

OmniCRM □□□□

OmniCRM □ Omnitouch □□□□□□□□□□□□□□□□ BSS/OSS □□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□ - □□□□□□□□□□□□□□□□

□□

□□□□□□□□

□□□□□□

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

□□□□□□

- □□□□ - □❓❓❓□□□□□□
- □□□□□□□□ - □□□□□□□□□□
- □□□□ - □□□□□□□
- □□□□ - □□□□□□□□

□□□□□□□□

□□□□□□

1. □□□□ - □□□□□
2. □□□□ - □□□□□
3. □□□□ - □□□□□□□
4. □□□□ - □□□□□□

5. 安全 - 认证与授权

认证

- 认证 - 验证
- **API 认证** - 认证
- 认证 - 双因素认证

授权

认证与授权

- 认证与授权 - 认证
 - 认证 - 认证
 - 认证与授权 - 认证
 - 认证 - 认证
 - 认证 - 认证
-

□□□□□□

□□...	□□
□□□□□	□□□□
□□□□□□□	□□□□
□□□□□□□□□□	□□□□□□
□□□□	□□□□
□□□□	□□□□
□□□□	□□□□□
□□□□	□□□□
□□□□	□□□□□□
□□□□□□	Ansible □□
□□□□□□	RBAC
□□ 2FA	□□□□□
□□□□□□	□□□□
□□□□	□□□□
□□ API	API □□

□□□

□□□□

- □□□□ - □□□□□□□□□□
- □□□□□□□□ - □□□□□□□□
- **Ansible** □□□□ - □□□□□□□□

□□□□

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

□□□□

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

□□□□□

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

API

- [API 2FA](#) - OmniCRM
- [API 2FA](#) - API
- [API 2FA](#) - API

API

- [API 2FA](#) - API
- [API 2FA](#) - 2FA
- [API 2FA](#) - API

API

- [Mailjet API](#) - API
- [API REST](#) - REST API

API

- [API REST](#) - API
- [API REST](#) - API
- [API REST](#) - API

API

- [API REST](#) - API

OmniCRM API 2FA

OmniCRM API 2FA

📁📁📁📁📁

📁📁📁📁 - OmniCRM 📁 **Ansible** 📁📁 📁📁📁📁📁📁📁📁📁📁📁 SIM 📁📁📁📁📁📁📁📁📁📁📁📁

📁📁📁📁

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

📁📁📁📁📁📁 | 📁 **Ansible** 📁📁📁

📁📁📁❓❓📁📁

📁📁 **CGRateS** 📁📁📁📁📁📁📁📁📁 Stripe 📁📁📁📁📁📁

📁📁📁📁

- 📁📁📁📁📁📁📁
- 📁📁📁📁
- 📁📁 Stripe 📁📁📁📁
- 📁📁📁📁📁📁📁
- 📁📁📁📁📁📁📁
- 📁📁📁📁📁

📁📁📁📁 | 📁📁📁 | 📁📁📁

📁📁📁📁📁

📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁

📁📁📁📁

- 📁📁📁📁📁📁📁📁
- 📁📁📁📁📁📁📁

- 繁體中文
- 繁體中文
- 繁體中文
- Google 繁體中文

繁體中文 | 繁體中文 | 繁體中文

繁體中文

繁體中文 - SIM 繁體中文 IP 繁體中文

繁體中文

- SIM 繁體中文
- 繁體中文 IP 繁體中文
- 繁體中文
- 繁體中文
- 繁體中文
- 繁體中文

繁體中文

繁體中文

繁體中文 繁體中文 繁體中文

繁體中文

- 繁體中文
- 繁體中文
- 繁體中文
- PDF 繁體中文
- 繁體中文
- 繁體中文

繁體中文 | 繁體中文

API

API 的基礎知識

API 安全

- **JWT** 認證
- **(2FA)** 認證 TOTP 認證
- **(RBAC)** 權限管理
- 數據加密
- 數據完整性
- 數據可用性

API 安全 | **RBAC** 認證 | **2FA** 認證

API 集成

OmniCRM 集成

- **CGRateS** - 數據集成
- **Stripe** - 支付集成
- **Mailjet** - 郵件集成
- **Google** 廣告 - 廣告集成
- **RESTful API** - 數據集成

API 集成 | **Mailjet** 集成

API 測試

API 測試工具 - 測試 API 的基礎知識

API 測試

- 測試 - 測試 API 的基礎知識
- 測試 - 測試 API 的基礎知識 + 測試 + 測試
- 測試 - 測試 API 的基礎知識
- 測試 - 測試 API 的基礎知識

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXX | XXXXXXXX

XXXXXXXXXX

XXXXXXXXXX OmniCRM 与 XXXXXXXX 紧密集成

XXXXXX

- XXXXXXXXXXXXXXXXXXXX
- XXXXXXXX
- XXXXXXXXXXXXXXXXXXXX
- XXXXXXXX

XXXXXXXXXX

XXXXXXXXXX

与 XXXXX 无缝集成，提升效率

XXXXXX

- XXXXXXXX
- XXXXXXXX
- XXXXXXXX
- XXXXX

XXXXX | XXXXX

OmniCRM - ██████████ BSS/OSS

██████2025-12-23

2FA (2FA)

2FA

OmniCRM 2FA (2FA) (TOTP) 2FA

RBAC <rbac> 2FA Authentication Flows <authentication_flows>

2FA

2FA

1. RBAC — RBAC
2. 2FA — 2FA
3. Authentication Flows — Authentication Flows
4. TOTP — Google Authenticator, Authy, Microsoft Authenticator, TOTP

2FA

2FA

1. RBAC — RBAC
2. 2FA — 2FA/6
3. Authentication Flows — Authentication Flows
4. TOTP — TOTP

2FA

2FA

2FA

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

2FA

2FA

2FA

- 2FA
- 2FA

- 2FA 2FA

2FA

2FA 2FA
totp_secret

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"  
}
```

□□□□□□□□□□□□□□□□□□

□□□□□□□□

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

□□□□□□□□□□□□□□□□

□□□□

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

□□□□□□□□

□□□□□□□□□□ □□□□ TOTP □□□□□□Google Authenticator□Authy□Microsoft Authenticator□1Password □□□

□□□□□□□□□□ □□□□□□□□□□□□□□ 2FA□

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 令牌是 API 的重要组成部分

API 概念

API

API

API `MANAGE_API_KEYS` 是 API 的重要组成部分

API 概念

API 是应用程序编程接口

API

- API 是应用程序编程接口“应用程序”“应用程序”

- **API** - **Application Programming Interface**
- **API** - **Application Programming Interface**
- **API** - **Application Programming Interface**
- **API** - **Application Programming Interface**
- **API** - **Application Programming Interface**
- **API** - **Application Programming Interface**

API

API

API

- **API** - **Application Programming Interface**
- **API** - **Application Programming Interface**
- **API** - **Application Programming Interface**
- **API** - **Application Programming Interface**

API

1 "API"

API +

2

API

API
"API"

API
- API

API [API]
- API

[] []

[]

[]

- []
- []
- []

[]

- []
- [] 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 调用失败的原因

可能的原因

- 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 - API 调用失败

注意

- 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 관리

API 키 관리

- API 키는 32 바이트로 생성
- API 키는 API 키로 생성
- API 키는 API 키로 생성 `sk_live_` API 키

API 키 관리

CRM API

- API 키
- API 키
- API 키
- API 키

API 키 관리

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

API 키 관리

- 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 000000

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

0000 API 00

- 000 000000000000
- 000000
 - 000000000000
 - 0000000000/00
 - 0000000000???

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"
}
```

□□□□

□□ **1**□□□□□□

□□□ Ansible □□□□□□□□ API □□□

1. □□□ API □□ → □□
2. □□□“Ansible □□□□□”
3. □□□“□□□◆◆□□□□”
4. □□□□□□□□/□□□□□□□□/□□□□
5. □□□1 □
6. □□□□□□□□ Ansible `crm_config.yaml`

□□ **2**□□□□□□□□

□□□□□□□□□□□□□□

1. □□□“□□□□ - QuickBooks”
2. □□□□□□□□□□□□□□□□
3. □□□90 □□□□□□□□
4. □□□□□□□□□□

□□ **3**□□□□□□

□ Prometheus/Grafana □□□□□□□□□□□□

1. □□□“□□ - Grafana”
2. □□□□□□□□□□□□
3. □□□□□□□□□□□□□□
4. □ Grafana □□□□□□

□□ **4**□□□□□ **API**

□□□□□□□□□□□□□□

1. □□□“□□□□□□□□”
2. □□□□□□□□□□□□□□□□□□□□
3. □□□180 □

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"`

`Network`

`Network`

- `radius_server_group_name = "radius"`
- `radius_server_group_ip = "10.10.10.10"`
- `radius_server_group_secret = "secret"`
- `radius_server_group_authentication_port = 1812`
- `radius_server_group_accounting_port = 1813`

`Network`

- `radius_server_group_name = "radius"`
- `radius_server_group_ip = "10.10.10.10"`
- `radius_server_group_secret = "secret"`
- `radius_server_group_authentication_port = 1812`

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

1. 创建数据库表
2. 插入数据
3. 查询数据
4. 更新数据
5. 删除数据
6. 备份数据库

1. 创建数据库表

创建数据库表时，需要注意表的命名规范，如使用小写字母、下划线等。

2. 插入数据

插入数据时，需要注意数据的完整性，如主键、外键等。

1. 创建数据库表
2. 插入数据
3. 查询数据
4. 更新数据
5. 删除数据

3. 查询数据

查询数据时，需要注意查询语句的编写，如使用 WHERE 子句等。

4. 更新数据

API 接口

API 接口是应用程序之间进行数据交互的桥梁。

API 接口的调用方式有多种，如 REST、SOAP 等。

调用 API 接口的示例如下：

```
PUT /crm/attribute/
```

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"
}
```

PUT

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"
  }
]
```

