": { "Disabled": true }
          }]
        }

  - name: 删除行动计划
    uri:
      url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
      method: POST
      body:
        {
          "method": "ApierV1.RemoveActionPlan",
          "params": [{
            "Id": "ServiceID_{{ service_uuid }}__*"
          }]
        }

  - name: 设置CRM服务状态
    uri:
      url: "http://localhost:5000/crm/service/{{ service_id }}"
      method: PATCH
      body:
        {
          "service_status": "Deactivated",
          "service_deactivate_date": "{{ current_datetime }}"
        }

  - name: 设置SIM卡状态
    uri:
      url: "http://localhost:5000/crm/inventory/inventory_id/{{
sim_card_id }}"
      method: PATCH
```

```
body:
  {
    "service_id": null,
    "customer_id": null,
    "item_state": "Decommissioned"
  }
```

□□□□□□

1. □□ **OCS** □□ - □□□□□□□□□□□□
2. □□□□□□ - □□□□□□
3. □□ **CRM** □□ - □□□“□□□”□□□□□□
4. □□□□ - SIM □□□“□□□”□□□□□□□□□□□□

5. □□□□□□

□□□□□□□□

- □□□□□□□□□□□□□□□□
- □□□□□□ CRM □□□□□□□□
- □□□□□□□□□□□□
- □□□□□□□□□□
- OCS □□□□□□□□□□□□

□□□□□□□□□□□□

□□□□□□□□□□□□

- □□□□□□□□□□□□
- □□ OCS □□□□□□□□
- □□□□□□
- □□□□□□□□□□□□

```
- name: 0000
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body:
      {
        "method": "ApierV2.SetAccount",
        "params": [{
          "Account": "{{ service_uuid }}",
          "ExtraOptions": { "Disabled": true }
        }]
      }
}
```

0000000000000000

- 00000000
- 0000 OCS 00
- 0000
- 00000000

```
- name: 00 OCS 00
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body:
      {
        "method": "ApierV2.RemoveAccount",
        "params": [{
          "Account": "{{ service_uuid }}"
        }]
      }
}
```

0000000000000000

000000000000

000000

- `enabled: true` - 開
- `enabled: false` - 閉

設定

- 設定 `enabled: false` 状態
- 設定
- 設定/設定
- 設定

状態

状態

- `New` - 新
- `In Stock` - 在庫
- `Assigned` - 指定
- `Decommissioned` - 廃止
- `Damaged` - 破損

状態

状態

- SIM 状態“状態”
- 状態30 状態
- 状態“状態”

状態

状態

- 状態
- 状態
- 状態
- 状態

API Parameters

OmniCRM API **10** Parameters for Mailjet Integration

- **from_email** - Sender email address
- **from_name** - Sender name
- **template_id** - Mailjet template ID
- **subject** - Email subject

Example

Request

Response

```
api_crmCommunicationCustomerWelcome:  
  from_email: "support@yourcompany.com"  
  from_name: "Your Company Support"  
  template_id: 5977509  
  subject: "Welcome to our company!"
```

Variables

- Customer name
- Customer email
- Company name

URLs


- `{{ var:customer_name }}` - Customer name
- `{{ var:email }}` - Customer email
- `{{ var:company_name }}` - Company name
- `{{ var:login_url }}` - Login URL
- `{{ var:support_url }}` - Support URL

Footer

□□□□□□□□□□□□□□

```
api_crmCommunicationCustomerInvoice:  
  from_email: "billing@yourcompany.com"  
  from_name: "Your Company Billing"  
  template_id: 6759851  
  subject: "□□□□ - "
```

□□□□□

- □□□□□□□□□□□□
- □□□□□□□□
- □□□□□□□□□□

□□□□□□□□

- `{{ var:customer_name }}` - □□□□□□
- `{{ var:invoice_number }}` - □□ ID/□□
- `{{ var:invoice_date }}` - □□□□□□
- `{{ var:due_date }}` - □□□□□□
- `{{ var:total_amount }}` - □□□□□□
- `{{ var:invoice_url }}` - □□/□□□□ PDF □□□□
- `{{ var:pay_url }}` - □□□□□□□□□□

□□□□□□

□□ PDF □□□□□□□□□□□□□□

□□□□□□□

□□□□□□□□□□□□□□

```
api_crmCommunicationCustomerInvoiceReminder:  
  from_email: "billing@yourcompany.com"  
  from_name: "Your Company Billing"  
  template_id: 6759852  
  subject: "□□□□ - □□□□"
```

□□□□

- □□□□ X □□□□□□
- □□□□□□□□
- □□□□□□□□□□□□□□

□□□□□□□

- `{{ var:customer_name }}`
- `{{ var:invoice_number }}`
- `{{ var:due_date }}`
- `{{ var:days_overdue }}`
- `{{ var:total_amount }}`
- `{{ var:pay_url }}`

□□□□□□□□

□□□□□□□□□□□□□□

```
api_crmCommunicationUserWelcome:  
  from_email: "admin@yourcompany.com"  
  from_name: "Your Company Admin"  
  template_id: 5977510  
  subject: "□□□□□□"
```

□□□□□

- □□□□□□□□□□
- □□□□□□□□□□"□□□□□□"□□

□□□□□□□

- `{{ var:user_name }}` - □□□□□□□□
- `{{ var:email }}` - □□□□□□□□□□
- `{{ var:role }}` - □□□□□□
- `{{ var:login_url }}` - □□□□□□□□□□
- `{{ var:temp_password }}` - □□□□□□□□□□

- `{{ var:support_email }}` - IT 部門

メール

送信先

```
api_crmCommunicationUserPasswordReset:  
  from_email: "noreply@yourcompany.com"  
  from_name: "Your Company Security"  
  template_id: 5977511  
  subject: "パスワードリセット"
```

メール

- 送信先名: "パスワードリセット"
- 送信先アドレス

メール

- `{{ var:user_name }}`
- `{{ var:reset_url }}` - リセット URL (1 時間有効)
- `{{ var:expiry_time }}` - 有効期限

メール

送信先

送信先

送信先

```
api_crmCommunicationUserPasswordResetSuccess:  
  from_email: "noreply@yourcompany.com"  
  from_name: "Your Company Security"  
  template_id: 5977512  
  subject: "パスワードリセット完了"
```

メール

- 0000000000
- 0000000000

00000000

- {{ var:user_name }}
- {{ var:change_date }} - 00000000/00
- {{ var:ip_address }} - 000 IP 00000000
- {{ var:support_email }} - 000000000000

00000000

00000000000000000000

```
api_crmCommunicationUserPasswordChange:  
  from_email: "noreply@yourcompany.com"  
  from_name: "Your Company Security"  
  template_id: 5977513  
  subject: "00000000"
```

000000

- 00000000/00000000
- 0000000000

00000000

- {{ var:user_name }}
- {{ var:change_date }}
- {{ var:changed_by }} - "00"00000000
- {{ var:support_email }}

00000000

00000000000000000000

```
api_crmCommunicationEmailVerification:
  from_email: "noreply@yourcompany.com"
  from_name: "XXXX"
  template_id: 5977514
  subject: "XXXXXXXXXXXX"
```

XXXX

- XXXXXXXXXXXXXXX
- XXXXXXXXXXXXXXX
- XXXXXXXXXXXXXXX

XXXXXXXX

- {{ var:user_name }}
- {{ var:verification_url }} - XXXXXXXXXXXXXXX
- {{ var:verification_code }} - XXXXXXXXXXXXXXX

XXXXXX

XXXXXXXXXX/XXXXXXXXXX

```
api_crmCommunicationsBalanceExpired:
  from_email: "support@yourcompany.com"
  from_name: "Your Company Support"
  template_id: 5977515
  subject: "XXXXXXXXXXXX"
```

XXXX

- XXXXXXX
- XXXXXXXXXXXXXXX
- XXXXXXX

XXXXXXXX

- {{ var:customer_name }}
- {{ var:service_name }} - XXXXXXX

- `{{ var:expiry_date }}`
- `{{ var:balance_type }}` - “”“”“”“”
- `{{ var:renewal_url }}` - /

```
api_crmCommunicationsBalanceLow:
  from_email: "support@yourcompany.com"
  from_name: "Your Company Support"
  template_id: 5977516
  subject: ""
```

- 20%
- OCS
-

- `{{ var:customer_name }}`
- `{{ var:service_name }}`
- `{{ var:current_balance }}`
- `{{ var:threshold }}`
- `{{ var:balance_type }}`
- `{{ var:topup_url }}` -

Mailjet

Mailjet

1 Mailjet

1. Mailjet

2. 200 OK → 204 No Content
3. 200 OK
4. 200 OK (with headers) → 204 No Content

2. Mailjet

Mailjet HTML

- -
- - `{{ var:customer_name }}` `{{ var:user_name }}`
- -
- -
- -
- -

```

<!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>○○○○○○</h2>

  <p>○○ {{ var:user_name }}</p>

  <p>○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○</p>

  <a href="{{ var:reset_url }}" class="button">○○○○</a>

  <p>○○○○ {{ var:expiry_time }} ○○○</p>

  <p>○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○</p>

  <hr>
  <p style="font-size: 12px; color: #666;">
    ○○○ | support@yourcompany.com<br/>
    123 ○○○○○○○○○
  </p>
</body>
</html>

```

3 ○○○○○ ID

1. ○ Mailjet ○○○○
2. ○○ ○○ ID○○○○○○ 5977509
3. ○○ ID ○○○ crm_config.yaml

4. Mailjet Configuration

1. Mailjet API Key and Secret
2. Mailjet Account
3. Mailjet Template
4. Mailjet User

5. OmniCRM Configuration

Configuration file: `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: "Security Alert"
```

Restart OmniCRM API

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

6. Testing

OmniCRM configuration is correct. Mailjet

Configuration

- Configuration
- Configuration
- Configuration
- Configuration

- 環境変数

環境変数

環境変数の設定

- 環境変数
- 設定/取得
- 環境変数の名前

環境変数

環境変数の設定

- 環境変数 API キー Mailjet 環境変数 ID
- 環境変数
 - `crm_config.yaml` の `api_key` と `api_secret`
 - Mailjet 環境変数
 - 環境変数 ID 環境変数 Mailjet
 - API キー

環境変数の取得

- 環境変数 OmniCRM 環境変数
- 環境変数
 - 環境変数の名前
 - `{{ var:variable_name }}`
 - OmniCRM 環境変数 API キー
 - Mailjet 環境変数

PDF 環境

- PDF 環境
- 環境変数
 - 環境変数
 - 環境変数の名前

- PDF 15MB Mailjet
- PDF

Mailjet

- API
- - Mailjet API 200
 - OmniCRM-API
 -
 -

- `administration_configuration` - Mailjet
- `payments_invoices` -
- `authentication_flows` -
- `customer_care` -
- - Mailjet API

- Mailjet <<https://dev.mailjet.com/>>
- Mailjet API <<https://dev.mailjet.com/email/reference/>>



OmniCRM 50

API

OmniCRM API を Prometheus で監視するための Prometheus scrape config を提供します。
/crm/metrics を Prometheus が scrape できるように設定します。

URL

監視対象 URL

- `http://your-omnicrm-api:5000/crm/metrics`
- `https://your-omnicrm-api:5000/crm/metrics`
- `http://your-omnicrm-api:5000/crm/metrics` - OCS/CGRateS, Stripe, Mailjet API
- `https://your-omnicrm-api:5000/crm/metrics`
- **HTTP** プロキシ - API を監視するためのプロキシ

設定

URL: `http://your-omnicrm-api:5000/crm/metrics`

例: Prometheus scrape config

例: Prometheus scrape config (HTTPS)

監視

1. Ansible

Ansible を使用して Prometheus scrape config を自動的に生成するための [Provisioning System](#) `Ansible Playbooks`

omnicrm_provision_jobs_total

例: `omnicrm_provision_jobs_total`

📄:

- `status` - 📄📄📄📄: `success`, `failed`, `running`

📄: 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

omnicrm_provision_job_duration_seconds

📄: 📄📄

📄:

- `playbook` - 📄📄 Ansible 📄📄📄
- `status` - 📄📄📄📄: `success`, `failed`

📄: `[1, 5, 10, 30, 60, 120, 180, 300, 600]` 📄📄1📄📄10📄📄

📄: 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

omnicrm_provision_jobs_active

📄: 📄📄

📄: 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

omnicrm_provision_tasks_total

📄: 📄📄

📄:

- `playbook` - Ansible 📄📄📄
- `status` - 📄📄📄: `ok`, `failed`, `ignored`

📄: 📄📄📄📄📄 Ansible 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

omnicrm_provision_errors_total

📄: 📄📄

📄:

- `error_type` - 可能: `fatal`, `task_failed`, `timeout`
- `playbook` - Ansible 名称

例: 可能: `fatal`, `task_failed`, `timeout`

2. 可能

可能: `omnicrm_db_query_duration_seconds` `omnicrm_db_pool_size` `omnicrm_db_pool_overflow` `omnicrm_db_pool_connections_checked_out`
[System Architecture](#)

`omnicrm_db_query_duration_seconds`

例: 可能

例:

- `operation` - SQL 可能: `SELECT`, `INSERT`, `UPDATE`, `DELETE`
- `table` - 可能

例: `[0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0, 5.0, 10.0]`

例: 可能: `omnicrm_db_query_duration_seconds` `omnicrm_db_pool_size` `omnicrm_db_pool_overflow` `omnicrm_db_pool_connections_checked_out`

`omnicrm_db_pool_size`

例: 可能

例: 可能: `omnicrm_db_pool_size` `omnicrm_db_pool_overflow` `omnicrm_db_pool_connections_checked_out`

`omnicrm_db_pool_overflow`

例: 可能

例: 可能: `omnicrm_db_pool_size` `omnicrm_db_pool_overflow` `omnicrm_db_pool_connections_checked_out`

`omnicrm_db_pool_connections_checked_out`

例: 可能

例: 可能: `omnicrm_db_pool_size` `omnicrm_db_pool_overflow` `omnicrm_db_pool_connections_checked_out`

omnicrm_db_errors_total

概要: 概要

詳細:

- `error_type` - エラータイプ
- `operation` - 操作タイプ: `connection_error`, `query_error`, ...

概要: 概要

概要: 概要

3. OCS/CGRateS 概要

OCS/CGRateS 概要 [Billing Overview](#) [CGRateS Integration](#)

omnicrm_ocs_api_calls_total

概要: 概要

詳細:

- `method` - OCS API 呼び出し: `GetBalance`, `SetBalance`, `SetAccount`, `RemoveAccount`, ...
- `status` - ステータス: `success`, `failed`

概要: OCS/CGRateS API 呼び出し API 呼び出し OCS 概要

omnicrm_ocs_api_duration_seconds

概要: 概要

詳細:

- `method` - OCS API 呼び出し: `GetBalance`, `SetBalance`, `GetAccount`, ...

概要: `[0.1, 0.25, 0.5, 1.0, 2.0, 5.0, 10.0, 30.0]`

Table: OCS/CGRateS API Error Summary Table API Error Summary Table

omnicrm_ocs_api_errors_total

Table: Table

Table:

- `method` - OCS API Method
- `error_type` - Error Type: `timeout`, `connection_error`, `json_error`, `http_error`, ...