□□□□

□□□ DELETE /crm/attribute/attribute_id/{attribute_id}

□□□□□ delete_customer_attribute

□□□

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

□□□□□□□

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□

```
import requests

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

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

API

API

```
# API
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()

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

API

API

API

1. API

- external_system_id ✓
- billing_method ✓ method ✗
-
-

2. API

- 150
- "true"/"false"
- ISO 8601 "2025-01-04 10:30:00"
- JSON

3. []

- []
- []
- []

4. []

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

5. []

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

[][][]

[][]

[][][]

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

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

创建属性

更新属性

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

# 获取属性
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
)
```

获取属性

删除属性

```
# 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 endpoint `view_customer_attribute`
- UI component `view_customer_attribute`
- UI component `view_customer_attribute`

UI components

- UI component `update_customer_attribute`
- UI component `attribute_id`
- UI component `attribute_id`

UI components

- UI component `attribute_id`

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

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

Database

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

MySQL

- `username` - MySQL username
- `password` - MySQL password
- `server` - MySQL server IP address (localhost)

Database

- Database name: `omnicrm`
- Port: 3306 (MySQL default)
- Connection string: `mysql+pymysql://username:password@server/omnicrm`

Database connection string: `mysql+pymysql://username:password@server/omnicrm`

Database connection string: `.env` file

Service

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

Database connection string: `mysql+pymysql://username:password@server/omnicrm`

Database connection string: `.env` file

- Database

- `hss_peers`
- `apn_list`
- `hss`

hss_peers

HSS `hss_peers`

hss_peers HSS `hss_peers`

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

hss

- `hss_peers` - HSS `hss_peers`
- `apn_list` - APN ID `apn_list`

hss `hss_peers`

Mailjet `hss_peers`

OmniCRM `hss_peers` Mailjet `hss_peers`

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 - "
```

0000

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

000000000000/0000

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

000000

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

00000000

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

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

```
# 000000
```

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

```
# 000000
```

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

```
# 000000
```

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

```
# 000000
```

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

```
# eSIM 000000
```

```
# 0 template_id 000000 Mailjet 000000000000
```

```
# 000000 0 0000000 HTML 0000
```

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

eSIM 000000

0 eSIM 00000000000000

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

□□□□

- eSIM □□□□□□□□□□
- eSIM □□□□□□□□

□□□□□□□□ **Mailjet** □□□□□

- `{{ var:customer_name }}` - □□□□□
- `{{ var:iccid }}` - eSIM ICCID
- `{{ var:activation_code }}` - LPA □□□□□□□□
- `{{ var:has_qr_code }}` - □□□□□□□□ QR □□□□□□□□ true
- `{{ var:has_activation_code }}` - □□□□□□□□□□□□□□□□ true

□□□□□

□ `template_id` □□□ 0 □□OmniCRM □□□□ eSIM QR □□□□□□□□□□ HTML □□□□□□□□□□□□
□□□□ `branding` □□□□□□□□□□□□

□□□□□□□□□□□□□□□□

□ OmniCRM □□□□□ HTML □□□□□□□ Mailjet □□□□□□□□□□□□□□□□□ `branding` □□□□□□□□□□

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

□□□

項目	説明	値	値
brand_name	ブランド名	from_name	'Norfone'
primary_dark	プライマリダーク	#594712	'#1a73e8'
dark_color	ダークカラー	#32280B	'#0d47a1'
accent_bg	アクセント背景	#f5f0e0	'#e3f2fd'

注意事項

- eSIM 送信時に `api_crmCommunicationEsimDelivery.template_id` に 0 を指定

Mailjet 設定

- Mailjet アプリ (<<https://app.mailjet.com>>)
- Transactional → Templates
- テンプレートを作成
- Template ID を取得
- OmniCRM の設定
- `crm_config.yaml` に Template ID を設定

注意事項

テンプレート変数

- `{{customer_name}}` - 顧客名
- `{{service_name}}` - サービス名
- `{{invoice_id}}` - 請求ID
- `{{invoice_amount}}` - 請求金額
- `{{due_date}}` - 支払期日
- `{{reset_link}}` - リセット URL
- `{{verification_link}}` - 検証 URL
- `{{balance}}` - 残高
- `{{expiry_date}}` - 有効期限

□□□□

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

□□□

- `failure_list` - Ansible □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□

□□□□

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

□□□

□□□□ PDF □□□□□ Jinja2 HTML □□□

□□□□□ /OmniCRM-API/invoice_templates/

□□□□□□□□□□□□□□□□□□□□□□□□ □□ **PDF** □□ □□□

CRM □□ URL

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

□□□

- □ Ansible □□□□□□□□ API □□
- □□□□□□□□□□□□□□□□ CRM □□□
- □□□ API □□□□□□□□ URL

□□□

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

•••

- `ocsApi`
- `ocsTenant` URL
- `cgrates`

OCS `CGRates`

`ocs:`

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

•••

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

••• `source_msisdn`

SMSC `SMS`

`smc:`

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

•••

- `source_msisdn` SMS `2FA`
- `smc_url` - SMSC API

- `api_key` - SMSC API キー

環境変数

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

環境変数 CBC API キー

```
features_cell_broadcast
```

JWT キー

```
jwt_secret: 'CHANGE_ME_ON_DEPLOYMENT' # Ansible キー
```

環境変数

- JWT キー
- 環境変数 - キー
- 環境変数/環境変数
- 環境変数

環境変数 **Ansible** キー

Ansible キー JWT キー

1. Ansible キー 64 ビット 256 ビット
2. Ansible キー
3. 環境変数 - キー

環境変数

- Git キー
- 環境変数
- 環境変数
- 環境変数

Ansible 任务

Ansible 任务

```
# 生成随机令牌
python3 -c "import secrets; print(secrets.token_hex(32))"
# 生成随机令牌
openssl rand -hex 32
```

任务 `crm_config.yaml`

```
jwt_secret: 'your_generated_secret_here'
```

任务

- `CHANGE_ME_ON_DEPLOYMENT`
- 任务 - 生成随机令牌
- 任务
- 任务

Stripe 任务

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

任务

- `secret_key` - Stripe API 密钥
- `publishable_key` - Stripe 发布密钥
- `currency` - ISO 4217 货币代码 `aud` `usd` `gbp` `eur`
- `statement_descriptor_suffix` - 任务

任务

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

Example:

- `"YOURCOMPANY"`
- `22`
- `YOURCOMPANY`
- `PDF YOURCOMPANY_12345.pdf`

Example: `Payment Vendor Integrations`
`<integrations_payment_vendors>`

API IP

Example IP API

```
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 whitelist API
ip_whitelist:
  "10.179.2.142":
    roles: ["admin"]
```

Example:

- `API`
- `IP`
- `API`
- `API`

API

```
openssl rand -base64 48
```

□□□

- `admin` - □□□□□□□□□□
- □ RBAC □□□□□□□□□□

□□□□□□□

- □□□□□□□ API □□□□□ 32 □□◆◆□
- □ IP □□□□□□□□
- □□□□□□□□□
- □□□□□ API □□
- □□□□□□□ API □□□□□□□

□□□□□

- □□□□□□□□□□□□ IP □□□
- □□□□□□□□□□ IP□127.0.0.1□::1□
- □□□□□□□□□□□□ API □□
- □□□□□□□□□□□□□□

□□□□□□□□□□ `administration_api_keys` □ `concepts_api` □

□□□□□ (**.env**)

React UI □□ `OmniCRM-UI/.env` □□□□□□□□□□□□

□□□ `/OmniCRM/OmniCRM-UI/.env`

API

```
# Google Maps API
VITE_GOOGLE_API_KEY=your_google_api_key

# npm start
BROWSER=none
```



```
# stripe paypal
VITE_PAYMENT_VENDORS="stripe, paypal"

# Stripe Stripe
VITE_STRIPE_PUBLISHABLE_KEY=pk_test_xxxxx

# PayPal ID PayPal
VITE_PAYPAL_CLIENT_ID=YOUR_PAYPAL_CLIENT_ID
```


- `VITE_STRIPE_PUBLISHABLE_KEY` `crm_config.yaml` `stripe.publishable_key`
- `VITE_PAYPAL_CLIENT_ID` `crm_config.yaml` `paypal.client_id`

`VITE_*` UI


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

UI

- `COMPANY_NAME` - 公司名稱
- `PORTAL_NAME` - 門戶/系統名稱“ShellManager”
- `SELF_CARE_NAME` - 自助服務名稱“ShellCare”
- `COMPANY_TAGLINE` - 公司標語

環境變數

```
# 語言
# 支援語言 ar ch en fr gr it ru sp
VITE_DEFAULT_LANGUAGE=en
VITE_LOCALE="en-GB"

# 地點
VITE_DEFAULT_LOCATION="Sydney, Australia"
VITE_DEFAULT_COUNTRY="Australia"

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

語言

- `ar` - 阿拉伯語
- `ch` - 中文
- `en` - 英語
- `fr` - 法語
- `gr` - 希臘語
- `it` - 義大利語
- `ru` - 俄語
- `sp` - 西班牙語

貨幣

- USD - \$
- GBP - £
- EUR - €
- AUD - \$

- CAD - \$

crm_config.yaml stripe.currency

```
#
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_PRIMARY_COLOR -
- VITE_SECONDARY_COLOR -
- VITE_TERTIARY_COLOR -
- 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 -

#RRGGBB

-
-
-

- Font
- Font

Font

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

Font OmniCRM-UI Font

- Font - Font
- Font - Font
- Font - Font
- Font - Font
- Font - Font
- Font - Font

Font

Font

- Inter Roboto Open Sans Lato Quicksand Poppins Nunito

Montserrat Work Sans Source Sans Pro Raleway Ubuntu Josefin Sans

HKGrotesk

Font

- Merriweather Lora Playfair Display

Font

- System - Font

Font files

Font files

Font files

1. Create a directory `OmniCRM-UI/src/assets/fonts/your-font-name/`
 - o Create WOFF2 font files
 - o Create font files 300, 400, 500, 600, 700
 - o Create files `your-font-name-300.woff2`, `your-font-name-400.woff2`
2. Edit `OmniCRM-UI/src/assets/scss/fonts/_google-fonts.scss` and add `@font-face`

```
//  
// Font files - Font  
//  
  
@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. Add `.env` file

```
VITE_FONT_FAMILY=Your Font Name
```

HTTP Status Codes

- 300 - Multiple Choices
- 400 - Bad Request
- 500 - Internal Server Error
- 600 - Not Defined
- 700 - Not Defined

Web Applications

Configure 6 web applications in the configuration file

```
# Web App 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 App 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 App 3-6: Placeholder
# NAME URL ICON_PATH
```

Variables

- `VITE_WEB_APP_N_NAME` - Name
- `VITE_WEB_APP_N_URL` - URL
- `VITE_WEB_APP_N_ICON_PATH` - Icon path (relative to public/)

Supported web applications: GitHub, Xero, Monday.com, Gmail, MailJet, Slack

Grafana Integration

OmniCRM uses Grafana for monitoring. Access Grafana via `grafana_access` and `/grafana` endpoint.

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

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

•

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

• **ID**

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

•

OmniCRM “” OmniCRM

•

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

- ID `system-overview-123`
- “”

•

- ID
-
- `grafana_access`
- UI

URL

```
# URL
VITE_FAQS_URL=https://support.yourcompany.com/faq
VITE_SUPPORT_URL=https://support.yourcompany.com
```

UI

```
# yes | no
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. □□□□□□ 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. 部署

- 部署到生产环境
- 部署到生产环境
- 部署 Google Analytics 追踪

部署 - 部署

部署 - 部署

部署 - 部署

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

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

部署 - 部署

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

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

部署 - 部署

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




// 初始化 Google Analytics
trackPageView('/virtual/page-path', 'Page Title');
```

API 參數

函數名稱	參數	說明
<code>initializeGoogleTracking()</code>	無參數	初始化 Google Analytics
<code>trackEvent(eventName, eventParams)</code>	事件名稱 事件參數	追蹤事件
<code>trackPageView(path, title)</code>	頁面路徑 頁面標題	追蹤頁面瀏覽

注意事項

Google Analytics 使用條款

- **GDPR** 符合 - 符合歐盟通用數據保護條例
- **CCPA** 符合 - 符合加州消費者隱私法案
- 隱私 - 我們使用 Google Analytics
- **Cookie** 使用 - 我們使用 cookie    通知
- 隱私 - 我們使用 Google Analytics 追蹤用戶行為

更多資訊

項目	環境変数	説明
GA の ID	GA ID	GA ID の形式は G-XXXXXXXXXX と GTM-XXXXXXXX
環境変数の名前	環境変数 ID	環境変数 ID を .env に設定
ビルドコマンド	UI の .env 設定	npm run build コマンドで UI をビルド
環境変数	環境変数	環境変数の設定
環境変数	GA の ID	環境変数の設定

環境変数

- 環境変数の名前
- 環境変数の値 < 50KB を超えない
- 環境変数の形式
- 環境変数の名前 `VITE_GOOGLE_TAG_ID` を設定

環境変数

- 環境変数の名前 `OmniCRM-UI/src/utils/googleAnalytics.js`
- 環境変数の名前 `OmniCRM-UI/src/utils/usePageTracking.js`
- 環境変数の名前 `App.js` と `index.js` を設定

環境変数

VITE_TOPUP_PRICE_PER_DAY=10

環境変数の設定

環境変数

- 環境変数の名前 `VITE_TOPUP_PRICE_PER_DAY` - 環境変数の設定

이 문서는 10 페이지 중 10 페이지입니다

이 문서는 features_topup_recharge입니다

OmniCRM

OmniCRM은 고객 관계 관리 시스템을 제공하는 오픈 소스 소프트웨어입니다.

설치

이 문서는 /OmniCRM-UI/src/assets/images/omnitouch/ 폴더에 포함되어 있습니다.

이 문서는

- DefaultLogoDark.png - 다크 테마 로고
- DefaultLogoLight.png - 라이트 테마 로고

이 문서는

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

이 문서는

이 문서는

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

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

□□□□□□□□

- **logoSm.png** - □□□□□□□□□□□□□□□□
- **logoDark.png** - □□□□□□□□□□□□□□□□
- **logoLight.png** - □□/□□□□□□□□□□□□□□□□

□□□□□

1. □□□□□□□□□□
 - □□ PNG □□□□□□□□□□
 - □□□□□□□□□□□□□□□□□□□□
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. 이미지

- 이미지명 지정 `logoDark.png`
- 이미지명 지정 `logoLight.png`
- 이미지명 지정 `logoSm.png`
- 이미지명 지정 `logoLight.png`

이제

- 이미지 - 이미지명 지정
- 이미지 - 이미지명 지정
- 이미지 - 이미지명 지정 PNG
- 이미지 - 이미지 DPI 이미지 2 이미지
- 이미지 - 이미지명 지정

이제

이제

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

이제

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

// 이미지명 지정
<img src={logoLight} alt="" height="18" />
```

이제

1. 이미지 `/OmniCRM-UI/src/assets/images/` 이미지 `logo-light.png`
2. 이미지명 지정 CSS 이미지 `.auth-one-bg`
3. 이미지 `authCarousel.js` 이미지명 지정

□□□□□

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

□□ PDF □□

OmniCRM □□ Jinja2 HTML □□□ WeasyPrint PDF □□□□□ PDF □□□

PDF □□□□□

1. □□□□

- □□□□□ `crm_config.yaml` □□ `invoice.template_filename` □□□
- □ `/OmniCRM-API/invoice_templates/` □□□□□

2. □□□□

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

3. □□□□

- Jinja2 □□□□□□□□ HTML □□
- □ `{{ invoice_number }}`□`{{ total_amount }}` □□□□□□
- □□□ HTML □□□ `invoice_templates/rendered/` □□□□

4. PDF □□□

- WeasyPrint □□□□ HTML □□□ PDF
- PDF □□ CSS □□□□□□□□□□□□□□□□

- PDF 生成

5. 保存

- PDF を Base64 エンコードして `Invoice_PDF_Cache` に保存
- SHA256 ハッシュを生成
- ハッシュを PDF 名に追加

6. 送信

- 送信先 URL を取得
- 送信先 URL に送信

送信先 URL

送信先 URL は Jinja2 HTML テンプレート

送信先 URL は `/OmniCRM-`

`API/invoice_templates/yourcompany_invoice_template.html`

送信先 URL

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

□□□□□□□□

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

```

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

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

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

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

{% if paid %}

```


3. 設定

crm_config.yaml

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

4. 出力

- CRM 出力
- PDF 出力
- invoice_templates/rendered/{invoice_id}.html

5. 出力

出力

```
-- 出力 PDF  
DELETE FROM Invoice_PDF_Cache;
```

PDF 出力

OmniCRM 出力 PDF

出力

- 出力 - PDF
- 出力 - PDF
- 出力 - PDF
- 出力 - 30

出力

- 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 □□□□
- □□□□ □□□□□□□□□□ API/UI □□

YAML □□□□

- □□□□ □□□□ YAML □□□□□□□□□□□□
- □□□□ □□□□□□□□□□ `yamllint crm_config.yaml`

Stripe

- Stripe API key
- Stripe account ID

Stripe API

- Stripe API endpoint
- Stripe API `publishable_key` parameter

Mailjet

- Mailjet API key ID
- Mailjet API secret key ID

PDF

- WeasyPrint installation: `pip install weasyprint`
- Configuration file: `crm_config.yaml`
- Invoice templates: `invoice_templates/rendered/` (HTML files)
- API endpoint: Jinja2

PDF rendering

- Image path: `file:///full/path/to/image.png`
- Image format: PNG or JPEG
- Image alt text

HTML

- CSS classes
- CSS layout: flexbox, grid
- HTML tags
- `div` elements

Cache

- Cache invalidation: `invalidate_invoice_cache()`

- `Invoice_PDF_Cache` `Cache`
-

Configuration

OmniCRM Configuration

Environment

- `.env.example` to `.env`
- `CGRates`
- `crm_config.yaml` configuration
 - `CGRates`
 - `Mailjet API Key ID`
 - `Stripe API Key`
 - `CRM API URL`
 - `OCS/CGRates`
 - `SMSC`
 - `JWT Secret`
 - `Stripe API Key`
 - `API IP`

Deployment

- `OmniCRM-UI/.env.example` to `OmniCRM-UI/.env`
- `Google Maps API Key`
- `Stripe API Key`
- `CGRates`
 - `CGRates`
 - `CGRates`

- 0000000
- 0000
- 0000000
 - 0000
 - 00
 - 0000000000
 - 00000000
- 000000000
- 00 Web 000000000
- 0000000000 URL0000
- 00 Google Analytics/Tag Manager 00 ID0000

0000

- 00000000logoSm.png|logoDark.png|logoLight.png
- 0000000 `OmniCRM-UI/src/assets/images/omnitouch/`
- 0000000000 HTML
- 000000000 `OmniCRM-API/invoice_templates/`
- 00 `crm_config.yaml` 00000000000
- 0000 PDF 00
- 00 UI `npm run build`

000

- 000000000
- 000000 JWT 00
- 0000 Stripe 000000000000
- 00 Mailjet API 00
- 00000000
- 00 API 000 IP 000
- 00 SSL/TLS 00
- 00000000 HTTPS
- 00 CORS 00
- 0000000

0000000000

00

- 00000000
- 0000000000
- 0000000000000000
- 00000000 PDF 00
- 00000000Stripe
- 0000000000 2FA
- 0000000000
- 00 OCS 00000000
- 00 ActionPlan 000000
- 000000000000

00

- 00000000MySQL
- 00 CGRates
- 00 OmniCRM API
- 00 OmniCRM UI
- 00000000nginxtraefik
- 000000GrafanaPrometheus
- 00000000
- 00000000
- 0000000000

0000

- RBAC 000000 </rbac>
- 000000 </concepts_products_and_services>
- Ansible 00 </concepts_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

- 查詢裝置資訊

查詢裝置資訊

查詢裝置

- 查詢**1**查詢裝置資訊
▪ 查詢“ICCID”SIM“MAC”查詢裝置
- 查詢**2**查詢裝置資訊
▪ 查詢“IMSI”SIM“查詢”查詢裝置
- 查詢**3-20**查詢裝置資訊
▪ 查詢 查詢 查詢
- 查詢 查詢 查詢

查詢裝置

- 查詢裝置資訊
- 查詢裝置資訊/查詢裝置資訊

4. 查詢 查詢

查詢裝置

查詢裝置

1. 查詢 查詢 → 查詢
2. 查詢裝置
3. 查詢 查詢
4. 查詢裝置
5. 查詢 查詢

查詢

查詢裝置 `itemtext1_label` 查詢裝置

查詢


查詢 `inventory_items_list` 查詢裝置

□□□□□□

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□□□□□□□□□□

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
 - On - Available
 - Off - Unavailable
 - Pending - Pending
 - Reserved - Reserved
 - Damaged - Damaged
 - Expired - Expired
 - Deleted - Deleted
 - Archived - Archived

Inventory Item

Inventory Item 20

- **itemtext1** - Item Text 1 `itemtext1_label`
 - MAC Address
 - SIM ICCID
 - IMEI
- **itemtext2** `itemtext20` - Item Text 2
 - Item Text 2 Label
 - SIM 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
- → -
- → -
- / → -

-
-
-
-
-

-  000
- 00000000000000000000
- 00000000000000000000
- 0000000000

000000

0000000000

1. 0000000000
2. 00 00 000
3. 00 0000 00
4. 00000000
 - 00000000000000000000
 - 00000000000000000000000000000000
 - 00000000000000000000 URL
 - 00000000000000000000
 - 00000000000000000000000000 2099-01-01
5. 000000000000
6. 00 0000

□□□□

□□□□□□□□

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}
```