Table: OCS/CGRateS API OCS API Error Summary Table

omnicrm_ocs_balance_queries_total

Table: Table

Table:

- `query_type` - Query Type: `single_account`, `multiple_accounts`

Table: OCS Error Summary Table

omnicrm_ocs_action_plan_operations_total

Table: Table

Table:

- `operation` - Operation: `create`, `remove`, `query`

Table: CGRateS Error Summary Table

4. Stripe Table

Stripe [Payment System Guide](#) [Payment Methods](#)

omnicrm_stripe_api_calls_total

type: string

type:

- operation - Stripe operation: create_customer, charge, refund, update_payment_method, ...
- status - status: success, failed

type: Stripe API ...

omnicrm_stripe_api_duration_seconds

type: string

type:

- operation - Stripe ...

type: [0.1, 0.5, 1.0, 2.0, 5.0, 10.0, 30.0]

type: Stripe API ...

omnicrm_stripe_api_errors_total

type: string

type:

- operation - Stripe ...
- error_type - error_type: card_declined, network_error, api_error, ...

type: Stripe API ...

omnicrm_stripe_payment_amount_cents

type: string

type:

- payment_type - payment_type: charge, refund

📄: Stripe 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

omnicrm_stripe_payment_failures_total

📄: 📄📄

📄:

- `reason` - 📄📄📄: `card_declined`, `insufficient_funds`, `expired_card`, 📄📄

📄: 📄📄📄📄📄 Stripe 📄📄📄📄📄

5. Mailjet 📄📄📄

Mailjet 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄 [Mailjet Integration](#) 📄

omnicrm_mailjet_api_calls_total

📄: 📄📄

📄:

- `email_type` - 📄📄📄📄: `welcome`, `user_welcome`, `invoice`, `notification`
- `status` - 📄📄📄: `success`, `failed`

📄: Mailjet API 📄📄📄📄📄 `@track_mailjet_call` 📄📄📄📄

omnicrm_mailjet_api_duration_seconds

📄: 📄📄

📄:

- `email_type` - 📄📄📄📄

📄: `[0.1, 0.5, 1.0, 2.0, 5.0, 10.0, 30.0]` 📄

📄: Mailjet API 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄 Mailjet API 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

omnicrm_mailjet_api_errors_total

表名: 表名

表名:

- email_type - 表名
- error_type - 表名

表名: Mailjet API 表名

omnicrm_mailjet_emails_sent_total

表名: 表名

表名:

- email_type - 表名
- template_id - Mailjet 表名 ID

表名: 表名 Mailjet 表名 API 表名

omnicrm_mailjet_email_recipients_total

表名: 表名

表名:

- email_type - 表名

表名: 表名

6. 表名

表名 Provisioning System

omnicrm_background_jobs_total

表名: 表名

表名:

- `job_type` - 可能値: `playbook_single`, `playbook_chain`, `periodic_task`

型: `BackgroundJobTimer`

omnicrm_background_jobs_active

型: 整数

型:

- `job_type` - 可能値

型: 整数

omnicrm_background_job_duration_seconds

型: 整数

型:

- `job_type` - 可能値
- `status` - 可能値: `success`, `failed`

型: `[1, 5, 10, 30, 60, 120, 180, 300, 600, 1800, 3600]`

型: 整数

omnicrm_background_job_errors_total

型: 整数

型:

- `job_type` - 可能値
- `error_type` - 可能値

型: 整数

7. Flask HTTP 메트릭

이 문서는 `prometheus-flask-exporter`가 제공하는 API의 HTTP 메트릭

`flask_http_request_duration_seconds`

타입: `gauge`

메트릭:

- `method` - HTTP 메서드: `GET`, `POST`, `PUT`, `DELETE`, ...
- `endpoint` - Flask 엔드포인트
- `status` - HTTP 상태

이 메트릭은 API의 HTTP 요청 처리 시간을 나타냅니다.

`flask_http_request_total`

타입: `counter`

메트릭:

- `method` - HTTP 메서드
- `endpoint` - Flask 엔드포인트
- `status` - HTTP 상태

이 메트릭은 HTTP 요청의 총 횟수를 나타냅니다.

`flask_http_request_exceptions_total`

타입: `counter`

메트릭:

- `method` - HTTP 메서드
- `endpoint` - Flask 엔드포인트

이 메트릭은 HTTP 요청 중 예외가 발생한 횟수를 나타냅니다.

8. API `omnicrm_api_errors_total`

API `omnicrm_api_errors_total` returns the total number of API errors.

`omnicrm_api_errors_total`

URI: `/api/errors`

Method: `GET`

- `endpoint` - API endpoint
- `error_type` - Error type
- `status_code` - HTTP status code

Response: `JSON`

`omnicrm_api_auth_failures_total`

URI: `/api/auth`

Method: `GET`

- `auth_method` - Authentication method: `jwt`, `api_key`, `ip_whitelist`
- `reason` - Reason for failure

Response: `JSON`

9. API `omnicrm_api`

`omnicrm_api`

URI: `/api`

Method: `GET`

- `version` - API version
- `environment` - Environment: `production`, `staging`, `development`

Response: OmniCRM API version and environment information

□□□□

update_db_pool_metrics(engine)

□ 30 □□□□□□□□□□□□□□□□□□

Prometheus □□

□□□□

□ OmniCRM □□□□ `prometheus.yml` □

```
scrape_configs:  
  - job_name: 'omnicrm-api'  
    scrape_interval: 15s  
    scrape_timeout: 10s  
    static_configs:  
      - targets: ['omnicrm-api:5000']  
    metrics_path: '/crm/metrics'
```

□□□□

□□□□□□

```

- 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: "High provision failure rate"
    description: "{{ $value | humanizePercentage }} high provision failure rate"

```

High provision failure rate

```

- alert: DatabasePoolExhausted
  expr: omnicrm_db_pool_overflow > 0
  for: 2m
  labels:
    severity: critical
  annotations:
    summary: "Database pool exhausted"
    description: "Database pool exhausted"

```

Database pool exhausted

```

- alert: SlowDatabaseQueries
  expr: |
    histogram_quantile(0.99,
      rate(omnicrm_db_query_duration_seconds_bucket[5m])
    ) > 1.0
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "Slow database queries"
    description: "99th percentile of database query duration is {{ $value }}s"

```

OCS API

```

- 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: "OCS API 🚨🚨🚨🚨"
    description: "{{ $value | humanizePercentage }} 🚨 OCS API 🚨🚨🚨🚨"

```

Stripe 🚨🚨🚨

```

- alert: StripePaymentFailures
  expr: rate(omnicrm_stripe_payment_failures_total[5m]) > 5
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "Stripe 🚨🚨🚨🚨🚨"
    description: "{{ $value }}" 🚨🚨🚨🚨🚨

```

🚨🚨🚨🚨

🚨🚨🚨🚨

- 🚨🚨🚨🚨 - 🚨🚨🚨🚨🚨🚨🚨
 - 🚨🚨🚨🚨 > 10%
 - 🚨🚨🚨🚨🚨🚨🚨
 - OCS/CGRateS API 🚨🚨
 - Stripe 🚨🚨🚨🚨🚨

2. Prometheus - KPI

- Stripe KPI
- KPI
- KPI

3. Prometheus - KPI

- API KPI
- KPI
- API KPI

4. Prometheus - KPI

- KPI
- KPI
- API KPI

API

Prometheus KPI:

- 15 - KPI 15 KPI
- 90 - KPI 5 KPI
- 1 - KPI 1 KPI

API Thanos Cortex VictoriaMetrics KPI

API

API

API /metrics KPI nginx KPI

API:

```
curl http://localhost:5000/crm/metrics
```

📄 Prometheus 配置文件:

```
# 📄 Prometheus 配置  
http://prometheus:9090/targets
```

📄 配置

📄 配置

- 📄 配置
- Stripe 📄 配置
- OCS 📄 配置

📄 配置

📄 Prometheus 配置

```
# 📄 配置  
count by (__name__) ({__name__=~".+"})
```

📄 >10,000 📄 配置

📄 配置

📄: 34📄31 📄 + 3 📄 Flask 📄

📄:

- 📄: 20
- 📄: 5
- 📄: 8

- 1

1:

- 5
 - 5
 - OCS/CGRateS: 5
 - Stripe: 5
 - Mailjet: 5
 - : 4
 - HTTP/Flask: 3
 - : 1
 - : 2
-

1

- [System Architecture](#) -
- [Provisioning System](#) -
- [Ansible Playbooks](#) -
- [API Documentation](#) - API
- [Billing Overview](#) -
- [Payment System Guide](#) - Stripe
- [Mailjet Integration](#) -
- [Administration Configuration](#) -



OmniCRM [Payment System Guide <payment_system_guide>](#) [Billing Overview <billing_overview>](#) [Payment Processing <payments_process>](#) [Invoices <payments_invoices>](#)

OmniCRM [Payment System Guide <payment_system_guide>](#) [Billing Overview <billing_overview>](#) [Payment Processing <payments_process>](#) [Invoices <payments_invoices>](#)



Payment System Guide

- [Payment System Guide <payment_system_guide>](#)
- [Billing Overview <billing_overview>](#)
- [Payment Processing <payments_process>](#)
- [Invoices <payments_invoices>](#)
- [Self-Care Portal <self_care_portal>](#)
- [Payment System Guide <payment_system_guide>](#)

Payment System Guide

- [Payment System Guide <payment_system_guide>](#)
 - [Payment System Guide <payment_system_guide>](#)
 - [Payment System Guide <payment_system_guide>](#)
 - [Payment System Guide <payment_system_guide>](#)
- **PayPal** [Payment System Guide <payment_system_guide>](#)

Payment System Guide

Payment System Guide

- [Payment System Guide <payment_system_guide>](#)
- [Payment System Guide <payment_system_guide>](#)

- 00000
- 0000/00
- 00000/00
- 00000000000000

00000000

000000

1. 000 **Customers** → [00000]
2. 00 **Billing** 00
3. 000 **Payment Methods** 00

00000

00000000000000

0000000000000000

00000000000000 60 000000000000

00000000

0000000000000000

□□□□

- □□ - □□□□□□□□□□“□□□”□“□□ Visa”□
- □□□ - □□□□□□□□□□
- □□ - □□□□/□□□□MM/YYYY □□□□
- □□ - □□□□□□□□□□
- □□ - □□□□□□□□□□□□□□□□□□

□□□□□□□□

□□□□□□□□□□□□ : □□□□□□□□

- □□□□ - □□□□□□□□□□□□□□
- □□ - □□□□□□□□□□

□□□□□□□□

□□ **"Add Payment Method"** □□□□□□□□□□□□

□□□□□□□□□□□□□□□□

□□□□□□□□□□□ Stripe Elements □ PayPal SDK □□□□□□□□

□□□□□□

- 請求書 - 請求額・CVV Stripe 決済
- 請求書 - 請求額
- 請求/入金 - 請求額

請求書

- 請求書 - 請求額

決済

- 請求書の決済処理 (iframe Stripe Elements / PayPal SDK)
- OmniCRM 請求書の決済
- PCI DSS 請求書の決済
- 請求書の決済

請求書の決済

請求書 "Add Payment Method" 決済

1. 請求書の決済
 - 請求書の決済
 - 請求書の決済
 - 請求 CVC 決済
2. 請求書の決済
 - 請求書の決済 OmniCRM
 - 請求書の決済 (pm_1A2B3C4D) Stripe
 - 請求書の決済 OmniCRM
3. 請求書の決済
 - OmniCRM 請求書の決済
 - 請求書の決済
 - 請求書の決済 OmniCRM 決済

請求書の決済

請求書の決済

請求書 | Visa 請求額 1234

□□□□□□□□□□□□

□□□□□□

- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□

□□□□□□□□

□□□□□□□□

- □□□□□□□□
- □□□□
- □□□□□
- □□□□□□□□□□□□

□□□□□□□□

1. □□□□□□□□□□□□□□□□
2. □□□□□□□ □□□□□ : □
3. □□ **"Set as Default"**
4. □□□□□□

Visa □□ 5678 □□□□□□□□□□□□

□□□□□□□□□□□□

□□□□□□

□□□□□□□□

□□□□□□□□□□□□□□□□

刪除付款方法

請按照以下步驟操作

步驟一

- 進入「我的帳戶」
- 點選「付款方法」
- 點選 **"Delete"**

步驟二

請確認

刪除付款方法

刪除 Visa 卡 1234 有效期至 12/2026

⚠️ 刪除後將無法用於付款

[刪除] [取消]

刪除 "Delete Payment Method" 後

注意事項

請注意

刪除付款方法後

請注意

- 刪除付款方法後，將無法用於付款
- 刪除付款方法後，將無法用於付款
- 刪除付款方法後，將無法用於付款

XXXXXXXXXX

OmniCRM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXX

XXX **Billing** → **Expiring Cards** XXXXXXXXXXXXXXX

XX XX XX XXXXXXX John Smith Visa ****1234 02/2025 12** X XX **Acme Corp**
MC5678 03/2025 45 X XX **Jane Doe Amex**9012 01/2025** XXX XX

🔗🔗 XXX

- XXXX - XXXX 30/60/90 XXXXX
- XXXX - XXXXX
- XXXX - XXXXXXXXXXXXXXX

XXX

- XX - XXXXXXXXXXXXXXXXXXXXXXX
- XX - XXXXXXXXXXXXXXXXXXXXXXX Mailjet

□□□□

□□□□ Mailjet □□□□□□□□□□

- □□□ **60** □ - □□□□□
- □□□ **30** □ - □□□□□
- □□□ **7** □ - □□□□
- □□□ - □□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□

Mailjet □□□□□

- □□□□□
- □□□□□□□□□□
- □□□□□
- □□□□□□□□□□□□□□

□□□ `integrations_mailjet` □□□□□□□□□□□□□□□□

□□□□□□□□□□□□

□□□□□□□□

1. □□□□□□□□□□□□□□□□
2. □□□□□□□□□□□□
3. □□□ **Billing → Payment Methods**
4. □□ **"Add Payment Method"**
5. □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
6. □□□□□□□□□□
7. □□□□□/□□□□□□□

□□□□□□□□□□

□□□□□□□□□□□□

□ **1-3** □□□□□□

- □□□□□□
- □□□□□□□□□□□□
- □□□□□□□□□□□□

□ **4-7** □□

- □□□□□□□□□□□□
- □□□□□□□□
- □□□□□□□□□□

□ **8-14** □□

- □□□□□□□□□□
- □□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□

□ **15** □□□□□

- □□□□□□□□□□
- □□□SIM □□□□□□□□□□
- □□□□□□
- □□□□□□□□□□□□

□□□□□□□□

□□□□□□□□

- □□□□ **30** □ □□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□
- □□□□□□□□□□□□□□

□□□□□□□□

□□□□□□□□□□□□

1. □□□□□□□□□□

2. 商家
3. 商家代辦商
4. 商家代辦商代辦商
5. 商家代辦商代辦商代辦商

商家代辦商

商家

OmniCRM 商家代辦商

1. 商家代辦商 → 商家代辦商代辦商
2. 商家代辦商代辦商 → 商家代辦商
3. 商家代辦商 **OmniCRM** → 商家代辦商
4. 商家代辦商 → 商家代辦商代辦商代辦商

OmniCRM 商家代辦商

```
{
  "vendor": "stripe",
  "vendor_payment_method_id": "pm_1A2B3C4D5E6F",
  "payment_type": "card",
  "brand": "visa",
  "last4": "1234",
  "exp_month": 12,
  "exp_year": 2026,
  "name": "John Smith",
  "nickname": "Personal Card",
  "is_default": true
}
```

OmniCRM 商家代辦商

- 商家代辦商
- CVV/CVC 碼
- 商家代辦商
- PIN 碼

PCI 概要

概要

- PCI 概要 - OmniCRM 概要
- 概要 - iframe
- 概要 - 概要
- 概要 - HTTPS/TLS

Payment System Guide <payment_system_guide> 概要

概要

1. 概要

概要

1. 概要
2. 概要
3. 概要
4. Stripe 概要
5. 概要
6. 概要
7. 概要
8. 概要

2. 概要

概要

1. 概要 60
2. 概要
3. Billing → Payment Methods
4. 概要 12/2025
5. "Add Payment Method"

6. 0000 12/2028 0000
7. 00000000
8. 0000
9. 00000000

0000 **3**00000000000000

000 00000000“000000”

1. 00000000
2. 00000000000000
3. 00000000000000 01/2025
4. 000“00000000000000”
5. 00000000000000
6. 00000 **Customers → [Customer] → Billing**
7. 00 **"Add Payment Method"**
8. 00000000000000
9. 00000000
10. 00000000
11. 00000000
12. 00000000“0000000000”

0000 **4**00000000000000

000 0000000000000000

1. 0000000000 Visa 00 1111
2. 0000000000000000
3. 000000 Mastercard 00 2222000000
4. 000000 Amex 00 333300000000
5. 0000000000000000 Mastercard
6. 000 Visa 0000000000

📅 5 📅

📅

1. 📅 **Billing → Expiring Cards**
2. 📅 “📅 30 📅”
3. 📅 15 📅
4. 📅 → **"Send Reminder Emails"**
5. Mailjet 📅
6. 📅
7. 📅
8. 📅
9. 📅

📅

“📅”

- 📅 📅
- 📅
 - 📅
 - 📅
 - 📅
 - 📅

“📅”📅

- 📅 📅 API 📅
- 📅
 - 📅
 - 📅
 - 📅
 - 📅
 - 📅

API API API API API

- API API API API API API
- API
 - API API API
 - API API API API API API

API API API API API API API

- API API API API API
- API
 - API
 - API API API API API API
 - API API API API API

API API

- API API API API
- API
 - API API API
 - API API API API
 - API API API API

API API API

- API API API SDK API API API API
- API
 - API API API API
 - API API API API
 - API API API API API
 - API API SDK API API API
 - API API/ API API

API API API

- API Mailjet API API API
- API

- 編輯 crm_config.yaml 編輯 Mailjet 編輯
- 編輯
- 編輯
- 編輯 Mailjet 編輯

編輯

編輯

- 編輯
- 編輯 2 編輯
- 編輯 30 編輯
- 編輯/編輯
- 編輯“**Visa**”“**Amex**”
- 編輯

編輯

- 編輯
- 編輯/SMS/編輯
- 編輯
- 編輯
- 編輯
- 編輯 £0.01 編輯

編輯

- 編輯
- 編輯 60/30/7 編輯
- 編輯/編輯
- 編輯 Mailjet 編輯
- 編輯
- 編輯

□□□□

- `Payment System Guide <payment_system_guide>` - □□□□ API □□□□
- `payments_process` - □□□□□□□□□□
- `payments_invoices` - □□□□□□□□□□
- `features_topup_recharge` - □□□□□□□□□□
- `basics_payment` - □□□□□□□□
- `customer_care` - □□□□□□□□□□□□□□

OmniCRM API

API

OmniCRM API integrates with various payment gateways including Stripe, PayPal, Square, Adyen, Braintree, and others.