□□□□□

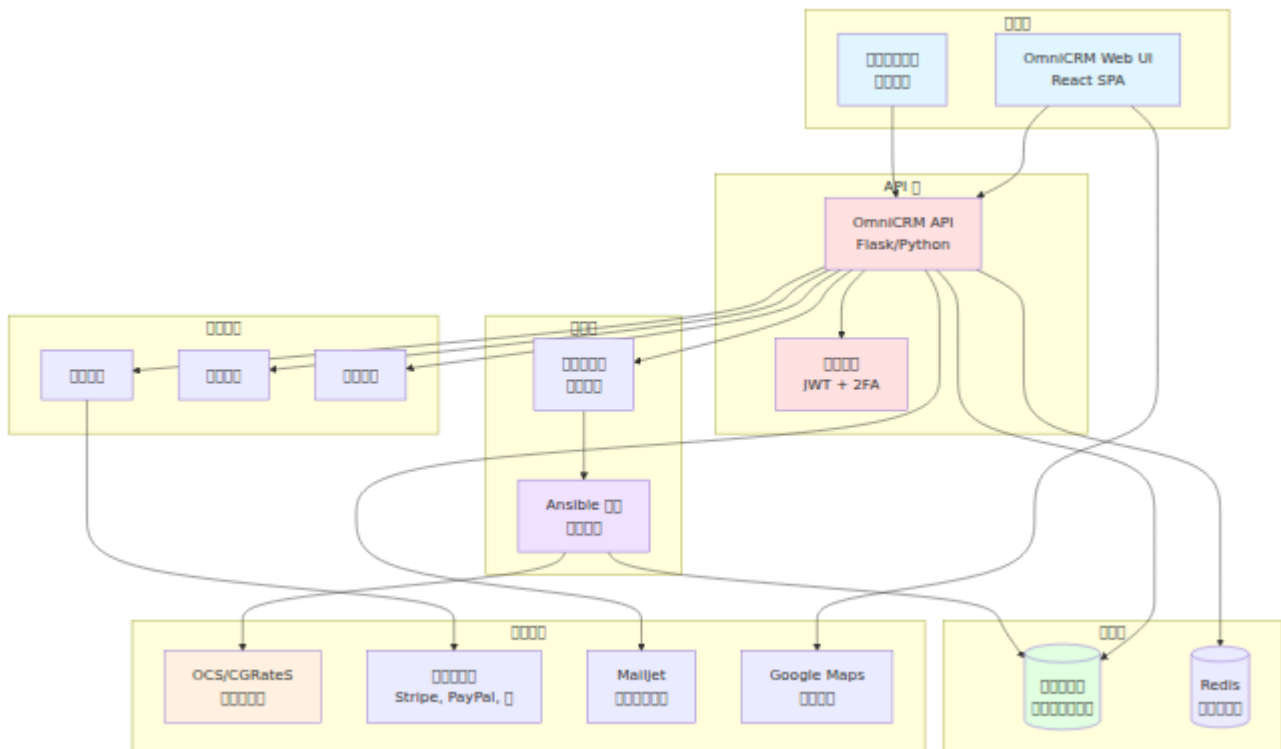
```
DELETE /crm/tag/tag_id/{tag_id}
```

OmniCRM □□□□

□□□□□□ OmniCRM □□□□□□□□□□□□□□□□□□

□□□□□□

OmniCRM □□□□□□ BSS/OSS □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□



□□□□

1. □□□□

OmniCRM Web UI

- React □□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□
- □□□□□□□□□□

□□□□□□

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

2. API □

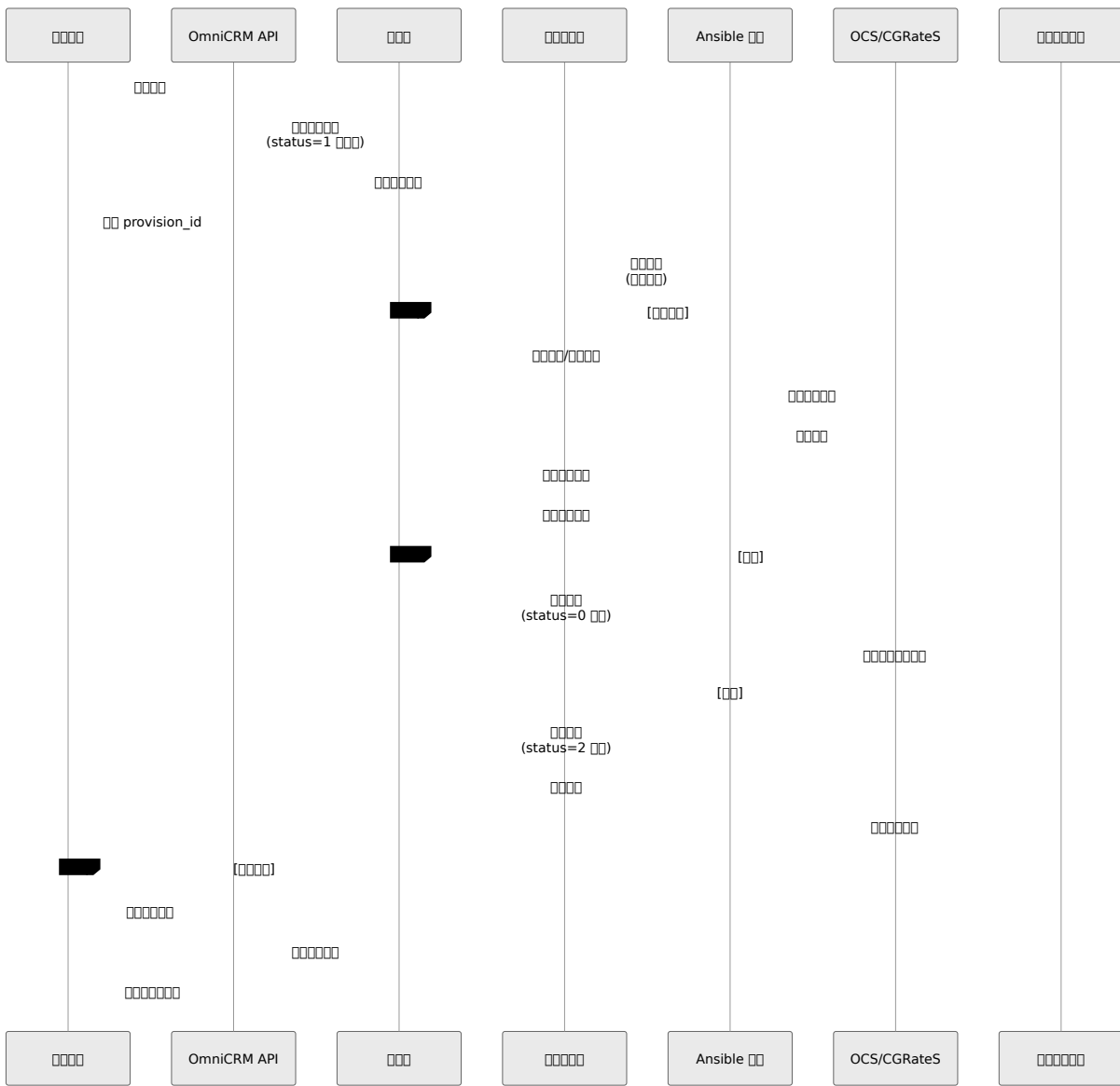
OmniCRM API (Flask/Python)

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

□□ **API** □□:






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

3.