API endpoints

Ansible Playbook for API integration. See [Playbooks](#) for details.

API

1. [API](#)
2. [API](#)
3. [API](#)
4. [API](#)
5. [API](#)
6. [API](#)
7. [API](#)
8. [API](#)

API

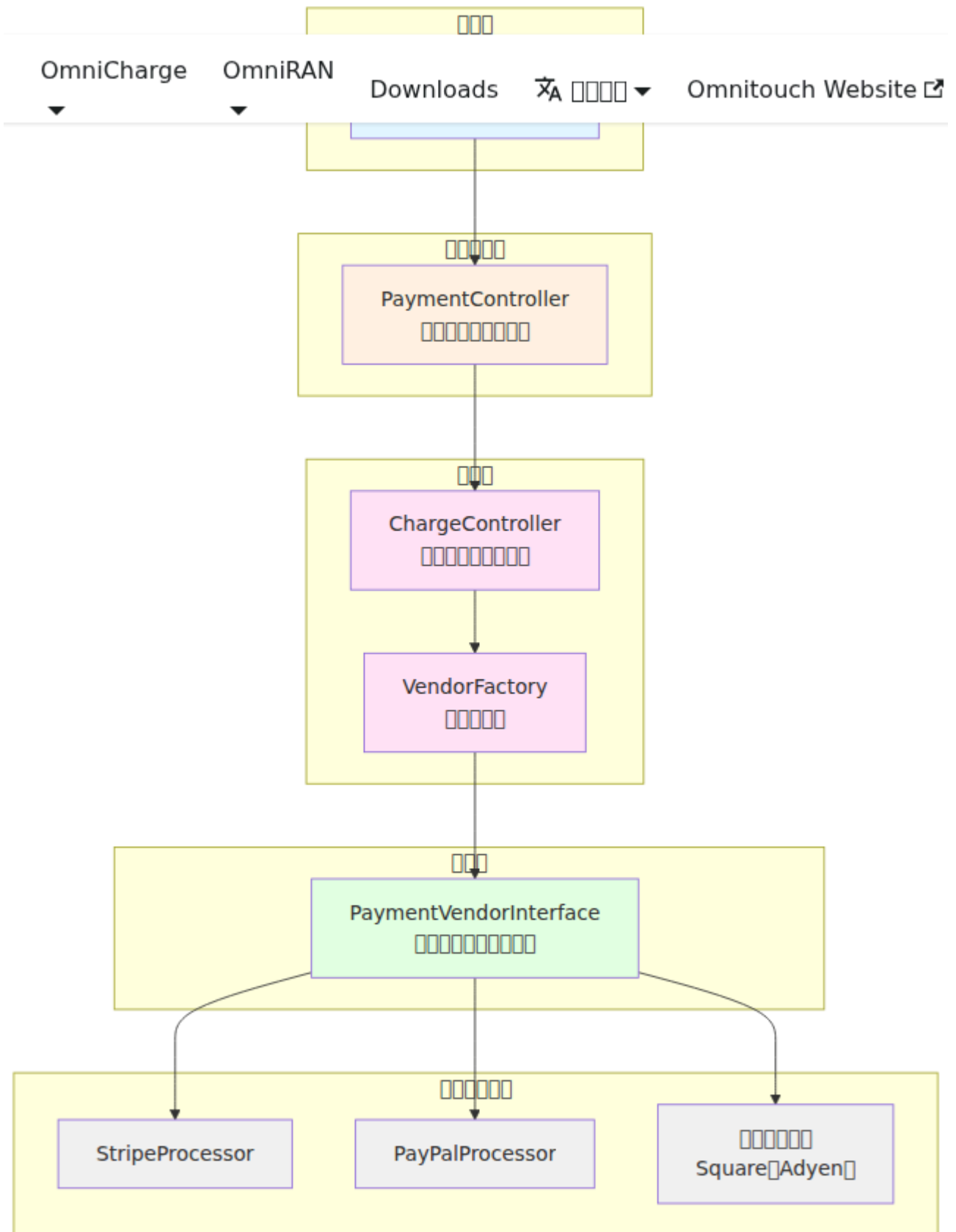
API

API

- [API](#)
- [API](#)

- □ □□□□□□□
- □ □□□□□□□API□□□□□

□□□



Payment Gateway

Payment Gateway: Square, Adyen, Braintree, OmniCRM

Payment Gateway: Stripe, PayPal, **API**

Payment Gateway



Payment Gateway

1. `PaymentVendorInterface`
2. `VendorFactory`
3. `PaymentGateway`
4. `API`

Payment Gateway

```
# Square API
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"
  }'
```

Payment Gateway API

Payment Gateway 1-2

□□□□□

Stripe□□

Stripe□□□□□□□□□□□□□□□□3D□□□□□□□□□□

1. □□Stripe API□□

- □ <https://stripe.com> □□
- □□□□□□ → API□□
- □□□□□□□□□□pk_live_...□pk_test_...□
- □□□□□□□□□□sk_live_...□sk_test_...□

□□□

- □□□□□□□□□□pk_test_/sk_test_□
- □□□□□□□□□□pk_live_/sk_live_□
- □□□□API□□□□□□□□□□

2. □□□□

□□□ `crm_config.yaml` □

```
payment_vendors:  
  stripe:  
    api_key: "sk_live_YOUR_SECRET_KEY_HERE"  
    publishable_key: "pk_live_YOUR_PUBLISHABLE_KEY_HERE"
```

□□□□□□□□

```
export STRIPE_SECRET_KEY="sk_live_YOUR_SECRET_KEY_HERE"  
export STRIPE_PUBLISHABLE_KEY="pk_live_YOUR_PUBLISHABLE_KEY_HERE"
```

3. □□□□

□□□ `.env` □



API - API Square API

```

{
  "vendor": "square",
  "vendor_payment_method_id": "sq_card_abc123",
  "payment_type": "card"
}

```

API API API API API API API

API API

API API

PaymentMethod

API API API API API API API

```

{
  "payment_method_id": 789,
  "customer_id": 123,
  "vendor": "stripe", // 'stripe', 'paypal' API
  "vendor_payment_method_id": "pm_xxx", // API ID
  "payment_type": "card", // 'card', 'paypal',
  'ach'
  "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

□□□□□□□□□□□□□□□□

```
{
  "authorization_id": 301,
  "customer_id": 123,
  "payment_method_id": 789,
  "vendor": "stripe", // □□□□□□□□
  "vendor_authorization_id": "auth_xxx", // □□□□□□□□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

□□□/□□□□□□□□

```
{
  "capture_id": 103,
  "authorization_id": 301,
  "customer_id": 123,
  "payment_method_id": 789,
  "vendor": "stripe",
  "vendor_transaction_id": "ch_xxx", // □□□□□□□□ID
  "amount": 200.00,
  "currency": "USD",
  "status": "succeeded", // 'succeeded', 'failed',
  'refunded'
  "captured_at": "2025-12-27T10:30:00Z",
  "vendor_response": {}, // □□□□□□□□□□
  "meta": {}
}
```

WalletAccount

钱包账户信息

```
{
  "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

钱包交易记录

```
{
  "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", // 支付捕获
  "reference_id": 103,
  "meta": {},
  "created_at": "2025-12-27T10:35:00Z"
}
```

钱包

钱包交易记录**交易**

API

API ID invoice_id

** **

API

```

{
  "invoice_id": 12345,
  "invoice_number": "INV-2025-000001", // INV-YYYY-NNNNNN
  "customer_id": 123,
  "title": " ",
  "paid": true, // 
  "void": false, // 
  "payment_reference": "ch_xxxxx", // /ID
  "payment_type": "stripe_capture", // 
  "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, // 
  "wholesale_cost": 250.00
}

```

INV-YYYY-NNNNNN 1 1

API

retail_cost

API

- " "
-
-
-

API

- 零售店
- 零售店
- 零售店 - 零售店

零售店

```
// 1. 零售店
{
  "transaction_id": 7001,
  "invoice_id": 12345,
  "retail_cost": 100.00, // 零售店 = 零售店
  "title": "零售店"
}

// 2. 零售店
{
  "transaction_id": 7002,
  "invoice_id": 12345,
  "retail_cost": -100.00, // 零售店 = 零售店
  "title": "零售店12345",
  "payment_type": "stripe_capture",
  "payment_reference": "ch_xxxxx"
}

// 零售店$0000 → 零售店
```

零售店 payment_reference payment_type payment_time

- 零售店 invoice_id 零售店 Hayden
- 零售店/零售店
- 零售店

零售店

零售店

API

Stripe API 的端点

端点

1. 创建支付方法
2. 更新支付方法
3. 删除支付方法
4. 获取支付方法
 - 获取所有支付方法
 - 获取单个支付方法

Stripe API 的端点 `Stripe API` `Stripe API`

API

端点 URL `/api/payments/`

端点 `**API**`

端点

端点

端点 `POST /api/payments/methods`

端点

```
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"
}'
```

□□□□

```
{
  "customer_id": 123,
  "vendor": "stripe", // 'stripe', 'paypal'□□□□
  "payment_token": "pm_xxxxx", // SDK□□□□□□□□
  "is_default": false, // □□□□□□□□□□
  "card_nickname": "My Visa Card" // □□□□□□□□
}
```

□□□201□□□□□□

```
{
  "success": true,
  "message": "Success",
  "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"
  }
}
```

Success

Success

GET /api/payments/methods?customer_id={id}

GET

```
curl -X GET "https://your-domain.com/api/payments/methods?
customer_id=123" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

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
    }
  ]
}
```

□□□□□□□□

□□□□□□□□□□□□

□□□ GET /api/payments/methods/default?customer_id={id}

□□□

```
curl -X GET "https://your-domain.com/api/payments/methods/default?
customer_id=123" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

□□□200 OK□□

```
{
  "success": true,
  "data": {
    "payment_method_id": 789,
    "vendor": "stripe",
    "payment_type": "card",
    "card_brand": "visa",
    "card_last4": "4242",
    "is_default": true
  }
}
```

□□□□□□□□

□□□□□□□□□□

□□□ PUT /api/payments/methods/set-default

□□□

```
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
  }'
```

□□□200 OK□□

```
{
  "success": true,
  "message": "□□□□□□□□□□",
  "data": {
    "payment_method_id": 790,
    "is_default": true
  }
}
```

DELETE

DELETE

DELETE /api/payments/methods/{payment_method_id}?customer_id={id}

curl

```
curl -X DELETE "https://your-domain.com/api/payments/methods/789?customer_id=123" \
-H "Authorization: Bearer YOUR_API_KEY"
```

200 OK

```
{
  "success": true,
  "message": "Deleted successfully"
}
```

API

DELETE

1. DELETE

DELETE

POST /api/payments/charge

curl

```
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"  
  }  
'
```

□□□□

```
{  
  "customer_id": 123,  
  "amount": 50.00,  
  "currency": "USD", // □□□USD  
  "payment_method_id": 789, // □□ - □□□□□□□□□□  
  "vendor": "stripe", // □□□□payment_token□□□□  
  "payment_token": "pm_xxxxx", // □□ - □□□□□  
  "save_method": false, // □□□□□□□□□□□□□□  
  "metadata": {  
    "order_id": "12345"  
  }  
}
```

□□□200 OK□□


```
{
  "success": true,
  "message": "成功",
  "data": {
    "transaction_id": "ch_xxxxx",
    "capture_id": 101,
    "amount": 50.00,
    "currency": "USD",
    "status": "succeeded"
  }
}
```

2. 如何接收支付通知

如何接收支付通知

通过 `POST /api/payments/invoice`

步骤

1. 创建支付通知
2. 接收支付通知
3. 验证支付通知 
4. 处理支付通知
5. 更新支付通知

步骤

POST /api/payments/authorize/hold

요청 파라미터

1. 고객 ID
2. 금액 - 통화
3. 결제 방법 ID
4. 지갑 사용 여부 (선택)

예시

```
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"
  }
}'
```

응답

```
{
  "customer_id": 123,
  "amount": 200.00,
  "currency": "USD",
  "payment_method_id": 789,           // 결제 방법 ID
  "vendor": "stripe",                // 결제 수단 payment_token
  "payment_token": "pm_xxxxx",      // 결제 토큰
  "save_method": false,
  "use_wallet": true,                // 지갑 사용 여부 true
  "metadata": {
    "reservation_id": "RES-001"
  }
}
```

200 OK

```
{
  "success": true,
  "message": "Success",
  "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,           // wallet
    "card_amount": 50.00,           // card
    "message": "Card $50 and wallet $200"
  }
}
```

2

POST /api/payments/capture/{authorization_id}

1. Success - OK
2. Success
3. Success
4. Success/Success

```
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": "测试",
    "description": "3测试",
    "wholesale_cost": 100.00,
    "contract_days": 3,
    "send_email": true
  }
}'
```

测试

```
{
  "amount": 200.00, // 金额 - 测试金额
  "metadata": {
    "invoice": true, // 是否生成发票
    "create_transaction": true, // 是否创建交易
    "title": "测试",
    "description": "3测试",
    "wholesale_cost": 100.00,
    "contract_days": 3,
    "send_email": true // 是否发送邮件
  }
}
```

测试200 OK

```
{
  "success": true,
  "message": "成功",
  "data": {
    "capture_id": 103,
    "transaction_id": "ch_xxxxx",
    "authorization_id": 301,
    "amount": 200.00,
    "currency": "USD",
    "status": "succeeded",
    "wallet_credit": { // 入账
      "ledger_id": 401,
      "amount": 50.00
    },
    "wallet_debit": { // 扣款
      "ledger_id": 402,
      "amount": 200.00
    },
    "transaction": { // 交易
      "transaction_id": 7001
    },
    "invoice": { // 发票
      "invoice_id": 12345,
      "invoice_number": "INV-2025-000001"
    }
  }
}
```

3

POST /api/payments/release/{authorization_id}

```
curl -X POST https://your-domain.com/api/payments/release/301 \
-H "Authorization: Bearer YOUR_API_KEY"
```

200 OK

```
{
  "success": true,
  "message": "成功",
  "data": {
    "authorization_id": 301,
    "vendor_authorization_id": "auth_xxxxx",
    "status": "released",
    "released_at": "2025-12-27T10:45:00Z"
  }
}
```

成功

4. 成功

成功

POST /api/payments/topup

成功 → 成功 → 成功

成功

```
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
  }
}'
```

成功/成功

```
curl -X POST https://your-domain.com/api/payments/topup \  
-H "Content-Type: application/json" \  
-H "Authorization: Bearer YOUR_API_KEY" \  
-d '{  
  "customer_id": 1,  
  "amount": 5.00,  
  "vendor": "stripe",  
  "payment_token": "pm_xxxxx",  
  "service_uuid": "hotspot-uuid",  
  "imsi": "123456789012345",  
  "days": 1,  
  "metadata": {  
    "is_rental": true,  
    "billing_email": "user@example.com"  
  }  
'
```

200 OK

```
{  
  "success": true,  
  "message": "Success",  
  "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 // false  
  }  
}
```

5. 請求の送信

APIクライアントを使用してAPIにリクエストを送信します。

リクエストは `POST /api/payments/rental` に送信されます。

リクエストの例

```
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": "レンタル",
    "metadata": {
      "rental_agreement_id": "RA-001"
    }
  }'
```

レスポンスの例

```
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": "レンタル",
    "metadata": {
      "billing_email": "renter@example.com"
    }
  }'
```

レスポンスは `200 OK` です。

```
{
  "success": true,
  "message": "成功",
  "data": {
    "transaction_id": "ch_xxxxx",
    "amount": 75.00,
    "payment_method_saved": false // 是否保存
  }
}
```

6. 退款

POST /api/payments/refund

curl

```
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"
}'
```

curl

```
{
  "transaction_id": "ch_xxxxx", // 交易ID
  "vendor": "stripe", // 'stripe', 'paypal'
  "amount": 50.00, // 金额 - 退款金额
  "reason": "customer_request" // 原因
}
```

返回200 OK

```
{
  "success": true,
  "message": " refunded ",
  "data": {
    "refund_id": "re_xxxxx",
    "amount": 50.00,
    "status": "succeeded"
  }
}
```