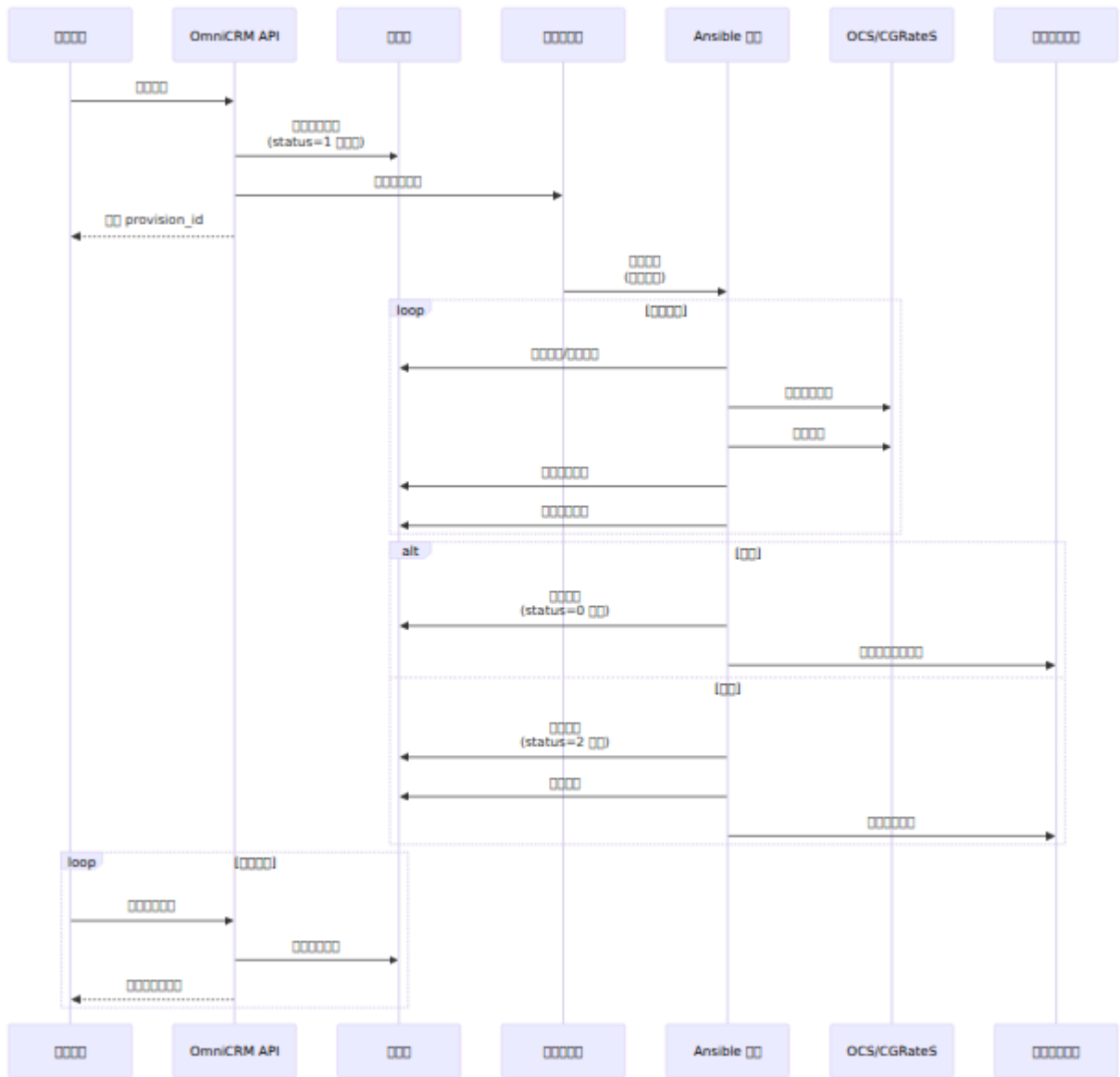


4.

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' 或 '法人'
customer_status	文字列	'有効', '無効', '凍結', '削除'
customer_payment_type	文字列	'現金' 或 '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 - 制約

欄名	制約	データ型
contact_id	PK	整数
customer_id	FK	整数
contact_firstname		文字
contact_lastname		文字
contact_email		文字
contact_phone		文字
contact_types		'A', 'B', 'C'

住所 - 住所情報

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

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	ホストリスト
retail_cost	整数	小売価格
retail_setup_cost	整数	小売設置コスト
enabled	ブール値	有効/無効

備考

00 - 0000/00

欄名	制約	データ型
transaction_id	PK	整数
customer_id	FK	整数
invoice_id	FK	整数
service_id	FK	整数
title		文字列
retail_cost		小数
tax_percentage		小数
tax_amount		小数
void		整数

PK - 主キー

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	varchar	品名 (SIM 含む)
item_state	enum	'0', '1', '2' のいずれか
itemtext1-20	varchar	品名補足
management_url	varchar	管理URL
config_content	text	設定内容

在庫 - 在庫テンプレート

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

在庫

品名 - 品名補足

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 - ユーザー

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 - ユーザー

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

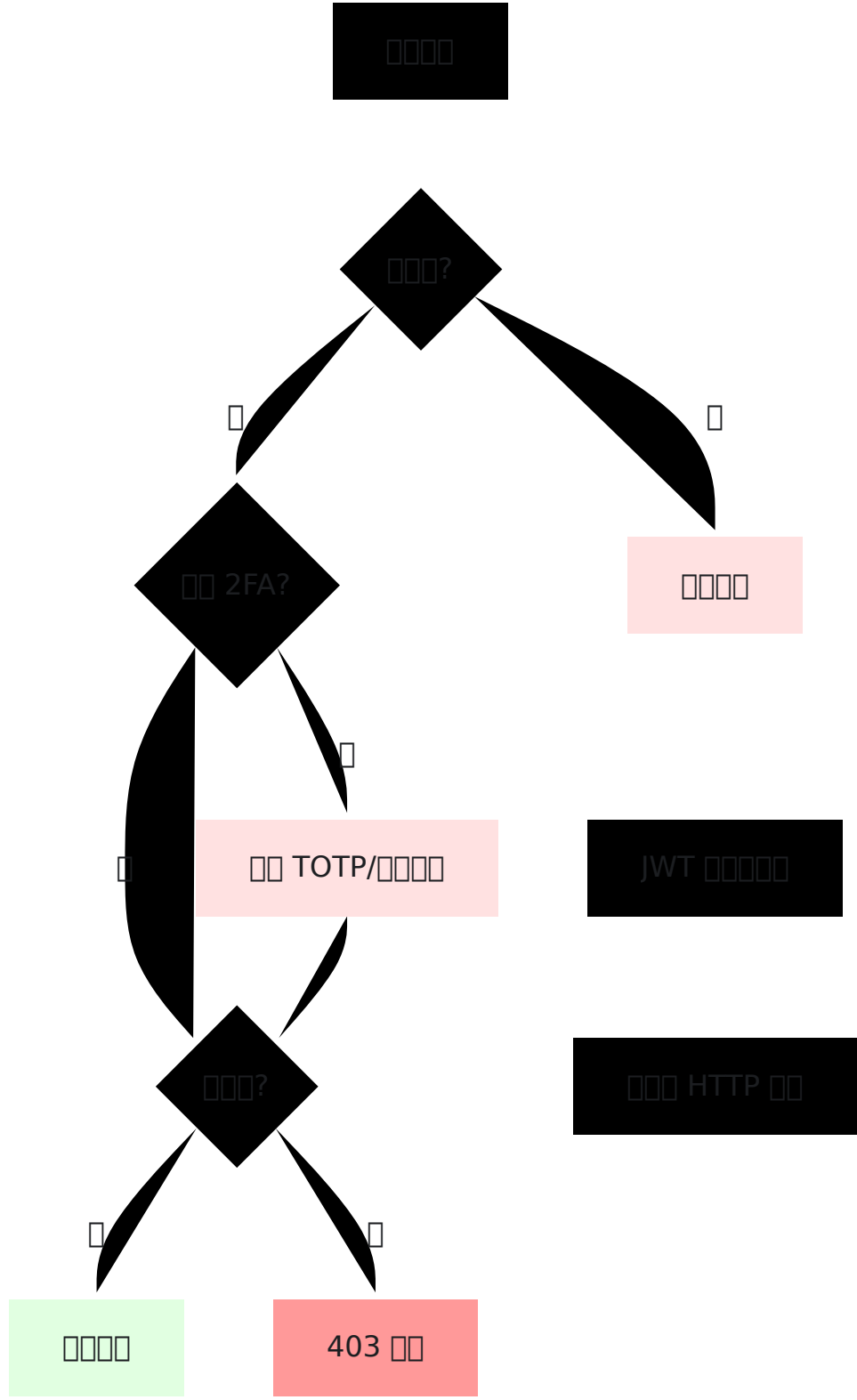
-
-
- Plus Code
-

OCS/CGRateS

-
-
-
- CDR

- □□□□□□□□

□□□□

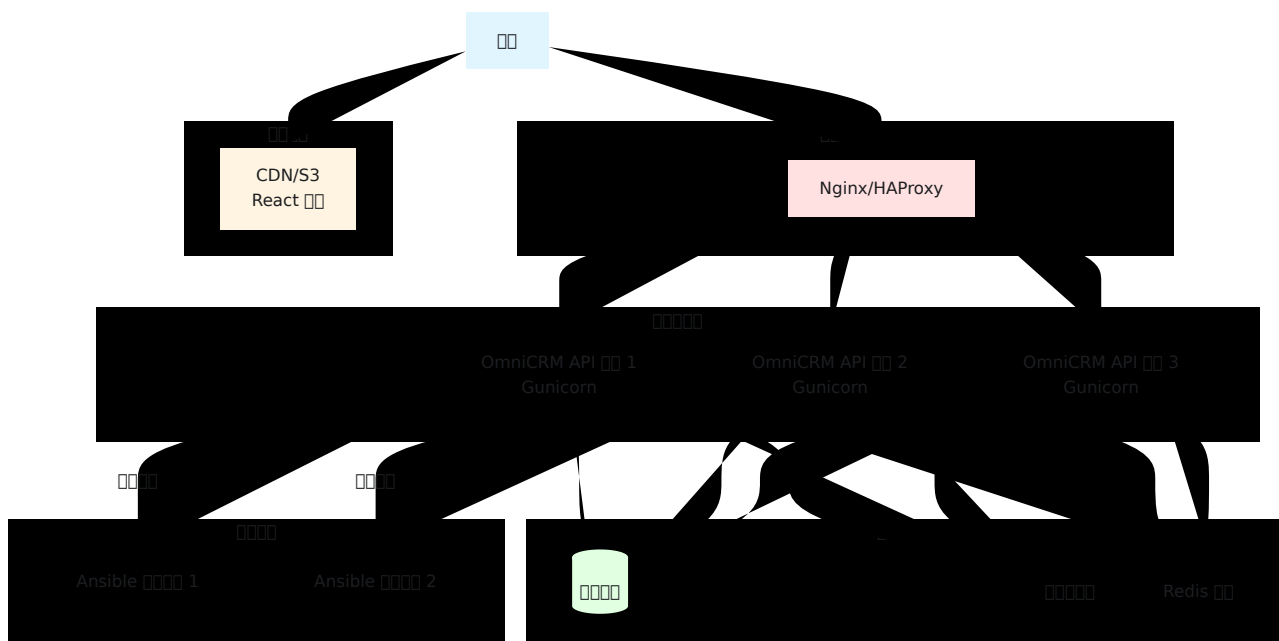


□□□□:

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

□□□□

□□□□□□:



□□□

□□:

- 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.

Endpoints:

- API
- Frontend/Backend
- Database
- OCS, Stripe, Mailjet
- External services

Tools:

- Flask
- Ansible
- Docker
- Kubernetes

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

Stack

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

Table of Contents

OmniCRM 2FA 2FA UI

Self-Care Portal <self_care_portal> RBAC <rbac>

Table of Contents

OmniCRM

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

Table of Contents

□□□□

□□□□□

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 2FA 如何 / 如何 2FA 如何

如何

1. 如何 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

如何

如何

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

如何

如何 “如何” 如何

1. 安裝認證應用程式
 - iOS → Apple App Store → “Google Authenticator”
 - Android → Google Play → “Google Authenticator”
 - 其他：Authy、Microsoft Authenticator、1Password
2. 掃描QR碼
 - 掃描QR碼
 - 點選 “+” 或 “掃描”
 - 點選 “QR碼”
 - 掃描QR碼
 - 掃描QR碼 “OmniCRM” 掃描器 6 個月
3. 設定認證
 - 設定 密碼 8 位數
 - 設定 時間同步
 - 設定 通知
 - 設定 自動登入
 - 點選 “掃描” 掃描QR碼
4. 測試認證
 - 設定 密碼 6 位數
 - 點選 “測試”
 - 設定 2FA 測試

如何設定 2FA 測試

測試方法

如何設定 2FA 測試

□□□□□□□□□□

□□□□□□□□□□□□□□□□

□□□□□□□□

□□□□□□ “□□□□”

1. 00 0000000
2. 00 “0000000”

0000000

- 0000000 0000000000000000
- 00000000 Mailjet 000000000000
- 000000000000000000000000 1 000000
- 00000000 “000000000000000000000000”

000000000000

000000 “000000000000000000000000”

0000000000000000

0000000000000000

0 [00]0

OmniCRM

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

1

[]

- 1.
- 2.
3. “ ”

- 8
-

- “ ”
-
-

/

XXXXXXXXXXXXXXXX

[XXXXXXXXXXXX]

XXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

Hi John Smith <john@example.com> +44 123... ☎ Jane Doe <jane@example.com> +44 456... ☎ Bob Wilson <bob@example.com> +44 789... ☎

XXXXXX

- ☎ XXX - XXXXXXXXXXXXXXXXXXXX
- ☎ XXX - XXXXXXX
- ☎ XXXXX - XXXXXXX
- ☎ XXX **2FA** - XXXXXX 2FAXXXXXX 2FA XXXXX
- ☎ XXXXXXX - XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXX (X)

XXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

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

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[] []

“ ”

John Smith

[]

△ • • •

[]

-
-
-
-

- 1.
2. “ ”
- 3.
- 4.

2FA

2FA ()

2FA

- 0000000000
2. 0000000000
 3. 0000000000000000 2FA

0000000000000000

0000000000000000000000

0000

00000 (✉) 00 0000 00000 (login_count = 0)

00000000000000 (✉)

0000000000000000

000 Bob Wilson (<bob@example.com>)

00000000 • 0000 • 0000 • 0000000000000000 • 00000000

[00] [00000000]

00 “00000000”

000000

00 **Mailjet** 0000000000

000000 api_crmCommunicationUserWelcome

0000000000

00 0000 (⇒) 0000000000

00

00

0000

□□□□

□□ 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 認證
- 認證
- 認證
- 認證 2FA
- 認證 2FA

認證

- 認證
- 認證
- 認證
- 認證
- 認證

認證

“認證”

- 認證
- 認證
 - 認證
 - 認證
 - 認證
 - 認證

2FA 認證

- 認證
- 認證
 - 認證 → 認證 → 認證
 - 認證 30 分鐘
 - 認證
 - 認證
 - 認證 2FA

“”

- Cookies
- - Cookies
 - Cookies
 - OmniCRM
 -

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

- 1
- -
 -
 -

2FA

-
- -
 -
 -

##

- 2FA 的用途/限制
- 2FA 的類型
 - 基於時間的 2FA
 - 基於位置的 2FA
 - 基於設備的 2FA
 - 基於 SMS 的 2FA

2FA 的實現

- 2FA 的實現
- 2FA 的實現
 - 基於時間的 2FA
 - 基於 API 的 2FA
 - 基於設備的 2FA

2FA 的實現

- 2FA 的實現
- 2FA 的實現
 - 基於 API 的 2FA
 - 基於設備的 2FA
 - 基於 SMS 的 2FA
 - 基於 API 的 2FA
 - 基於設備的 2FA

2FA 的實現

2FA 的實現

- 2FA 的實現
- “2FA”的實現 30 秒
- 2FA 的實現 HTTP-only cookies JavaScript 的實現
- 2FA 的實現 cookies 的實現 HTTPS 的實現

2FA 的實現

- 2FA 的實現

- 2FA 2FA API
- rbac - 2FA API
- administration_configuration - Mailjet
- integrations_mailjet - Mailjet
- customer_care - Mailjet