API

URL `/api/wallet/`

`GET /api/wallet/balance?customer_id={id}`

```
curl -X GET "https://your-domain.com/api/wallet/balance?
customer_id=123" \
-H "Authorization: Bearer YOUR_API_KEY"
```

`200 OK`

```
{
  "customer_id": 123,
  "balance": 150.50,
  "currency": "USD"
}
```

GET /api/wallet/info?customer_id={id}

```
curl -X GET "https://your-domain.com/api/wallet/info?customer_id=123" \
-H "Authorization: Bearer YOUR_API_KEY"
```

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
  }
}
```

POST /api/wallet/topup

/PayPal →

```
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  
'
```

请求

```
{  
  "customer_id": 123,  
  "amount": 100.00,  
  "currency": "USD", // 币种USD  
  "payment_method_id": 789 // 支付方式 - 信用卡  
}
```

返回200 OK

```
{  
  "success": true,  
  "payment": {  
    "transaction_id": "ch_xxxxx",  
    "amount": 100.00  
  },  
  "wallet": {  
    "ledger_id": 503,  
    "balance_after": 250.50  
  },  
  "message": "充值100.00成功"  
}
```

返回数据

成功返回数据

```
GET /api/wallet/transactions?customer_id={id}&limit={n}&offset={n}&type={type}
```

curl

```
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"
```

parameters

- `customer_id` ID
- `limit` 50
- `offset` 0
- `type` 'credit' | 'debit' | 'refund' | 'adjustment'

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,
      "description": "□□□",
      "reference_type": "payment_capture",
      "reference_id": 103,
      "created_at": "2025-12-27T10:35:00Z"
    },
    {
      "ledger_id": 502,
      "transaction_type": "debit",
      "amount": 50.00,
      "balance_before": 250.50,
      "balance_after": 200.50,
      "description": "□□□□",
      "reference_type": "service_charge",
      "reference_id": 789,
      "created_at": "2025-12-27T11:00:00Z"
    }
  ]
}
```

API

Payment Methods

Method	Endpoint	Response
POST	<code>/api/payments/methods</code>	Payment Method
GET	<code>/api/payments/methods?customer_id={id}</code>	Payment Methods
GET	<code>/api/payments/methods/default?customer_id={id}</code>	Payment Method
PUT	<code>/api/payments/methods/set-default</code>	Payment Method
DELETE	<code>/api/payments/methods/{id}?customer_id={id}</code>	Payment Method


Payment Transactions

Method	Endpoint	Response
POST	<code>/api/payments/charge</code>	Payment Transaction
POST	<code>/api/payments/invoice</code>	Payment Transaction
POST	<code>/api/payments/topup</code>	Payment Transaction
POST	<code>/api/payments/authorize/hold</code>	Payment Transaction
POST	<code>/api/payments/capture/{id}</code>	Payment Transaction
POST	<code>/api/payments/release/{id}</code>	Payment Transaction
POST	<code>/api/payments/rental</code>	Payment Transaction
POST	<code>/api/payments/refund</code>	Payment Transaction

□□□□

□□	□□	□□
GET	<code>/api/wallet/balance?customer_id={id}</code>	□□□□□□
GET	<code>/api/wallet/info?customer_id={id}</code>	□□□□□□ + □□
POST	<code>/api/wallet/topup</code>	□□□□□□□□□□
GET	<code>/api/wallet/transactions?customer_id={id}</code>	□□□□□□

PayPal□□□□

 □	□□	□□
POST	<code>/api/payments/paypal/vault/setup-token</code>	□Card Fields SDK□ □PayPal□□□□
POST	<code>/api/payments/paypal/vault/finalize</code>	□□PayPal□□□□□□□□ □□
POST	<code>/api/payments/paypal/vault/update-setup-token</code>	□□3DS/SCA□□□□□□ □□

□□□□

□□**1**□□□□□□□□□□

□□□□□□□□□□□□□□□□30□□□□□□

```
# 创建支付记录
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": "30天会员"
    }
  }'
```

返回

```
{
  "success": true,
  "message": "支付成功",
  "data": {
    "service_amount": 99.99,
    "wallet_portion_used": 50.00, // 钱包$50
    "card_portion_used": 49.99, // 卡片$49.99
    "final_balance": 0.00
  }
}
```

2. 创建订单

创建订单时，需要指定商品 ID 和数量。同时，还可以指定优惠券 ID 和支付方式。

```
# 実行コマンド
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"
    }
  }'
```

結果

1. 請求額\$149.99
2. 請求内容
3. 請求額 → 請求内容
4. 請求額 → 請求内容

3. 請求内容

請求内容: WiFi 請求内容

```
# 创建充值
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": "测试地点"
    }
  }'
```

返回

- 充值成功
- 充值失败
- 充值失败原因
- 充值失败原因

4 充值失败原因 + 返回

充值失败原因

```
# 1. authorize - authorize
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"
    }
  }'
```

```
# 2. check - check
curl -X GET https://your-domain.com/api/payments/authorize/301 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{"authorization_id": 301, "status": "authorized"}'
```

```
# 3a. capture - capture
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": "Invoice - 301",
      "description": "3020001202700300",
      "send_email": true
    }
  }'
```

```
# 3b. release - release
curl -X POST https://your-domain.com/api/payments/release/301 \
  -H "Authorization: Bearer YOUR_API_KEY"
```

5 Stripe

Visa

```
# 1 Stripe.js Stripe
# const {token} = await stripe.createToken(cardElement);

# 2
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": "Visa"
}'
```

6

\$100

```
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
}'
```

```
{
  "success": true,
  "payment": {
    "transaction_id": "ch_xxxxx",
    "amount": 100.00
  },
  "wallet": {
    "balance_after": 250.50
  },
  "message": "成功支付100.00元"
}
```

成功支付

成功支付

成功支付400元

```
{
  "error": "成功支付50.00元"
}
```

成功支付400元

```
{
  "error": "成功支付"
}
```

成功支付400元

```
{
  "error": "customer_id amount"
}
```

404

```
{  
  "error": "999"  
}
```

500

```
{  
  "error": "500"  
}
```

Stripe/PayPal

POST

Stripe/PayPal

1. Stripe/PayPal

Stripe/PayPal

Stripe/PayPal

POST /api/payments/refund

POST

1. 購買\$100
2. 購買
3. 購買
4. 購買\$100 5-10 + 購買
 - 購買\$100
 - 購買
 - 購買
 - 購買

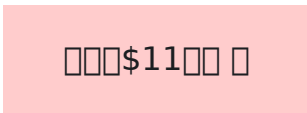
購買

購買

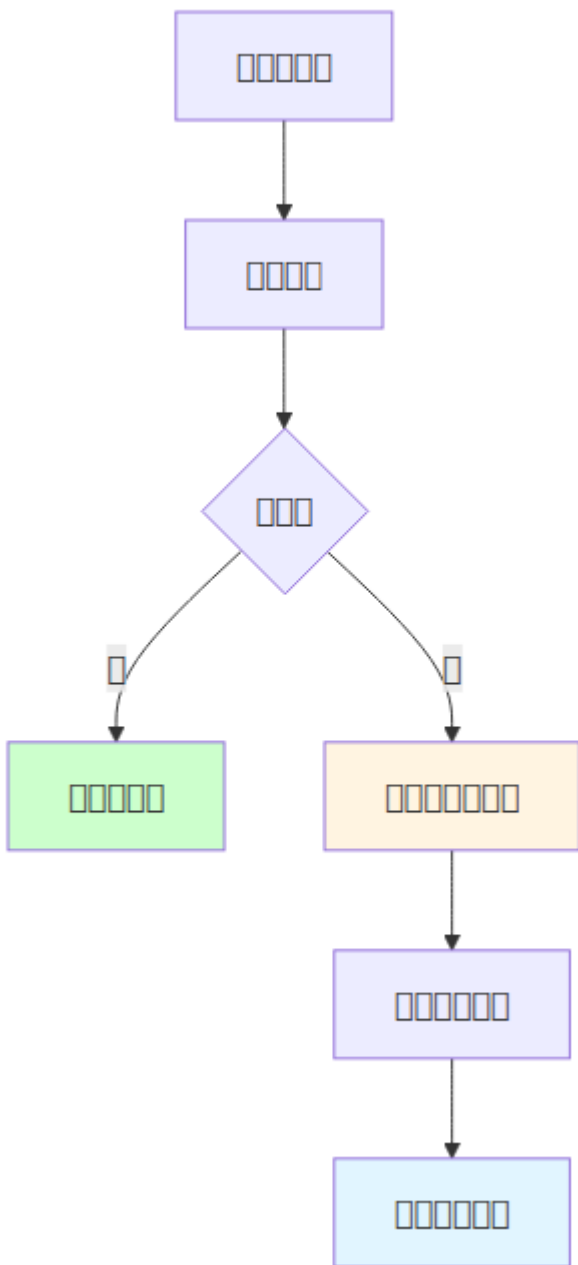
1 \$1 + \$10

購買 \$1 \$10

購買



OmniCRM



API

```
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
  }
}'
```

API

```
{
  "success": true,
  "message": "成功",
  "data": {
    "customer_id": 123,
    "service_amount": 10.00,
    "routing_mode": "hybrid",
    "initial_balance": 1.00,
    "wallet_portion_used": 1.00, // 財布$1
    "card_portion_used": 9.00, // カード$9
    "charged_amount": 9.00, // ← カード
    "wallet_credited": 9.00, // ← 財布
    "wallet_debited": 10.00, // ← カード
    "final_balance": 0.00
  }
}
```

2\$50 + \$30

\$50\$30

□□□□\$50

□□□□
\$30 - \$50 = \$0

□□□

□□□□

□□□□

□□□□-\$30

□□□□\$20
□□□□\$0
□□□□\$30 □

API□□□

```

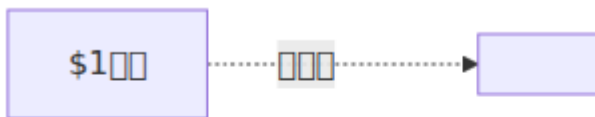
{
  "success": true,
  "data": {
    "service_amount": 30.00,
    "initial_balance": 50.00,
    "charged_amount": 0, // ← 0000
    "wallet_debited": 30.00,
    "final_balance": 20.00,
    "payment_method_used": false // ← 0000
  }
}

```

3

\$500 \$150

1 POST /api/payments/authorize/hold



API

```

{
  "authorization_id": 301,
  "amount": 500.00,
  "status": "authorized",
  "wallet_balance": 150.00,
  "wallet_to_use": 150.00,
  "card_amount": 350.00,
  "message": "$350 $500"
}

```

2 POST /api/payments/capture/301

□□□□

□□□□\$350

□□□□+\$350
□□□\$500

□□□□-\$500
□□□□

□□□□\$0
□□□□\$350

□□□□

- □ □□□□□□□\$350□□□□\$500
- □ □□□□□□□□□□□
- □ □□□□□□□□□
- □ □□□□□□□□□

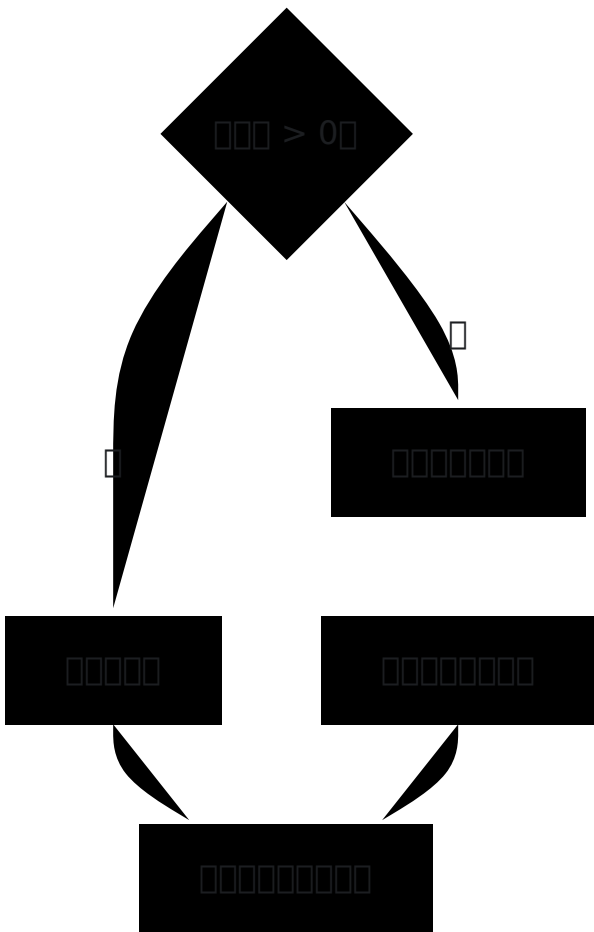
□□□□

□□□□□□□□□□□□

total

total > 0

```
total
wallet_portion = min(total, wallet_max)
card_portion = total - wallet_portion
```



00

00000

1. 0000000000
2. 00000000 - 00000
3. 0000 > 00000000000
4. 0000000000
5. 0000000000
6. 00000000000000000000

0000000

00000000000000000000

0000 - 0000000000000000

```
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"  
  }  
'
```

000000\$100000000\$10000000\$100000\$0

0000000000000000/000000000000

0000

1. 0000000000000000000000000000

2. ...
3. ...
4. ...
5. ...
6. ...
7. ...7...
8. ...

Playbook

Ansible playbook

1. ...try/catch...
2. **authorization_id** - ...
3. **API** - ...
4. ...2...
5. ...

Playbook

API

API Authorization

```
Authorization: Bearer YOUR_API_KEY
```

API

API Gateway

Payment Gateway

- **Stripe** - 支持ACH
- **PayPal** - 支持PayPal Card Fields SDK

API Gateway

API Gateway 支持多种支付方式

API Gateway

Payment Gateway

- **Playbooks** - Ansible playbook
 - 配置
 - 部署/更新
 - 备份
 - 清理

API Gateway

- **API Gateway** - 支持多种支付方式
- **Payment Gateway** - 支持PayPal
- **API Gateway** - 支持多种支付方式
- **API Gateway** - 支持多种支付方式
- **API Gateway** - 支持多种支付方式

API Gateway

- API Gateway → 支持多种支付方式
- 支持Stripe/PayPal → 支持多种支付方式

- 000000 → 0000000
 - 000000 → 000Playbook00
-



API API API API

Transactions </payments_transaction>

Mailjet <integrations_mailjet> PDF

OmniCRM API

OmniCRM API HTML templates Jinja2 templates

Templates

OmniCRM-API/invoice_templates/

Templates

- norfone_invoice_template.html - Norfone
- cifi_invoice_template.html - CIFI

Config

OmniCRM-API/crm_config.yaml

```
invoice:  
  template_filename: 'norfone_invoice_template.html'
```

Usage

OmniCRM API Jinja2 templates

□□□□

- `{{ invoice_number }}` - □□□□ ID□□□□ `INV-2025-001234` □
- `{{ date }}` - □□□□□□□□ ISO □□□ `2025-01-10T12:00:00` □
- `{{ due_date }}` - □□□□□□□□□□ `2025-02-10` □
- `{{ start_date }}` - □□□□□□□□
- `{{ end_date }}` - □□□□□□□□
- `{{ total_amount }}` - □□□□□□□□□□
- `{{ total_tax }}` - □□□□□□□□□□□□

□□□□

- `{{ client.name }}` - □□□□□□□□□□
- `{{ client.address.address_line_1 }}` - □□□ 1
- `{{ client.address.address_line_2 }}` - □□□ 2
- `{{ client.address.city }}` - □□
- `{{ client.address.state }}` - □/□
- `{{ client.address.zip_code }}` - □□□□
- `{{ client.address.country }}` - □□

□□□□

□□□□□□□□□□□□

```
{% 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 %}
```

□□□□

- `sub_transaction.transaction_id` - □□ ID

- `sub_transaction.created` - 日期/时间
- `sub_transaction.title` - 标题
- `sub_transaction.description` - 描述
- `sub_transaction.retail_cost` - 零售成本
- `sub_transaction.tax_percentage` - 税率 (例如 10 或 10%)
- `sub_transaction.tax_amount` - 税额

模板代码

```
<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>
```

模板文件

第 1 步

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

第 2 步 HTML/CSS

将 `your_company_invoice_template.html` 复制到

模板目录

1. 创建模板

```
<!-- Placeholder URL -->


<!-- Placeholder -->
<h1>Placeholder</h1>
```

2. CSS

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

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

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

3. Footer

```
<footer>
  <p>Placeholder</p>
  <p>123 Placeholder</p>
  <p>Placeholder+1-555-123-4567 | Placeholderbilling@yourcompany.com</p>
  <p>ABN/Placeholder12345678900</p>
</footer>
```