2FA 2FA

- TOTP 2FA API
- 2FA API
- 2FA API
- 2FA API

2FA 2FA

- 2FA 2FA API
- 2FA API
- 2FA API
- 2FA API IP

2FA 2FA

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



0fT52ZvoZBE

XXXXXXXXXXXX

XXXXXXXXXX XX XXXXXXXXXXXXXXXXXXXX XX XX XXXXX XX

XX XX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/XXXX

XX XX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XX XX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XX XX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```
XXXXXXXX <csa_activity_log>XXXXXXXXXXXX <administration_tags>XX  
<administration_attributes> XXXXXXXXXXXXXXXXXX  
<administration_inventory> XXXXXXXXXXXX <payments_transaction>XX  
<payments_invoices> X <basics_payment>X
```

XXXXXXXXXXXXXXXXXXXX <csa_add_service>XXXXXXXXXXXXXXXXXXXX

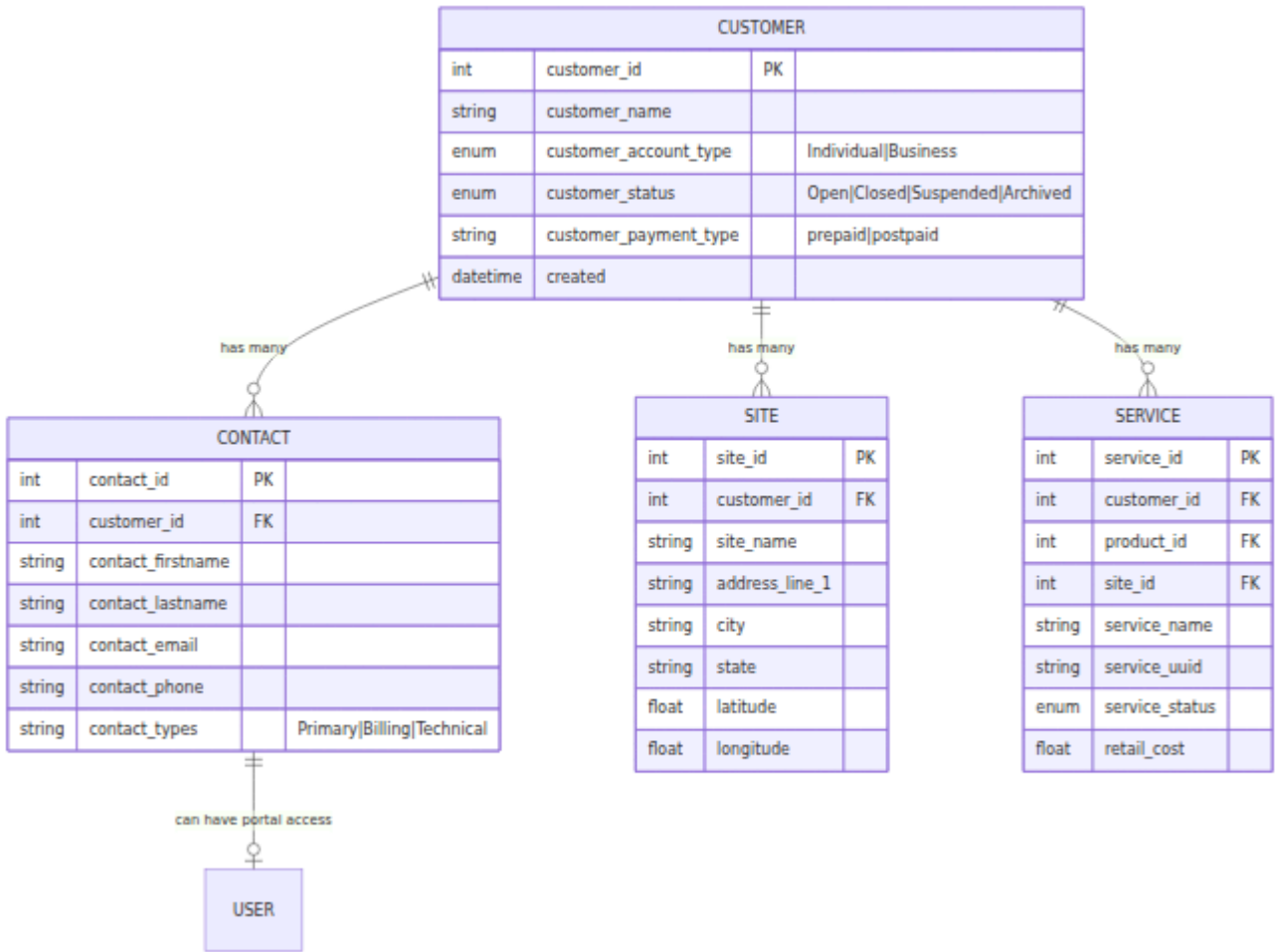
XXXXXXXXXXXXXXXXXXXX XXXXXX

XXXXXX

OmniCRM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

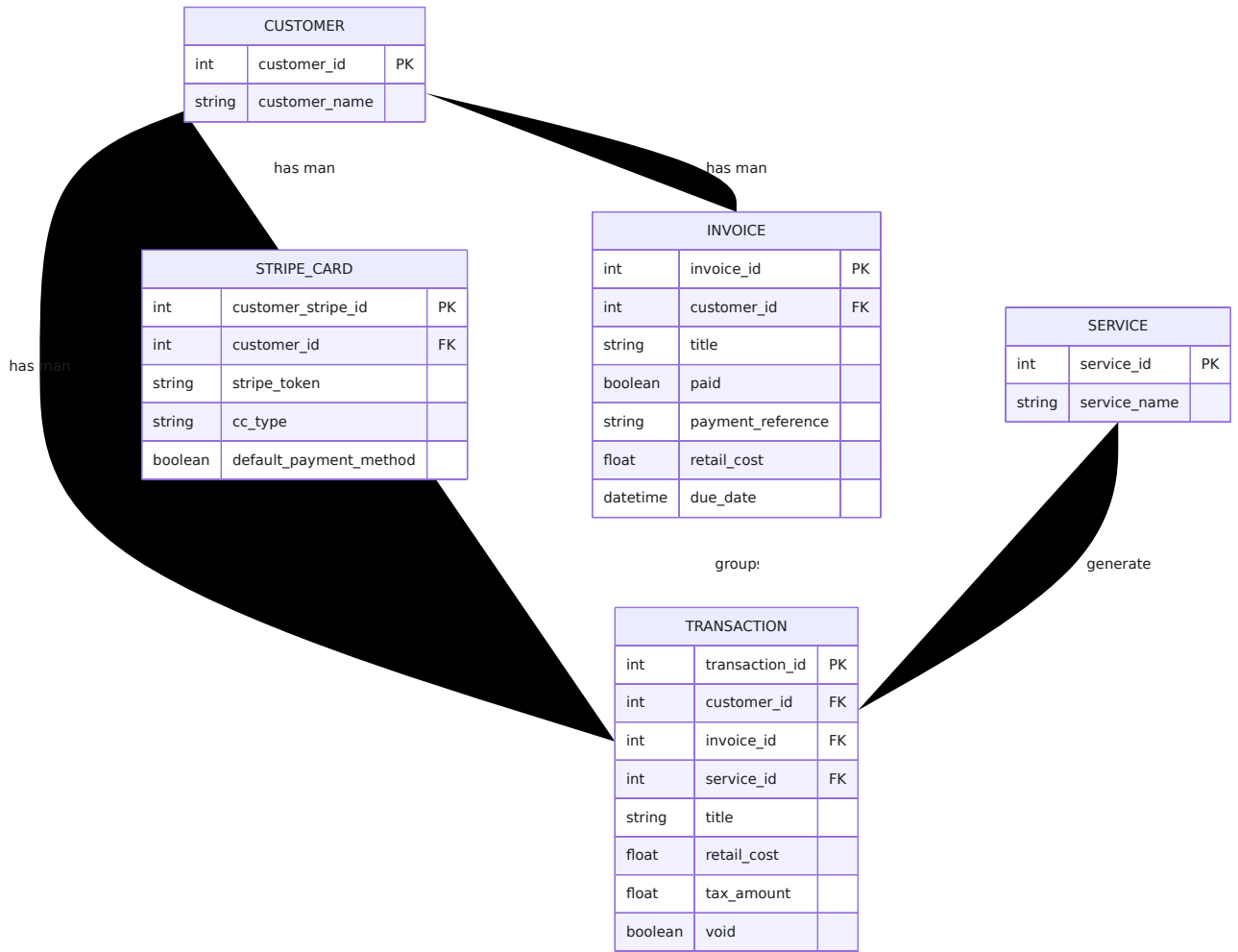


□□□□

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

□□□□□□□□

□□□□□□□□□□□□□□□□

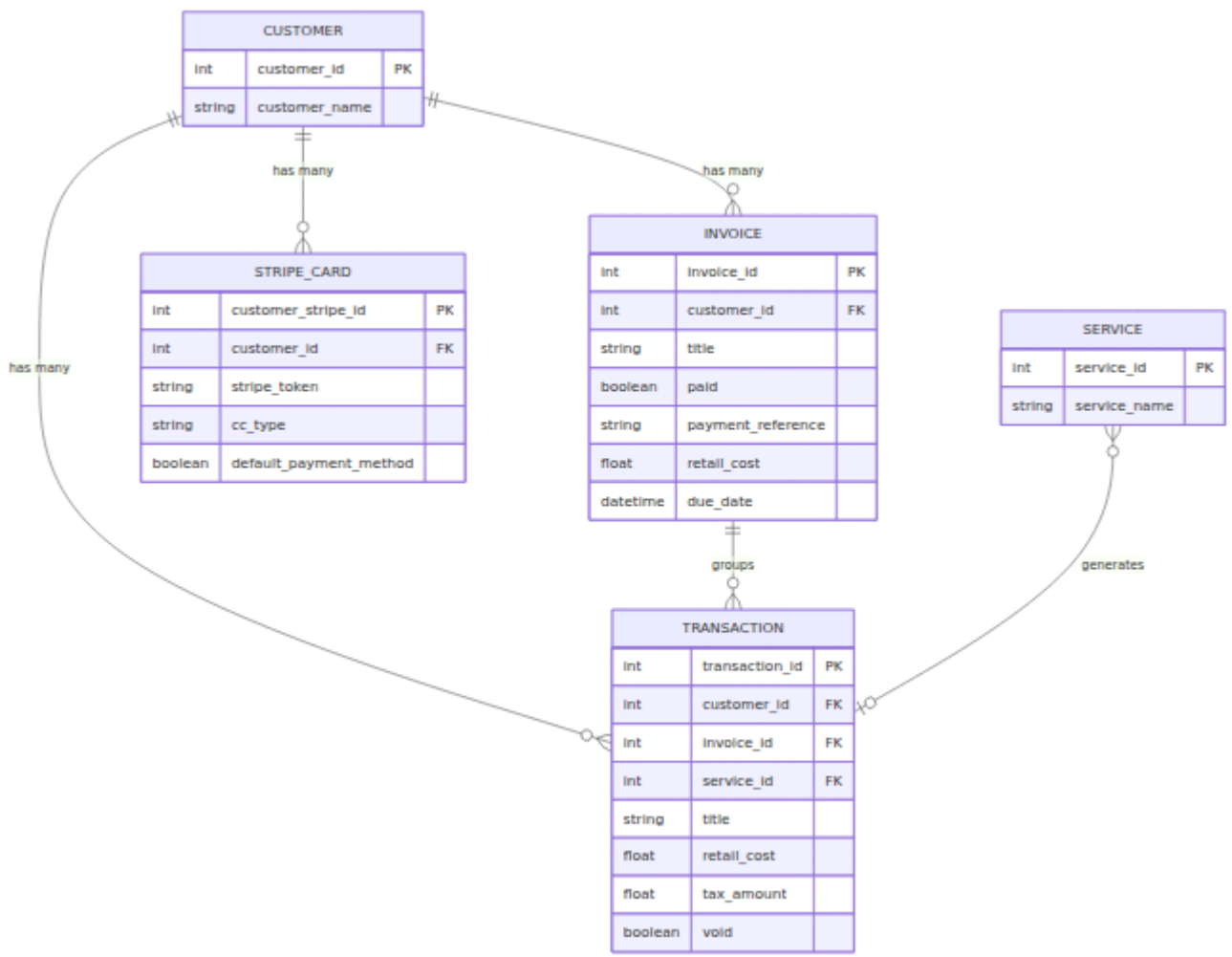


□□□□

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

2.

-
-
-
-

11

✓ "123 " ✓ 1 "123 " ✓ " " ✓ " " ✓ " " ✓ "SW1A 1AA" ✓ 51.5074 ✓ -0.1278 ✓ "9C3XGPHC+3Q" ✓ Google ID "ChIjdd4hrwug2EcRmSrV3Vo6III"

-
- VITE_DEFAULT_COUNTRY
- 12 11 + 1

□

□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□ OmniCRM □□□□

1. □□□□□□

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

2. □□□□

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

□□□□

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

□□□□47 □□□□□□□□ □□□□ 47 □□□□□□□□□□□□

3. □□□□□□

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

4. □□□□□□□□

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

XXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

1. XXXXXXXX

- o XXXXXXXX
- o XXXXXXXXXXXXXXXX/XX
- o XXXXXXXXXXXXXXXX

2. XXXXXXXX

- o XXXXXXXXXXXXXXXX **8C3MFJV8+2F** X
- o XXXXX Google Maps XXXXXXXXXXXXXXXX
- o XXXXXXXXXXXXXXXX

3. XXXXX

- o XX“XXXX”XXXXXXXXXX
- o XX“XXXXXXXXXXXX 500 X”
- o XXXXXXXXXXXXX

XX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX



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 - 00000000000000

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

- 零售店
- 零售店“店”
- “零售店”

零售店 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 $retail_cost \times (tax_percentage / 100)$
 - 零售店 - 零售店 \$10.00 (10%)
- 零售店/零售店 - 零售店
 - 零售店
 - 零售店 0%
 - 零售店

零售店

- 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"

0000 payments_invoices 0000000000000000

00000000

00000**1**00000000000000

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` - □□□□□□□□□□

□□□□

□□ </payments_transaction> □□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

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

□□□□□□□ □ Mailjet <integrations_mailjet> □□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□PDF□

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 Name, City, State, ZIP</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 Info</h3>
  <p><strong>Payment URL</strong> | https://yourcompany.com/pay |</p>
</div>
<p><strong>Payment Details</strong></p>
<ul>
  <li>Account Number: 123456789012</li>
  <li>BSB: 123-456</li>
  <li>Account ID: 987654321</li>
  <li>Invoice Number: {{ invoice_number }}</li>
</ul>
</div>
```