4. CSS

```
<div class="payment-info">
  <h3>결제 정보</h3>
  <p><strong>결제 URL</strong> | https://yourcompany.com/pay |</p>
</div>
<p><strong>결제 정보</strong></p>
<ul>
  <li>회사명: Your Company Ltd</li>
  <li>은행: BSB 123-456</li>
  <li>계좌번호: 987654321</li>
  <li>계좌번호: {{ invoice_number }}</li>
</ul>
</div>
```

5. 결제 조건

```
<div class="terms">
  <h4>결제 조건</h4>
  <p>결제 기한: 30일 이내</p>
  <p>결제 수수료: 2%</p>
  <p>문의: billing@yourcompany.com</p>
</div>
```

3. 결제 정보

OmniCRM-API/crm_config.yaml

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

4. API

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

5. 테스트

- 결제 정보 확인
- 결제 조건 확인

3. PDF 生成

4. 静的コンテンツ

静的コンテンツ

静的コンテンツ

Jinja2 テンプレート

```
{% if total_amount > 1000 %}
  <div class="high-value-notice">
    <p><strong>注意</strong> 高価な注文です - ご注意ください</p>
  </div>
{% endif %}

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

静的コンテンツ

静的コンテンツ

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

静的コンテンツ

静的コンテンツ

```

<!-- 表头 -->
<tr>
  <td colspan="4" class="text-right"><strong>总计</strong></td>
  <td>${{ "%.2f"|format(total_amount) }}</td>
</tr>
<tr>
  <td colspan="4" class="text-right"><strong>税金</strong></td>
  <td>${{ "%.2f"|format(total_tax) }}</td>
</tr>
<tr>
  <td colspan="4" class="text-right"><strong>总计(含税)</strong></td>
  <td>${{ "%.2f"|format(total_amount + total_tax) }}</td>
</tr>

```

total_tax 税金总额 tax_amount 商品金额 tax_percentage 税率 0%

表头

表尾

```

<div class="qr-payment">
  ![[QR]](https://api.qrserver.com/v1/create-qr-code/?
  size=150x150&data={{ payment_url }})
  <p>QR支付</p>
</div>

```

PDF 打印

OmniCRM 使用 WeasyPrint 将 HTML 转换为 PDF

使用 CSS

- 使用 CSS 2.1
- 使用 CSS3 flexbox
- 使用 @font-face 嵌入字体

打印

- JavaScript
- CSS □□□□□□□□
- □□□□
- □□□□ CSS □□

□□□□□□□□

```
@page {  
  size: A4;  
  margin: 1cm;  
}  
  
body {  
  font-family: Arial, sans-serif;  
  font-size: 10pt;  
}
```

□□□□□□□□

```
@media print {  
  .no-print {  
    display: none;  
  }  
  
  .page-break {  
    page-break-after: always;  
  }  
}
```

□□□□□□

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

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

□□□□□

□□□□□□□□□□□□□□□□□□□□

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

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

□□□□□□□

□□□□□

1. □□□□□
 - □□□□□□
 - □□□□□□□□
 - □□□□□□□□□□
 - □□□□□□
 - □□□□□□
2. □□□□□□
 - □□□□□□□□
 - □□□□□□□□

- `Content-Disposition`
- `Content-Type`

3. PDF `Content-Type`

- `Content-Disposition: inline; filename="invoice.pdf"`
- `Content-Type: application/pdf`
- `Content-Length: 12345`
- `Cache-Control: no-cache`

4. `Content-Disposition`

- `Content-Disposition: inline`
- `Content-Disposition: attachment; filename="invoice.pdf"`
- `Content-Disposition: attachment; filename="invoice.pdf"; filename*=UTF-8''invoice.pdf`

5. `Content-Type`

- `Content-Type: application/pdf`
- `Content-Type: text/html` Mailjet `Content-Type: text/html; charset=UTF-8`
- `Content-Type: text/html; charset=UTF-8` Gmail/Outlook/Apple Mail

`Content-Type: application/pdf`

`Content-Type: application/pdf` API

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

`Content-Type: application/pdf`

`Content-Type: application/pdf`

- `Content-Type: application/pdf`
- `Content-Type: application/pdf; charset=UTF-8`

PDF `Content-Type`

- `Content-Type: application/pdf` CSS
- `Content-Type: application/pdf` CSS 2.1 `Content-Type: application/pdf` WeasyPrint `Content-Type: application/pdf` CSS

URL

- URL 格式
- HTTPS URL

Table Layout

- Table Layout
- `table-layout: fixed`

Font

- Font
- Arial, Times New Roman

PDF

- HTML to PDF (WeasyPrint)
- PDF to HTML (WeasyPrint)

PDF

PDF 格式, OmniCRM PDF, PDF 格式, PDF 格式

PDF

1. PDF 格式
 - PDF
 - PDF Base64
 - PDF SHA256
 - `Invoice_PDF_Cache`
 - ID
 - PDF Base64
 -
 -
 -

2. 查詢 - 查詢所有已過期
 - 查詢 invoice_id 查詢 PDF
 - 查詢所有已過期 PDF
 - 增加 last_accessed 查詢所有已過期
3. 查詢 - 查詢所有已過期 PDF 查詢
 - 查詢所有已過期/查詢所有已過期
 - 查詢所有已過期
 - 查詢所有已過期

查詢

- 查詢 - 查詢所有已過期 PDF 查詢
- 查詢 - 查詢所有已過期 PDF 查詢
- 查詢 - 查詢 PDF 查詢 CPU 查詢
- 查詢 - 查詢所有已過期 PDF 查詢

查詢

查詢 PDF 查詢所有已過期 PDF 查詢

- 查詢所有已過期 90 查詢
- 查詢所有已過期 30 查詢
- 查詢所有已過期 PDF 查詢

API 查詢

查詢 API 查詢 UI 查詢

- 查詢所有已過期 PDF 查詢
- 查詢所有已過期 PDF 查詢
- 查詢所有已過期 PDF 查詢

查詢所有已過期 PDF 查詢

```
-- 查詢所有已過期 PDF 查詢 MySQL 查詢
DELETE FROM Invoice_PDF_Cache;
```

crm_config.yaml

crm

crm

crm

1. crm → [crm]
2. crm
3. crm

crm

1. crm → crm
2. crm

crm

crm

crm

- crm - crm

- 00000 - 00000000000000000000
- 0000 - 0000000000000000
- 0000 - 0000000000000000

00000

- 00 1,000 000000 "k" 000000\$1.5k
- 00 1,000,000 000000 "M" 000000\$2.3M
- 00 1,000,000,000 000000 "B" 000000\$1.1B

00000

- "00"0"00"000000000000
- 00000000000000000000
- 00000000000000000000
- 00000000000000

00000

00000000000000000000

0000

- **ID** - ID
- - /
- - "N/A"
- -
- -
- -
- -
- -

- ↓ () - PDF
- () -
- () -
- ☒ () -
- () - Stripe Stripe

" + " "

- 00 - 00000000000000000000

0000 PDF

000000 0000 (↓) 0000000000 "00 PDF" 0000000000 PDF

000000

1. 000000000000
2. 00000000000000000000
3. 0000000000 Invoice_01234.pdf
4. PDF 000000000000

PDF 000000

- 0000 - 000000 PDF0000000000000000 2-3 00
- 0000 - 00000000
- 0000 - 00000000000000000000

0000000000

- 0000000000 - 0000000000API 000000
- **PDF** 000000 - 0000000000000000
- 0000 - 0000000000000000000000

000000

00 0000 (0) 00000000

목차

1. 결제 수단
2. 결제 수단
3. 결제 수단
 - **Stripe** 결제 - 결제 수단
 - 결제 - 결제 수단
 - 결제 - 결제 수단
 - **POS** 결제 - 결제 수단
 - 결제 - 결제 수단
4. 결제 수단 Stripe
 - 결제 수단
 - 결제 수단
 - 결제 수단
5. 결제 수단
6. 결제 수단 "결제" 결제
7. 결제 수단
 - **Stripe** - 결제 Stripe API 결제
 - 결제 - 결제 수단
8. 결제 수단 "결제"
9. 결제 수단

2. PDF
3. Mailjet <integrations_mailjet> api_crmCommunicationCustomerInvoice
4.
 - o PDF
 - o
 - o
 - o
 - o
 - o
 - o
5. “”

“”

- {{ var:customer_name }} -
- {{ var:invoice_number }} - ID
- {{ var:invoice_date }} -
- {{ var:due_date }} -
- {{ var:total_amount }} -
- {{ var:invoice_url }} - PDF
- {{ var:pay_url }} -

- - crm_config.yaml Mailjet API
- -
- **PDF** - PDF

()

□□□

- □□□□□ □□□
- □□□□□□□□□□□□□□□□

□□□□□

1. □□□□□□□□□□□□
2. □□□□□□ (□)
3. □□□□□□□□

□□□□□□

- □□□□□ `void = true`
- □□□□□□□□□□□□ `invoice_id` □□□ `null`
- □□□□□□ "□□□"
- □□□□□□□□□□□□
- □□□□□□□□ "□□□" □□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□ "□□" □□□□□□

□□□□□

- □□□□□□□□
- □□ = "□□□□□□□□□□□□□□□□□□□□□□"

- `id = "xxxxxxxxxxxx"xxxxxxxxxxxx`

退款

退款 () 退款

退款

- 退款
- 退款 **Stripe**
- 退款 `payment_reference` Stripe ID
- 退款

退款

1. Stripe
2. ()
3. 退款

4. "退款"
5. Stripe
 - Stripe API
 - Stripe
 - `refund_reference`
6. "退款"

7. 支払い方法

🔗🔗🔗 支払い方法

- 支払い方法の種類
- 支払い方法 "カード"
- 支払い方法の種類
- 支払い方法の種類3-7 支払い方法
- Stripe 支払い方法の種類

支払い方法

- 支払い方法の種類 POS 支払い方法の種類
- 支払い方法の種類
- 支払い方法

支払い方法の種類

支払い

支払い方法の種類

- 支払い ID
- 支払い方法
- 支払い方法の種類

支払い

支払い方法の種類

支払い方法

- 支払い方法 - 支払い方法の種類
- 支払い方法 - 支払い方法の種類

支払い方法



7.
 -
 -
 -
8. " " "15"
9.
10.
11. " " "
12. PDF

2

1. +
2.
3.
 - "+ "
 - " - 2025 1 "
 - 1 1 31
 - 2 15
 - X
 - X
 -
 -
 - " " "
4.
 - "+ "
 - " - 2025 1 "
 - 1 1 31
 - - X
 -
 - " "

5. 00000000000000000000
 - o 0000 "+ 0000000"
 - o 000"0000 - 2025 0 1 0"
 - o 00000000000000000000
 - o 00 "00" 000000
 - o 00 "0000"
6. 000000000000000000000000

0000 **3**0000000000000000

1. 0000000
2. 000000   0 000
3. 00 "+ 0000000"
4. 0000000000
5. 0000000 20 000
6. 000000000000000000000000
7. 000000000
 - o 00000000000"0000000"0
 - o 00 **X** 0000000
 - o 000000000000"00000"0
 - o 00 **X** 0000000
 - o 00000000"0000 18 000"
 - o 0000000000
8. 000000000000000000000000
9. 00 "000000**18**0"
10. 00000000000000000000
11. 00/000000000000000000000000

0000 **4**0000000000000000

1. 00000000000000
2. 0000000000000000
3. 00000000000000
 - o 00000000000000

- **API X** 認證
- 認證
- **API X** 認證

4. 認證
5. 認證
6. 認證 "API X" 認證
7. 認證/認證
8. 認證

認證 5 認證

1. 認證
2. 認證
3. 認證 認證
4. 認證 "+ 認證"
5. 認證
 - 認證
 - 認證
 - 認證 + 7 認證
 - 認證“認證”
6. 認證
7. 認證
8. **API X** 認證
9. 認證 "API X"
10. 認證
11. 認證 Stripe 認證

認證 6 認證

1. 認證
2. 認證
3. 認證 認證
4. 認證
 - 認證 ID 認證

- 0000000000
 - 0000000000
 - 0000000000
5. 0000000000000000
 6. 0000000 PDF0000000000 "00 **PDF**" 00
 7. 00000000000000000000 "0000000" 00
 8. 0000000000

00000 **7**0000000000

1. 0000000000
2. 0000000000000000
3. 000000000000000000
4. 00000000
5. 0000000000000000
6. 0000000000 "00" 00
7. 00000
8. 00000000000000
9. 0000000000000000000000
10. 000000000000000000
 - 0000000000000000000000
 - 0000000000
11. 0000000000000000000000

00000 **8**0000000000

1. 00000000000000
2. 000000 00 0000
3. 0000000000
4. 0000000000000000
5. 0000000000
6. 0000000000 "0000" 00
7. 00 "00" 00000
8. 000000 "0000 2025-01-15"

9. `return` " "
10. `return` " "
11. `return`
12. `return` " "

9

1. `return`
2. `return` Stripe
3. `return`
4. `return`
5. `return`
6. `return` " " Stripe
7. `return`
8. `return` Stripe
9. `return` " "
10. `return` 3-7
11. `return`

-

-
-

PDF

- WeasyPrint
- HTML API

Stripe

- Stripe API
- Stripe

環境構築

- 環境構築
- 環境構築に必要なライブラリをインストール Stripe 環境構築に必要なライブラリ

メール送信

- Mailjet API 環境構築に必要なライブラリをインストール
- 環境構築に必要なライブラリをインストール `crm_config.yaml` Mailjet 環境構築に必要なライブラリ

決済処理

- 決済処理/POS/決済処理に必要な Stripe 環境構築に必要なライブラリ
- 決済処理に必要な Stripe 環境構築に必要なライブラリ

PDF 生成

- PDF 生成に必要なライブラリ
- PDF 生成に必要なライブラリ `DELETE FROM Invoice_PDF_Cache WHERE invoice_id = X;`

データベース

- データベース構築
- データベース構築に必要なライブラリ

テスト

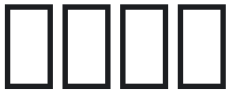
- テスト環境構築
- テスト環境構築に必要なライブラリ

デプロイ

- デプロイ環境構築
- デプロイ環境構築に必要なライブラリ

運用

- 運用環境構築
- 運用環境構築に必要なライブラリ



□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□”□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□”□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

1. 10^0 to 10^1
2. 10^1 to 10^2

10 100 1000

10000 100000 1000000

1000000

- 1000 - 1000000000000
- 1000000 - 1000000000000
- 1000000000000 - 1000000000000
- 1000000000000000 - 1000000000000000

1000000

- 1,000 千 "k" 1.5k
- 1,000,000 万 "M" 2.3M
- 1,000,000,000 十亿 "B" 1.1B

10000

1000000000000

□□□□

- **ID** - □□□□ID
- □□ - □□□□□□
- □□ - □□□□□□
- □□ - □□□□□□□□□□
- □□ - □□□□□□□□□□□□□□□□
- □□ - □□□□□□□□□□□□□□□□ID□□□□□□□□
- □□ - □□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□

□□□□□□□□□□□□□□ : □□□□□□□□

- □□□□ - □□□□□□□□□□
- □□□□**PDF** - □□PDF□□□□□□□□□□
- □□□□ - □□□□□□□□□□□□□□□□

□□□□□

□□□□□□□□□

□□□□□□□□

□□□□□□□□□□□□□□

- □□□□□□ - □□□□□□□□□□□□
- □□□□□□ - □□□□□□□□
- □□□□ - □□□□□□□□□□□□
- □□□□ - □□□□□□□□□□□□□□□□SIM□□□□□
- □□□□□□ - □□□□□□□□□□
- □□□□ - □□□□□□□□□□□□

□□□□□□□/□□□□

□□□□□□□□□□□□

- □□□□ - □□□□□□□□
- □□□□ - □□□□□□□□/□□□□□□
- □□□□□□□□ - □□□□□□□□□□□□
- □□□□□□ - □□□□□□□□
- □□ - □□□□□□□□
- □□ - □□□□□□□□

□□□□□□□□

□□ " + □□□□ " □□□□□□□□□□□□

□□□□□□□□□□

□□□□□□□□/□□□□

□□□□

- □□□□ - □□□□□□□□□□□□□□/□□□□
- □□□□ - □□□□□□□□□□□□□□□□□□□□□□
- □□ - □□□□□□□□□□
- □□ - □□□□□□□□
- □□□□ - □□□□□□□□□□□□□□
- □□□□ - □□□□□□□□□□□□□□
- □□ - □□□□□□□□□□□□□□□□□□□□0%
- □□ - □□□□□□□□□□□□□□
- □□ - □□□□□□□□□□□□□□
- □□□□ - □□□□□□□□□□□□

□□□

- □□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□

1. □□□□□□□□□□□□
2. □□□□□□□□□□□□

3. 0000“000000”0000
4. 000000000000
5. 0000000000

0000000000

00

00000000000000000000

- 00ID
- 00
- 00
- 00ID

0000

0000000000000000

00000000

- 00000 - 000000000000
- 00000 - 000000000000

00000000

- 000000 - 00000000000000
- 000000 - 000000000000000000

00

0000000000000000

- **ID** - 0000ID000000/00000000
- 00 - 00000000
- 00 - 00000000

- 00 - 000000000000/0000
- 00 - 0000!D00

0000000000000000 ↔ 0000

0000

0000000000 0000000000

000

- 00000000
- 00000000000000000000

00000

1. 00000000
2. 00000000 : 0
3. 00 "0000"
4. 00000000

000000

- 00000 void = true
- 00000000000000

- 零售店
- 零售店“店”
- “零售店”

零售店 零售店“店” 零售店“店”

零售店

零售店零售店零售店零售店零售店零售店零售店零售店零售店零售店

零售店

- 零售店 - 零售店
 - 零售店 - 零售店零售店零售店零售店零售店
 - 零售店 - 零售店
 - 零售店 - 零售店 $\text{retail_cost} \times (\text{tax_percentage} / 100)$
 - 零售店 - 零售店零售店零售店\$10.00 (10%)
- 零售店/零售店 - 零售店
 - 零售店零售店零售店
 - 零售店零售店零售店零售店0%
 - 零售店零售店零售店零售店零售店零售店

零售店

- 10% of \$50.00
- $\$50.00 \times 0.10 = \5.00
- \$5.00 (10%)

NIL/

- 0
- 0%
- "-"

-
-
-
- " " "

- 00000000 (-)

00000000

- 00000000
- 00000000000000000000
- 00ID0000000000000000
- 000000000 (✓)
- 00000

000000

0000000000000000

1. 00000000000000000000
2. 000000000000
3. 000000000000
4. 000 invoice_id 000000
5. 00000000"0000"

000 payments_invoices 0000000000000000

00000000

00000 **1**0000000000000000

1. 000000"00000020"
2. 00000000£100000
3. 000000 00 00
4. 00 "+ 00000"
5. 00 00 00000
6. 00 00000 00000
7. 000000"00000000"
8. 000000"0108-90000000"
9. 00000000010.00

10. 0000000000000000
11. 00 "0000"
12. 000-£10.00000000
13. 0000000000000000

0000 2000000000

1. 000000000000
2. 000000£75000000
3. 000000 00 00
4. 00 "+ 0000"
5. 00 00 0000
6. 000000"0000"
7. 000000"00000000000000"
8. 0000000075.00
9. 0000000045.00000000000000
10. 0000000000
11. 00000000
12. 00 "0000"
13. 00000000000000
14. 000000000000

0000 3000000000

1. 000000000000
2. 0000000000
3. 0000000000000000 : 0
4. 00 "0000"
5. 0000000000
6. 0000000000
7. 000000000000

□□□□

- `payments_invoices` - □□□□□□
- `payments_process` - □□□□□□
- `basics_payment` - □□□□□
- `csa_activity_log` - □□□□□□□□□□

Playbooks

Ansible playbooks OmniCRM

API API API

OmniCRM Ansible Playbooks CRM API

1. -
2. -

Playbooks

retail_cost wholesale_cost retail_setup_cost
playbook

Ansible playbook

- api_response_product.json.retail_cost
- -
- -
- -
- -
- -
- -
- -

□□□□□□□□

```
# □□ retail_cost □ $99.00□□□□□□ $79.00 □□□□□□
- name: □□□□□□□□□□□□□□□□ retail_cost□
  set_fact:
    charge_amount: 79.00

- name: □□□□□□
  uri:
    url: "http://localhost:5000/crm/payments/authorize/hold"
    method: POST
    body_format: json
    body:
      customer_id: "{{ customer_id | int }}"
      amount: "{{ charge_amount | float }}" # □□□□□□□□□□□□
retail_cost
# ...
```

□□□□□□□□

```
# □□□□□□□□□□□□
- name: □□□□□□□□
  uri:
    url: "http://localhost:5000/crm/customer/{{ customer_id }}"
    # ...
  register: customer_info

- name: □□□□□□□□
  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: □□□□□□□□□□□□
  set_fact:
    final_price: "{{ (base_price | float) * (1 - (discount_percent
  | float / 100)) | round(2) }}"
```

Ansible

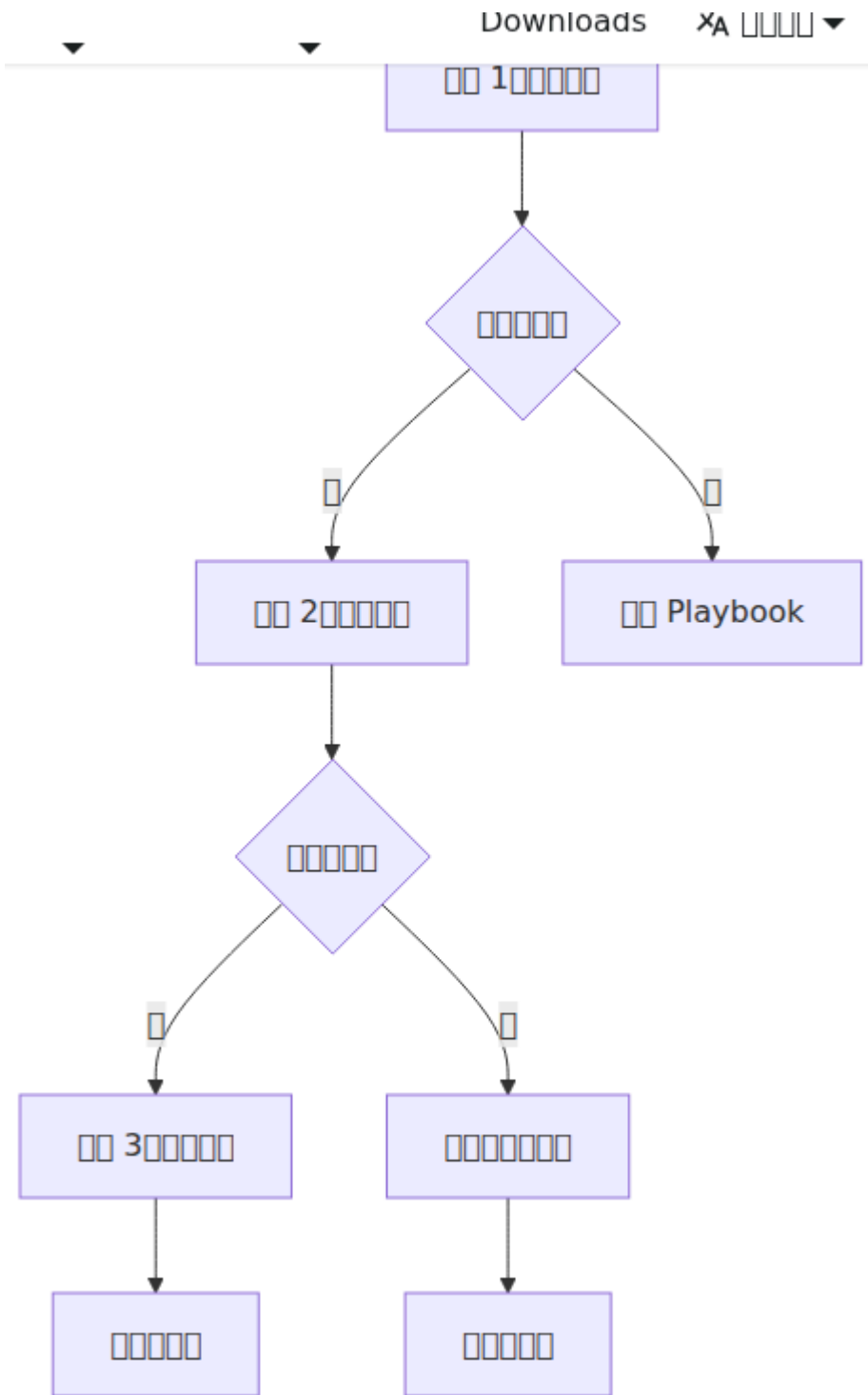
```
# 定義する fact 名は wholesale_cost あるいは retail_cost
- name: 設定する fact
  set_fact:
    charge_amount: >-
      {% if customer_info.json.is_wholesale_partner %}
      {{ api_response_product.json.wholesale_cost | float }}
      {% else %}
      {{ api_response_product.json.retail_cost | float }}
      {% endif %}
```

ansible-playbook ansible-playbook CRM 実行/実行
ansible-playbooks ansible-playbooks

Ansible

Ansible

□□□□



□□□□

□□□□□□□□□□

□□ **1**□□□ - □□□□□□□□□□

□□ **2**□□□ - □□ OCS □□/□□□□

00 **3**000 - 0000000000

00 - 0000000000

0000

00000 play_topup_charge_then_action.yaml 000000

```

- name: Play Topup - Charge card then action the Topup
  hosts: localhost
  gather_facts: no
  become: False

tasks:
  - name: [] crm_config []
    ansible.builtin.include_vars:
      file: "../..//crm_config.yaml"
      name: crm_config

# []
- name: [] 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: [] 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_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 }}"

# =====
- name: =====
  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: ===== payment_method_id
  set_fact:
    payment_method_id: "{{ api_response_payment_methods.json |
json_query(query) }}"
  vars:
    query: "data[?is_default==`true`].payment_method_id | [0]"

# =====
# =====
# =====

- name: "==== 1=====
  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: 00000000
 - assert:
 - that:
 - api_response_authorization.status == 200
 - api_response_authorization.json.success == true

```

- name: 00 authorization_id 0000/00
  set_fact:
    authorization_id: "{{
api_response_authorization.json.data.authorization_id }}"

```

```
# 00 200CS 00000000000000000000
```

- block:
 - name: "00 2000 CGRateS 00"
 - 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: OK
  assert:
    that:
      - action_execute_response.status == 200
      - action_execute_response.json.result == "OK"

# 3 - 
- name: "3"
  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:
    that:
      - api_response_capture.status == 200
      - api_response_capture.json.success == true

rescue:
  # 
  - name: "3"
    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: 创建 playbook 失败
fail:
  msg: "OCS 创建失败 {{ authorization_id }} 失败原因"

```

API 接口

1. 创建/更新支付记录
 2. 创建/更新支付记录 \$1 到 \$10 的支付记录
 3. \$9 的支付记录 创建/更新支付记录

1. 创建/更新支付记录

POST /crm/payments/authorize/hold

创建/更新支付记录

创建/更新支付记录 \$150 到 \$500 的支付记录 \$350 的支付记录
 创建/更新支付记录

创建/更新支付记录

```
{
  "customer_id": 123,
  "amount": 49.99,
  "currency": "AUD",
  "payment_method_id": 456,
  "metadata": {
    "description": "商品",
    "service_id": 789,
    "site_id": 12,
    "product_id": "34",
    "user_id": 5,
    "title": "商品",
    "wholesale_cost": 25.00,
    "invoice": true,
    "contract_days": 30,
    "send_email": true
  }
}
```

商品

- **customer_id**商品 - 商品 ID
- **amount**商品 - 商品
- **currency**商品 - 商品
- **payment_method_id**商品 - 商品
- **metadata**商品 - 商品
 - **description** - 商品
 - **service_id** - 商品
 - **site_id** - 商品
 - **product_id** - 商品
 - **user_id** - 商品
 - **title** - 商品
 - **wholesale_cost** - 商品
 - **invoice** - 商品 `true`商品
 - **contract_days** - 商品
 - **send_email** - 商品 `true`商品

000

```
{
  "success": true,
  "message": "000000000000",
  "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": "0000 $3500000000000000 $500 000000"
  }
}
```

000

- 00 authorization_id 0000000000000000
- 00 card_amount 0000000000
- 000000000000000000000000

2. 0000

000 POST /crm/payments/capture/{authorization_id}

00000000000000000000000000000000 invoice: true 0000000000


0000

```
{
  "metadata": {
    "provisioning_status": "success",
    "cgr_action": "Action_Topup_Standard",
    "additional_info": "00000000"
  }
}
```

□□□

```
{
  "success": true,
  "data": {
    "payment_id": "pay_xyz789",
    "transaction_id": 1234
  }
}
```

□□□□□□

1. □□□□□□□□□□□□□□□□
2. **□□□**□□□□□□
3. □□□□□□□□□□
4. □□□□□□□□□□ `invoice: true`
 - □□□□□□□□□□ = □□□
 - □□□□□□□□□□□□
 - □□□□□□□□□□ = □□□□□□
 - □□□□□□□□□□□□□□□□□□□□□□
 - □□□□□□□□□□□
5. □□□□□□□□□□□
6. □□ `send_email: true`□□□□□□□□□□□□□□□□

□□□\$500 □□□ \$150 □□□□□□

- □□□□\$350
- □□□□□+\$350□□□□□□□ \$500□
- □□□□□-\$500□□□□□□□
- □□□□□\$0

□□□□□□□□□□ □□□□□□□□□□

3. □□□□□□□

□□□ `POST /crm/payments/release/{authorization_id}`

API 调用失败时返回的响应体示例

响应体

```
{  
  "metadata": {  
    "release_reason": "provisioning_failed",  
    "error_details": "OCS 配置失败"  
  }  
}
```

成功

```
{  
  "success": true,  
  "message": "成功"  
}
```

故障排除

1. 检查网络连接
2. 检查 API 端点是否正确
3. 检查请求头是否正确
4. 检查请求体是否正确
5. 检查 API 文档

更多 API 调用示例请参考 API 文档

API 调用

API 调用示例

API 调用

PUT /crm/transaction/

play_simple_service.yaml

```

- name: 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 }} - ",
        "description": " {{ 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:
      - api_response_transaction.status == 200

```

- **customer_id** - ID
- **service_id** - ID
- **title** -
- **description** -
- **invoice_id** - ID null
- **wholesale_cost** -
- **retail_cost** -
- **site_id** - ID

- **tax_percentage** -

-
-
-
-
-

-
- /
-

```
pro_rata_charge = (monthly_cost * days_remaining) / days_in_month
```

playbook

```

# 000000000000
# 15 00000000 1
# 000000000000

- name: 0000000000
  command: "date +%d"
  register: current_day

- name: 0000000000
  command: "date -d 'last day of this month' +%d"
  register: days_in_month

- name: 0000000000
  command: "date -d 'last day of this month' +%Y-%m-%d"
  register: last_day_of_month

- name: 00000000
  set_fact:
    days_remaining: "{{ (days_in_month.stdout | int) -
(current_day.stdout | int) + 1 }}"

- name: 00000000
  set_fact:
    pro_rata_cost: "{{ ((monthly_cost | float) * (days_remaining |
float) / (days_in_month.stdout | float)) | round(2) }}"

- name: 00000000
  debug:
    msg:
      - "00000000$ {{ monthly_cost }}"
      - "00000000$ {{ days_in_month.stdout }}"
      - "00000000$ {{ days_remaining }}"
      - "00000000$ {{ pro_rata_cost }}"

# 000000000000 pro_rata_cost
- name: "00000000"
  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 }} ({{ days_remaining }}
    )",
      "description": "  {{ days_remaining }}/{{
days_in_month.stdout }}  {{ package_name }}  ",
      "service_id": "{{ service_id | int }}",
      "invoice": true
    }
  }
}
```

XXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

```

- name: 0000000000000000
  set_fact:
    service_start_date: "2024-01-15"
    next_billing_date: "2024-02-01"

- name: 0000000000
  shell: |
    echo $(( ( $(date -d "{{ next_billing_date }}" +%s) - $(date -
d "{{ service_start_date }}" +%s) ) / 86400 ))
  register: days_until_billing

- name: 0000000000000000 30
  set_fact:
    billing_period_days: 30

- name: 00000000
  set_fact:
    pro_rata_cost: "{{ ((monthly_cost | float) *
(days_until_billing.stdout | float) / (billing_period_days |
float)) | round(2) }}"

- name: 0000
  debug:
    msg:
      - "000000{{ service_start_date }}"
      - "00000000{{ next_billing_date }}"
      - "0000000000{{ days_until_billing.stdout }}"
      - "00000000${{ pro_rata_cost }}"

```

0000000000

00 1000000

- 000 1 0 15 000
- 000000\$60.00
- 1 00000031
- 00000017000 15 00 31 00
- 00000000 $\$60.00 \times 17 \div 31 = \32.90

00 2000000

- 10 個
- \$30/個
- \$50/個
- 30
- 21
- \$20/個
- $\$20.00 \times 21 \div 30 = \14.00

1. 1. 1. 1.

1. 1. 1. 1.

1. 1. 1. 1.

```

- block:
  # 1. 1. 1. 1.
  - name: 1. 1. 1. 1.
    uri: ...

  # 1. 1. 1. 1.
  - name: 1. 1. 1. 1.
    uri: ...

rescue:
  # 1. 1. 1. 1.
  - name: 1. 1. 1. 1.
    uri: ...

  - name: 1. 1. 1. 1. playbook
    fail:
      msg: "1. 1. 1. 1."

```

2. 1. 1. 1. 1.

1. 1. 1. 1.

```

- name: 0000
  uri: ...
  register: api_response_authorization

- name: 0000000
  assert:
    that:
      - api_response_authorization.status == 200
      - api_response_authorization.json.success == true
    fail_msg: "00000000{{ api_response_authorization.json }}"

```

3. 0000 ID

0000 authorization_id 0000/00000

```

- name: 00 authorization_id 0000/00
  set_fact:
    authorization_id: "{{
api_response_authorization.json.data.authorization_id }}"

```

4. 000000000

00000000000000000000

```

metadata:
  description: "00000000000"
  service_id: "{{ service_id | int }}"
  product_id: "{{ product_id }}"
  user_id: "{{ initiating_user | int }}"
  title: "00000000"
  wholesale_cost: "{{ wholesale_cost | float }}"
  invoice: true # 000000000000
  send_email: true # 000000

```

5. 000000000

000000000000000000000000

```
- name: 000000000000
  set_fact:
    final_cost: "{{ (base_cost | float * multiplier | float) |
round(2) }}"
```

6. 000000000000

000000000000000000000000000000000000

```
- name: 000000000000
  set_fact:
    payment_method_id: "{{ api_response_payment_methods.json |
json_query(query) }}"
  vars:
    query: "data[?is_default==`true`].payment_method_id | [0]"

- name: 000000000000
  assert:
    that:
      - payment_method_id is defined
      - payment_method_id != ""
      - payment_method_id != None
    fail_msg: "000000 {{ customer_id }} 00000000"
```

00000

00 **1**000000000000

000000000000000000

1. 000000000000
2. 00000
3. 0 OCS/CGRateS 000000
4. 0 CRM 0000000000
5. 00000
6. 0000000000000000

000 play_topup_charge_then_action.yaml 00000000

00 20000000000000

00000000000000000000

1. 0000
2. 0000000
3. 000000000000000000000000
4. 0000000000000000

000 play_simple_service.yaml 00 202-232 0000000000

00 3000000/00000

0000000000000000

1. 0000000
2. 00 CGRateS 00
3. 0000000
4. 00000000000

00 4000 ActionPlan 0000000

00000000000

1. 0000 *http_post 000000000000
2. 0000 *monthly 000 ActionPlan
3. 0 ActionPlan 000000
4. CGRateS 000000000000
5. 00 playbook 0000000

□□□□

□□□□

□□□ □□□□□□□□

□□□□

- □□□□□□□□□□
- □□□□
- □□□□□□
- □□ ID □□□

□□□□□ □□□□□□□□□□□□□□

□□□□□□□□□□□□

□□□ □□□□□□□□□□□□

□□□ □□□□□□□□□□ - □□□□□□□□□□□□□□

□□□□

- □□□□□□□□□□ 7 □□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□

□□□□□□□□□□

□□□ □□□□□□□□□□□□

□□□ □□□□□□□□□□ `invoice: true`

□□□□□ □□□



- □□□□□□□□□□ `invoice: true`□□□

- 000000000000

00000000

000 000000000000

00000

- 000000000000/0000/00000
- 00000   00000000
- 000000

00000

- 00000000000000000000
- 000000000000
- 000000000
- 0000000000

00000000

0000

000000000000

1. 0000000 - 00000000/PayPal

```

- name: 請求
  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. 請求 - 請求

請求

- 請求
- 請求
- 請求

請求

請求

請求

- Stripe ACH
- PayPal PayPal

請求 playbooks 請求 Square Adyen Braintree

請求

- 請求 - 請求
- 請求 - Stripe/PayPal API

□□□□

□□□ **Playbook** □□□

- `concepts_ansible.md` - □□ playbook □□□□□
- `concepts_provisioning.md` - □□□□□□

□□□□□□

- □□□□ **API** □□ - □□□ API □□□□□□□□□□
 - □□□□ API
 - □□□□ API
 - □□□□□□
 - □□□□
 - □□□□□
- `payments_transaction.md` - □□□□□□□□□□
- `payments_process.md` - □□□□□□□□
- `basics_payment.md` - □□□□□□□□□□



OmniCRM

OmniCRM **RBAC**
view_customer create_inventory



RBAC

1. —
2. —
3. —
4. — “...”



- — OmniCRM
- — view_customer delete_product

-   — `Admin` `Support` `Finance`
- `admin` — `admin`

JWT API IP /

`admin`

OmniCRM `admin`

`admin`

admin - `admin`

admin - `Admin` `Support` `Finance` `CRM`

admin - `admin`

OmniTouch

□□□□□□□□□□□□□□□□

□□□□□□□□ Web UI □□ API □□□□□□□□□□

□□ Web UI

1. □□□□□□□□ - □□□□□□□□□□□□□□
2. □□“□□□□” - □□□□□□□□

3. □□□□□□□□
 - □□□ - □□□□□□□□□□□□□□
 - □□□□ - □□□□□□□□□□□□□□□□□□□□□
 - □□ - □□□□□□□□□□□□□□□□□□□□□□□□
 - □□ - □□□□□□□□□□
 - □□□ - □□□□□□□□□□□□
 - □□ - □□□□□□□□□□
 - □□□□ - □□□□□□□□□□
 - □□ - □□□□□□□□□□□□□□□□□□□□□□
 - □□□□□ - □□□□□□□□□□□□□□□□□□□□□□
4. □□“□□□□” - □□□□□□□□□□□□□□□□□□□□
5. □□□□□□ - □□□□□□□□□□□□□□□□□□□□

□□□□□

- TempP@ssw0rd! (password)
- (username)
- (email)
- (phone number)

API

(method)

POST /auth/users

(body) admin

(response)

```
{
  "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"
}
```

(response)

```
{
  "id": 123,
  "username": "john.smith",
  "email": "john.smith@company.com",
  "first_name": "John",
  "last_name": "Smith",
  "roles": ["Support"],
  "created": "2025-01-04T10:30:00Z"
}
```

(response)

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

```
{  
  "username": "jane.doe",  
  "email": "jane.doe@company.com",  
  "password": "TempP@ssw0rd!",  
  "first_name": "Jane",  
  "last_name": "Doe",  
  "role": "Support,Finance"  
}
```

□□□□□□□□□□□□□□□□□□□□

```
POST /auth/roles/{role_id}/users/{user_id}
```

□□□□□□□□□□

□□□□□□□□□□□□□□□□

```
GET /auth/users
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□

```
GET /auth/users/search?search={query}&filters={"role":  
["Support"]}&page=1&per_page=50
```

□□□□□□□□□□

- □□□□
- □□□□□□
- □□/□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□

API

POST /auth/roles/{role_id}/permissions

```
{
  "permission_id": 123
}
```

□□□□□□□□

```
{
  "permission_ids": [123, 124, 125]
}
```

□□□□□“□□□□”□□

□□□□□□□□□□□□□□□□□□

1. □□□□□

```
POST /auth/roles
{
  "name": "Provisioning_Specialist",
  "description": "Can provision services and manage customer
services"
}
```

2. □□□□□

```
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
  ]
}
```

□□□□□□□□

□□Web UI□

1. □□□□□□□□
2. □□□□□□□□□□□□□□□□**“X”□“□□”**□□
3. □□□□

□□API□

□□□ DELETE /auth/roles/{role_id}/permissions/{permission_id}

□□□

```
DELETE /auth/roles/45/permissions/31
```

□□□□□□□□ create_provision □□□

□□□□□□□□

□□□□□□□□

□□Web UI□

1. □□□□□□□□ → □□□□□□
2. □□□□□□□□
3. □□□□□□□□
4. □□**“□□”**

API

PUT /auth/roles/{role_id}

```
{  
  "name": "Senior_Support",  
  "description": "Senior support team with full customer access"  
}
```

API

DELETE /auth/roles/{role_id}

データベースに「ユーザー」を追加

データベース

Web UI

1. ユーザーID → ユーザー
2. ユーザーID
3. ユーザーID/パスワード
4. ユーザーID**パスワード**

API

データベース

PUT /auth/users/{user_id}

```
{  
  "role": "Support,Finance"  
}
```

データベース

POST /auth/roles/{role_id}/users/{user_id}

API

API

GET /auth/roles/{role_id}/users

API

API

GET /auth/users/{user_id}

API roles API

API

OmniCRM API

API

API

1. API “API”
2. API - API
3. API - API 1
4. API - API
5. API - API
 - o 8
 - o API
 - o API
 - o API
 - o API
6. API - API

API

1. API

POST /auth/forgot_password

```
{  
  "email": "user@example.com"  
}
```

Response

2. Reset Password

POST /auth/reset_password

```
{  
  "token": "abc123...",  
  "new_password": "NewSecureP@ssw0rd!"  
}
```

Web UI

Implementation

Web UI

1. User clicks "Forgot Password" →
2. User enters email address
3. User clicks "Send Reset Link"
4. User receives email with reset link
5. User clicks reset link and is redirected to reset password page
6. User enters new password and confirms it

API

POST /auth/users/{user_id}/admin_reset_password

Request Body: admin

Response

```
{
  "new_password": "TempP@ssw0rd!",
  "force_change": true
}
```

□□□

- `new_password` - □□□□□□□□
- `force_change` □□□□ - □□□true□□□□□□□□□□□□□□□□

□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□ `POST /auth/change_password`

□□□

```
{
  "current_password": "OldP@ssw0rd!",
  "new_password": "NewSecureP@ssw0rd!"
}
```

□□□□□□□□□□□□□□□□

□□□□□




- □□□□bcrypt□werkzeug□□□□□□□□□□□□
- □□□□□□□□
- □□□□□1□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□

OmniCRM 2FA

OmniCRM 2FA

2FA

Web UI

1. →
2.   
3. **"2FA"**
4.
 - o TOTP QR
 - o 10
5. Google Authenticator Authy QR
- 6.
- 7.
8. 2FA

API

1. TOTP

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

```
{
  "totp_secret": "JBSWY3DPEHPK3PXP",
  "qr_code_url": "otpauth://totp/OmniCRM:user@example.com?secret=JBSWY3DPEHPK3PXP&issuer=OmniCRM",
  "backup_codes": [
    "12345678",
    "23456789",
    "34567890",
    ...
  ]
}
```

2. 验证

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

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

返回 {"verified": true}

2FA 实现

实现 2FA 的步骤

1. 生成 TOTP 密钥
2. 生成 QR 码
3. 用户扫描二维码设置 2FA
4. 验证用户输入的验证码
5. 生成备用码
6. 存储备用码

□□□□

- 2FA□□□□□□□□10□□□
- □□□□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□

□□2FA□□

□□ POST /2fa/verify/user/{user_id}

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

□□□□□□

- **TOTP**□□□□□□□□□□□□□□6□□□□
- □□□□□□□□□□□□□□8□□□□

バックアップコード

バックアップコードを生成する

Web UI

- バックアップコード
- バックアップコードを生成する
- バックアップコードを再生成する
- バックアップコード

API

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

レスポンス

```
{
  "backup_codes": [
    "98765432",
    "87654321",
    "76543210",
    ...
  ]
}
```

2FA

2FAを有効にする

Web UI

- バックアップコード → 2FA
- バックアップコード
- バックアップコードを生成する
- バックアップコード
- 2FAを有効にする
- バックアップコードを再生成する

API

POST /2fa/admin/disable/user/{user_id}

admin

2FA

- TOTP
-
- `is_2fa_enabled` `false`

2FA

2FA

POST /2fa/reset-for-new-device/user/{user_id}

```
{
  "backup_code": "12345678"
}
```

TOTP

2FA

- 2FA
-
-
- Google Authenticator, Authy, Microsoft Authenticator
- 2FA

- 2FA - 2FA

□□□□□□

□□□□□□◆◆◆□□□□□□

Web UI

1. □□□□□□ → □□
2. □□□□□□
3. □□□□□□□□
 - □□□□□□□□
 - □□□□□□□□□□
 - □□□□
 - □□
 - □□□□□□
4. □□**"□□"***

API

□□ PUT /auth/users/{user_id}

```
{
  "first_name": "Jane",
  "last_name": "Doe-Smith",
  "email": "jane.doesmith@newcompany.com",
  "phone_number": "+61498765432",
  "role": "Support,Finance"
}
```

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□

- pending_email □□□□□□□□
- □□□□□□□□□□
- □□□□□□□□

- email [redacted]
- email_verified [redacted]true

API

OmniCRM API - [redacted]

API

Web UI

1. [redacted] → [redacted]
2. [redacted]
3. [redacted]**"[redacted]"**[redacted]
4. [redacted]
5. [redacted]

API

[redacted] DELETE /auth/users/{user_id}

[redacted] admin

API

- deleted [redacted] True
- [redacted] deleted_at [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]

API

API

GET /auth/users/search?filters={"deleted":[true]}

API

API endpoint for updating user information

PUT /auth/users/{user_id}

```
{  
  "deleted": false  
}
```

API endpoint for deleting user information

DELETE /auth/users/{user_id}

UI

UI for user management

UI for user management, including GDPR compliance

API

- API endpoint - GET /users
- API endpoint - POST /users
- API endpoint - PUT /users/{user_id}
- API endpoint - DELETE /users/{user_id}
- API endpoint - GET /users/{user_id}
- API endpoint - PUT /users/{user_id}
- API endpoint - DELETE /users/{user_id}

API

API endpoint for CRUD operations

- view_* — GET /users
- create_* — POST /users
- update_* — PUT /users/{user_id}

- delete_* — 删除/删除

权限名称**“权限名称...”**权限名称/权限

删除/删除

- `admin` — 权限名称/权限
- `can_impersonate` — 权限名称/权限
- `grafana_access` — Grafana权限名称/权限
Grafana权限

权限名称/权限

- 权限
 - view_customer[] create_customer[] update_customer[] delete_customer
 - 权限名称 权限名称
- 权限
 - view_customer_site[] create_customer_site[] update_customer_site[] delete_customer_site
 - 权限名称 权限名称
- 权限

- view_customer_contact[] create_customer_contact[]
update_customer_contact[] delete_customer_contact
- 管理属性 *Administration Attributes*
- 管理属性 *Administration Attributes* <administration_attributes>[]
 - view_customer_attribute[] create_customer_attribute[]
update_customer_attribute[] delete_customer_attribute
 - 管理属性 *Administration Attributes*
- 管理属性 *Administration Tags* <administration_tags>[]
 - view_customer_tag[] create_customer_tag[] update_customer_tag[]
delete_customer_tag
 - 管理属性 *Administration Tags*
- 管理属性 *Administration Services*
 - view_customer_service[] create_customer_service[]
update_customer_service[] delete_customer_service
 - 管理属性 *Administration Services*
- 管理属性 *Administration Activity*
 - view_customer_activity[] create_customer_activity[]
update_customer_activity[] delete_customer_activity
 - 管理属性 *Administration Activity*



- **Stripe**
 - view_customer_stripe_card[] create_customer_stripe_card[]
update_customer_stripe_card[] delete_customer_stripe_card
 - 管理属性 *Administration Stripe*
- 管理属性 *Administration Transactions*
 - view_customer_transaction[] create_customer_transaction[]
update_customer_transaction[] delete_customer_transaction
 - 管理属性 *Administration Transactions*
- 管理属性 *Administration Invoices*
 - view_customer_invoice[] create_customer_invoice[]
update_customer_invoice[] delete_customer_invoice
 - 管理属性 *Administration Invoices*

□□

- view_communication □ create_communication □ update_communication □ delete_communication
- □□□□ □□□□□□□□

□□□□□

- □□
 - view_inventory □ create_inventory □ update_inventory □ delete_inventory
 - □□□□ □□□□□□□□
- □□□□
 - view_inventory_template □ create_inventory_template □ update_inventory_template □ delete_inventory_template
 - □□□□ □□□□□□□□□□

□□

- view_product □ create_product □ update_product □ delete_product

□□□□ **CBC** □

- view_cbc_message □ create_cbc_message □ update_cbc_message □ delete_cbc_message

□□

- □□
 - view_provision □ create_provision □ update_provision □ delete_provision
 - □□□□ □□□□□□□□
- □□□□
 - view_provision_event □ create_provision_event □ update_provision_event □ delete_provision_event

“权限管理”

“权限管理...” 权限管理是系统中最重要的一部分，它决定了用户能做什么，不能做什么。权限管理的好坏，直接影响到系统的安全性和易用性。

权限管理

角色	权限	备注
管理员 角色	admin 角色 can_impersonate 权限 CRUD	系统/管理/权限管理
普通 用户	view_customer 权限 view_customer_service 权限 view_communication 权限 view_provision 权限	权限管理 can_impersonate
普通 用户	view_customer_invoice 权限 view_customer_transaction 权限 view_product 权限 create_customer_invoice	权限管理
普通 用户 角色 用户	“权限管理...” 权限管理是系统中最重要的一部分，它决定了用户能做什么，不能做什么。	权限管理
普通 用户 角色	权限 view_*	系统/管理/权限

权限管理是系统中最重要的一部分，它决定了用户能做什么，不能做什么。

API 权限管理

权限管理 admin 角色

权限

□□□ GET /auth/permissions

□□□□□□□□

□□□ POST /auth/permissions

□□□□

```
{  
  "name": "view_example",  
  "description": "Read-only access to example objects"  
}
```

□□□□

□□□ GET /auth/roles

□□□□

□□□ POST /auth/roles

□□□□

```
{  
  "name": "Support",  
  "description": "Tier-1 support team"  
}
```

□□□□□□□□

□□□ POST /auth/roles/{role_id}/permissions

□□□□

```
{  
  "permission_id": 123  
}
```

□□□□□□□□

DELETE /auth/roles/{role_id}/permissions/{permission_id}

□□□□□□□□

□□□□□□□□

POST /auth/users

□□□□

```
{
  "username": "sara",
  "email": "sara@example.com",
  "password": "TempP@ssw0rd!",
  "first_name": "Sara",
  "last_name": "Ng",
  "phone_number": "+61...",
  "role": "Support"
}
```

□□□□□□□□

PUT /auth/users/{user_id}

□□□□

```
{
  "role": "Finance"
}
```

□□□□□□□□□□

GET /auth/users

□□□□□□

- can_impersonate admin

□□□□

POST /auth/impersonate

Request

```
{ "user_id": 42 }
```

Response

POST /auth/stop_impersonation

Notes

- Requires the impersonation permission
- Returns "impersonated..." in the response
- Requires the impersonation permission
- Requires the `admin` or `can_impersonate` permission

Grafana

OmniCRM Grafana `grafana_access` Grafana

Grafana

`grafana_access` `/grafana` Grafana
OmniCRM - Grafana

Grafana

- Grafana
- Grafana
- Grafana
- Grafana

OmniCRM

Grafana OmniCRM CRM

1. **Grafana** - `/grafana`
2. **ID** - Grafana URL `/grafana/d/abc123/dashboard-name` ID `/d/` `abc123`
3. - ID
4. - OmniCRM “” OmniCRM

- **ID** - Grafana URL
- - OmniCRM

Grafana

`grafana_access` Grafana

-
-
-
-

□□□□□□□□ □□□□□□□□□□□□□□□□

“□□□□...”□□□□□□□□□□ □□□□□□/□□□□□□□□□□□□□□□□

□□□□□□□ **Grafana**□□□□ □□□□□□□ grafana_access □□□□□□ /grafana □□□□□□
Grafana□□□

- 00 - 0000000000000000

000000

0000000000000000000000

000000

00000000

- 00000000000000
- 0000
- 00
- 0000

000000

- 00ID
- 000000
- 00000000/0000
- 000000

□□□□□

□□□□□□

- □□□□ - □□□□□□□□□□“□□ - +44 7700 900123”□
- □□ - □□□□□□
- □□ - □□□□□□□□□□
- □□□□ - □□□□□□
- □□□□ - □□□□□□□□□□
- □□ - □□□□
- □□□□ - □□/□□□□

□□□□□

- □□□□□□ - □□□□□□□□□□□□□□□□□□□
- □□ - □□□□□□
- □□□□ - □□□□□□
- □□ - □□□□□□□□□□□□
- □□□□□□ - □□□□□□□□

□□□□□□□□

- □ □□ - □□□□□□
- □ □□□□ - □ <7 □□□□□□□□
- □ □□ - □□□□□□□□□□□□□□□□
- ● □□ - □□□□□□


□□□□□

□□□□□□□□□□□□□□□□

□□□□□

□□□□□□□□□□□□□□□□

□□□□□□

- □□□ - □□□□□□□□
- □□□□ - □□□□□□□□□□
- □□□ - □□□□□□□
- □□□□ - □□□□□□□□
- □□□ - □□□□□□□□□□□□

□□□□□

□□□□□□□□□□□□□□□□ + □□□□□□□□□□□□□□□□

- □□ **10** - □□□□□□□□□□□□□□□□
- □□ **20** - □□□□□□□□□□□□□□
- □□ **30** - □□□□□□□□□□□□□□□□

□□□□□

□□□□□□□□□□□□□□□□

□□□□□□

- □□□□ - □□□□□
- □□□□ - □□□□
- □□□□ - □□□□□□□□□□
- □□□□ - □□□□□□□□
- □□□□ - □□□□□□□□

□□□□□

- □□□□□ - □□□□□□□□
- □□□□□ - □□□□□□□□
- □□□□ - □□□□□□□
- □□□□ - □□□□□□

□□□□

□□□□□□□□□□□□

□□□□□□□□

- 0000 - 0000
- 0000 - 000000
- 00**MMS** - MMS000000
- 0000 - 0000000000

0000

0000000000000000

0000

- 0000 - 0000
- 0000 - 0000000000
- 0000 - 0000000000
- 0000 - 00/0000

0000

0000000000000000

00

0000000000000000

□□□□□□

- □□□□ - □□□□□□
- □□ - □□□□□□
- □□□□ - □□□□□□
- □□ - □□□□□□
- □□ - □□□□□□□□□□
- □□ - □□PDF□□□□□□

□□□□□

1. □□□ □□ → □□
2. □□□□□□□□□□□□
3. □□ “□□□□” □□
4. □□□□□□□□□□□□□□
5. □□□□
6. □□□□□□□□

□□□□□

1. □□□□□□ “□□” □□

2. PDF 生成

3. 印刷

□□

□□□□□□□□□□□□□□□□□□□□

□□□□□

- □□ - □□□□□□
- □□ - □□/□□□□□
- □□ - □□□□□□□□□□□□□□
- □□ - □□□□□□□
- □□ - □□□□□□□

□□□□□

□□□□□□□□□□□□□□□□□□□□

□□□□□□□

- □□□ - □□Stripe□□□□□□□□□□/□□□
- □□□□ - □□□□□□□□□
- □□□ - □□□□□□□□□□□
- □□□□□ - □□□□4□□□□□□□□□□□

□□□ □□□□ <payment_methods> □□□□□□□□□□□□□□□□

□□ / □□

□□□□□□□□□□□□□□□□□□□□

□□□□□

1. □□□ □□ → □□□□ → “□□□”
2. □□□□□□□□□□

3. 0000000000000000
4. 0000000000
5. 0000000
6. 0000
7. 0000000

000000

- 0000 - 1GB 5GB 10GB 20GB 50GB 00
- 0000 - 0000000
- 000 - 000
- 0000 - 000000 £5 £10 £20 £50 £100

000 000000 <features_topup_recharge> 0000000000000000

00000000

00000000000000000000

00000000

- 0000 - 0000000
- 00**IP**00 - 0000000000IP
- 0000 - IPTV0000000000
- 0000 - 00000000000000
- 0000 - 00000000

00000000

1. 000 00 → 0000 → “0000”
2. 000000000000000000
3. 000000000000 “000000”
4. 0000000000 + 0000
5. 00000
6. 0000000000

□□□ □□□□ - □□□□ <csa_modify> □□□□□□□□□□□□□□□□

□□□□□

□□□□□□□□□□□□□□□□

□□□□□□□

- □□□□□□□□□□
- □□□□□□□
- □□□□□□□□7□□3□□1□□
- □□□□□□□□□□□□□□□□
- □□□□□/□□
- □□□□□□□
- □□□□□□□□□□□□

□□□□□□□

- □□□□□□□
- □□□□□□□□
- □□□□□□□□□10%□
- □□□□□□□□
- □□□□□□□□□□□□□□□□□□□□

□□□□□

□□□□□□□

- □□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□□<20%□□□□□
- □□□□□□□□□□□□□

□□□□□□

□□□□□□□□◀▶

□□□□□□

- □□□□□□□□ ↔ □□□
- □□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□
- □□□□
- ◀▶▶□□□□□□
- □□□□□□□

□□□□□

- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□
- □□□□□□□□□
- □□□□□□□□□□□□□□□□□□

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□

□□□□□



1. □□□ □□ → □□□□
2. □□ □□□ □□□
3. □□□□□□□□□□□□
4. □□ “□□□□□□□□” □□
5. □□□□□□□□□□□□□□□□
6. □□□□□□□□□□□□□□
7. □□□□□□□□□□□□□□□□

□□□

- □□□□ - □□□□□□□□□□
- □□ - □□□□□□□□□□□□
- □□ - □□□□□□
- □□ - □□□□□□□□□□□□□□□□

□□□□□□

- □□ `can_impersonate` □ `admin` □□
- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□

□□□ □□□□ - □□□□ `<customer_care>` □□□□□□□□□□

□□□□□□

□□□□□□□□□□□□□□□□□□□□

□□□□□□

- 網路連線
- 網路存取
- 網路存取權限
- 網路存取權限
- 網路eSIM存取權限
- 網路存取
- GPS存取權限

網路Web存取 (PWA)

- 網路存取權限
- 網路存取權限
- 網路存取權限
- 網路存取權限

網路存取權限

網路存取權限

網路存取

1. 網路存取權限 “網路存取”
2. 網路存取權限
3. 網路存取權限100%
4. 網路存取權限
5. 網路存取權限
6. 網路存取權限

網路存取

- 網路80%
- 網路存取權限
- 網路存取權限
- 網路存取
- 網路存取權限 (!@#\$%^&*)

□□□□

□5□□□□□□

- □□□□30□□
- □□□□□□□□□□
- □□□□□□□□□□

□□□□□□

□□□□

1. □□2FA□□□□□□
2. □□□□□□□□□□
3. □□□□□□□□□□□□
4. □□□□□□□□□□□□
5. □□□□□□□□□□□□
6. □□□□□□□□□□□□□□2FA□
7. □□□□/□□□□□□□□

□□□□

1. □□□□□□□□□□10-20%□
2. □□□□□□□□□□□□
3. □□□□□□□□□□□□
4. □□□□□□□□□□□□
5. □□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□□□□

1. □□□□□□/□□□□□□□□□□
2. □□□□□□□□□□□□
3. □□□□□□□□□□□□□□□□
4. □□□□ID□□□□□□□□□□□□

API

API

API

1. API
2. "API"
- 3.
- 4.
5. API

API

-
-
-
-
-

API <concepts_api>

"API"

OmniCRM 目次

OmniCRM は Omnitouch を活用した BSS/OSS の

目次

目次

目次

1. 目次 - 目次
2. 目次 - 目次
3. 目次 - 目次
4. 目次 - 目次
5. 目次 - 目次

目次

- 目次 - 目次
- 目次 - 目次
- 目次 - 目次
- 目次 - 目次

目次

目次

1. 目次 - 目次
2. 目次 - 目次
3. 目次 - 目次
4. 目次 - 目次
5. 目次 - 目次

□□□□

- □□□□ - □□□□
- □□□ - □□□□□□□□□□
- **API** □□ - □□□□□□
- □□□□ - □□ 2FA □□□

□□□□

□□□□□□□□□□

- □□□□□□ - □□□□□
 - □□□□ - □□□□□□□□□□
 - □□□□□□ - □□□□□□□□
 - □□□□ - □□□□□
 - □□□□ - □□□□□□□
-

□□□□□□

□□□...	□□
□□□□□	□□□□
□□□□□□□	□□□□
□□□□□□□□□□	□□□□□□
□□□□	□□□□
□□□□	□□□□
□□□□	□□□□□
□□□□	□□□□
□□□□	□□□□□□
□□□□□□	Ansible □□
□□□□□□	RBAC
□□ 2FA	□□□□□
□□□□□□	□□□□
□□□□	□□□□
□□ API	API □□

□□□

□□□□

- □□□□ - □□□□□□□□□□
- □□□□□□□□ - □□□□□□□□
- **Ansible** □□□□ - □□□□□□□□

□□□□

- □□□□□□□□□□ - □□□□□□□□
- □□□□ - □□□□□□□□
- □□□□ - □□□□□□□□□□
- □□□□ - □□□□□□□□
- □□□□ - □□□□□□□□

□□□□

- □□□□ - □□
- □□□□ - □□□□□□□□
- □□□□ - □□□□□
- □□□□ - □□□□□
- □□□□□□□□□□ - □□□□□□□□
- □□□□□□ - □□□□□

□□□□□

- □□□□□ - □□□□□□□□□□
- □□□□□ - □□□□□□□□
- □□ - □□□□□□□
- □□□□□ - □□□□□□□□
- □□□□□ - □□□□□□□□□□

API

- [OmniCRM API](#) - OmniCRM API
- [API](#) - API
- [API](#) - API
- [API](#) - API

API

- [API](#) - API
- [API](#) - API 2FA
- [API](#) - API

API

- [Stripe API](#) - API
- [Mailjet API](#) - API
- [API](#) - REST API

API

- [API](#) - API
- [API](#) - API
- [API](#) - API

API

- [API](#) - API
- [API](#) - API

OmniCRM API

OmniCRM API

📄📄📄📄📄

📄📄📄📄 - OmniCRM 📄 **Ansible** 📄📄 📄📄📄📄📄📄📄📄📄📄📄 SIM 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

📄📄📄📄

- 📄📄📄📄
- 📄📄📄📄📄📄
- 📄📄📄📄📄📄
- 📄📄📄📄📄📄📄📄📄📄
- 📄📄📄📄📄📄📄

📄📄📄📄📄📄 | 📄 **Ansible** 📄📄

📄📄📄📄📄

📄📄 **CGRateS** 📄📄📄📄📄📄📄📄📄 **Stripe** 📄📄 📄📄📄📄📄

📄📄📄📄

- 📄📄📄📄📄📄📄
- 📄📄📄📄
- 📄📄 Stripe 📄📄📄📄
- 📄📄📄📄📄📄📄
- 📄📄📄📄📄📄
- 📄📄📄📄📄

📄📄📄📄 | 📄📄📄 | 📄📄📄

📄📄📄📄📄

📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

📄📄📄📄

- 📄📄📄📄📄📄📄📄
- 📄📄📄📄📄📄

- 000000000000
- 00000000
- 000000000000
- Google 000000000000

000000 | 0000 | 0000

0000

0000000000000000 - 0 SIM 00000000 IP 0000

00000

- 00 SIM 00000000
- 00000000 IP 0000
- 000000000000
- 000000000000
- 00000000
- 00000000

000000

0000000000

00 000000 0000000000000000

000000000

- 0000000000
- 0000000000
- 00000000
- 00 PDF 000000
- 00000000
- 0000000000

00000000 | 00000000

API

API 的基礎知識

認證

- **JWT** 認證
- **(2FA)** 認證 TOTP 認證
- **(RBAC)** 認證
- 認證
- 認證
- 認證

認證 | **RBAC** 認證 | **2FA** 認證

集成

OmniCRM 集成

- **CGRateS** - 集成
- **Stripe** - 集成
- **Mailjet** - 集成
- **Google** 廣告 - 集成
- **RESTful API** - 集成

API 集成 | **Stripe** 集成 | **Mailjet** 集成

部署

部署的基礎知識

部署

- 部署 - 部署
- 部署 - 部署 + 部署 + 部署
- 部署 - 部署
- 部署 - 部署

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXX | XXXXXXXX

XXXXXXXXXX

XXXXXXXXXX OmniCRM 与 XXXXXXXX XXXXXXXXXXXX

XXXXXX

- XXXXXXXXXXXXXXXXXXXX
- XXXXXXXX
- XXXXXXXXXXXXXXXXXXXX
- XXXXXXXX

XXXXXXXXXX

XXXXXX ? ? XXXXX

与 XXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXX

- XXXXXXXX
- XXXXXXXXXXXX
- XXXXXXXXXXXX
- XXXXX

XXXXX | XXXXX

□□□□□

□□ Ansible □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□

□□□□□□□□□□□□□□ - □□□□□□□□□□□ OmniCRM□

□□□□□

□□□□□□□□□□□ CGRateS □□□□□□□□□□□□□□□□

□□□□

JWT □□□□2FA□RBAC □□□□□□□□□□□□□□□□□□□□

□□ **API**

□□□ REST API □□□□□□□ OmniCRM □□□□□□□□□□□□□□□□

□□□□□

□□□□

- □ □□□□ □□□□□□□□□□
- □□ □□□□□□□□□□□□□□□□
- □□ □□□□□□□□□□□□□□
- □□ □□□□□□□□□□

□□□□□□□

□□□□□□□□□□□□ □□□□□ □□□□□□□□

OmniCRM - ██████████ BSS/OSS

██████2025-12-23