5. Terms and Conditions

```
<div class="terms">
  <h4>Terms and Conditions</h4>
  <p>These terms apply to all services provided by yourcompany.com</p>
  <p>All payments are due within 30 days of invoice date</p>
  <p>Late payments incur a 2% penalty</p>
  <p>For more information, contact billing@yourcompany.com</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
- 文字列

3. PDF

- 文字列 <5MB
- 文字列
- 文字列
- 文字列

4. 文字列

- 文字列
- 文字列
- 文字列

5. 文字列

- PDF
- Mailjet 15MB
- Gmail Outlook Apple Mail

文字列

API

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

文字列

文字列

- 文字列
- 文字列

PDF

- 文字列 CSS
- 文字列 CSS 2.1 WeasyPrint CSS

URL

- URL
- HTTPS URL

Table Layout

-
- `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

- 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 Configuration

CRM

1. CRM → [CRM]
2. CRM
3. CRM

CRM

1. CRM → CRM
2. CRM

CRM

CRM Configuration

CRM

- CRM - CRM

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

□□□□

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

□□□□

□□ "+ □□□□" □□□□

□□□□

- 𠄎𠄎 - 𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎
- 𠄎 - 𠄎𠄎𠄎/𠄎𠄎𠄎𠄎𠄎𠄎“𠄎 [𠄎]”
- 𠄎𠄎𠄎 - 𠄎𠄎𠄎𠄎𠄎𠄎𠄎14𠄎𠄎
- 𠄎𠄎𠄎 - 𠄎𠄎𠄎𠄎𠄎𠄎𠄎
- 𠄎𠄎 - 𠄎𠄎𠄎𠄎𠄎𠄎𠄎
- 𠄎𠄎𠄎 - 𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎/𠄎𠄎𠄎

𠄎𠄎𠄎

- ✓ (𠄎𠄎) - 𠄎𠄎𠄎𠄎𠄎
- × (𠄎X) - 𠄎𠄎𠄎𠄎𠄎
- 𠄎 - 𠄎𠄎𠄎𠄎
- 𠄎𠄎 - 𠄎𠄎
- 𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎
- 𠄎𠄎𠄎𠄎/𠄎𠄎𠄎𠄎

𠄎𠄎𠄎

1. 𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎
2. 𠄎𠄎𠄎𠄎𠄎/𠄎𠄎
3. 𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎
4. 𠄎𠄎𠄎𠄎𠄎𠄎
5. 𠄎𠄎PDF
6. 𠄎𠄎𠄎𠄎𠄎 `invoice_id` 𠄎𠄎
7. 𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎
8. 𠄎𠄎𠄎𠄎“𠄎”

𠄎𠄎

𠄎𠄎 𠄎𠄎 𠄎𠄎 𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎

𠄎𠄎𠄎𠄎 𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎

𠄎𠄎𠄎𠄎 𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎𠄎

- 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. 📌

□□□□□□□□

:doc:`□□□□□□ <self_care_portal>` □□□□□□

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

□□□□□□□□

- □□□□□□□□
- □□□□□□□□□□□□□□□□POS□□□□□□
- □□□□□□□□□□

□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□

□□ □□□□□□□□ □□□□□□□□□□

□□□□□□□

1. □□□□□□□□□□□□

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

API

- API를 호출하는 방법
- API를 호출하는 데 필요한 데이터

API 호출

1. API 호출하기
2. API 호출하기
3. API 호출하기

API 호출

- API 호출 시 `void = true`
- API 호출 시 `invoice_id` 값이 null이 아닌지 확인
- API 호출 시 "API"
- API 호출 시
- API 호출 시 "API:"
- API 호출 시
- API 호출 시 "API"를 호출

API 호출

- API 호출
- API = "API"를 호출

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

Stripe

Stripe API

Stripe

- Stripe API
- Stripe **Stripe** API
- Stripe API `payment_reference` Stripe ID
- Stripe API

Stripe

1. Stripe API
2. Stripe API
3. Stripe API

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

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

- 台灣 - 台灣
- 台灣 - 台灣

二

台灣

- **ID** - ID
- -
- -
- -
- -
- -
- -

台灣 ↔

二

10 25 50 100

台灣

台灣 **1**

1. 台灣 131
2. 台灣 →
3. 台灣 "+ 台灣"
4. 台灣
5. 台灣
 - 台灣 2025-01-01
 - 台灣 2025-01-31
 - 台灣 2025-02-15 15
 - 台灣 "2025 1" 台灣
6. 台灣 台灣 1

7. 00000
 - o 00000000
 - o 0000000000000000
 - o 0000000000
8. 00 "0000"000000000000"00000150"
9. 0000000000000000
10. 0000000000000000
11. 0000000000 "000000" 00000000000000
12. 00000PDF0000000000000000

0000200000000000

1. 00000000000+00000000000
2. 0000000000000000
3. 0000000000000000
 - o 00 "+ 0000000"
 - o 000"00000 - 2025010"
 - o 00/00010100310
 - o 0000020150
 - 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
 - 000000000000000000
 - 00 X 0000000000
 - 000000000000000000
 - o 00 "0000"

5. 00000000000000000000
 - o 0000 "+ 0000000"
 - o 0000"00000 - 2025010"
 - o 00000000000000000000
 - o 00 "00" 000000
 - o 00 "00000"
6. 0000000000000000000000

0000**3**0000000000000000

1. 0000000
2. 000000 00 00
3. 00 "+ 0000000"
4. 0000000000
5. 000000020000
6. 0000000000000000000000
7. 00000000
 - o 00000000000"0000000"0
 - o 00 **X** 0000000
 - o 000000000000"00000"0
 - o 00 **X** 0000000
 - o 00000000"0000180000"
 - o 0000000000
8. 00000000000000000000
9. 00 "000000**18**0"
10. 0000000000000000
11. 00/000000000000000000000000

0000**4**00000000000000

1. 0000000000
2. 0000000000000000
3. 000000000000
 - o 00000000000000

- 00 X 000000
- 0000000000000000
- 00 X 000000

4. 000000
5. 0000000000000000
6. 00 "0000"0000000000
7. 000000000000/00000000
8. 00000000000000000000

0000500000000000

1. 000000000000
2. 000000000000
3. 000000 00 00
4. 00 "+ 000000"
5. 00 000000
 - 00000000
 - 00000000
 - 00000000 + 70
 - 0000"0000"
6. 0000000000000000
7. 0000000000
8. 00 X 00000000000000000000
9. 00 "0000"
10. 00000000000000
11. 0000Stripe00000

00006000000000000000

1. 000000000000
2. 0000000000000000
3. 000000 000000000000
4. 00000000
 - 00ID00000000

- 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. 0000000000000000000000

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: Authentication, Authorization, and Data Protection
- API Security: Encryption, Rate Limiting, and Input Validation

API Design

- API Design: RESTful, SOAP, and GraphQL
- API Design: Versioning, Error Handling, and Documentation

API Performance

- API Performance: Caching, Load Balancing, and Compression
- API Performance: Monitoring, Logging, and Profiling

API Integration

- API Integration: OAuth, OpenID, and SAML
- API Integration: Webhooks, Web Services, and Data Feeds

API Governance

- API Governance: Policy, Standards, and Compliance
- API Governance: Security, Privacy, and Accessibility

API Resources

- `integrations_mailjet` - Mailjet Integration
- `administration_configuration` - Administration Configuration
- `payments_transaction` - Payments Transaction
- `payments_process` - Payments Process
- `payment_system_guide` - Payment System API Guide

Table of Contents

OmniCRM Integration with Stripe and PayPal
OmniCRM Integration

Payment System Guide | Billing Overview | Payments Process | Payments Invoices

Table of Contents

Table of Contents

- Introduction - Stripe and PayPal Integration with OmniCRM
- Getting Started - Stripe and PayPal Setup
- Payment System - Configuration
- Billing Overview - Configuration
- Payments Process - Configuration
- Payments Invoices - Configuration
- Self-Care Portal - Configuration
- Reporting - Configuration

Table of Contents

- Payment Methods - Visa, Mastercard, American Express, Discover
- Payment Gateway
- Payment Processor

OmniCRM Integration

OmniCRM Integration

- Stripe and PayPal
- Payment Methods - Visa, Mastercard
- 4-Digit PIN
- Payment Gateway
- Payment Processor
- Reporting

□□□□□□

□□□□□□

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 Overview

Prerequisites

- Development Environment (VS Code, Docker)
- Stripe Elements / PayPal SDK
- OmniCRM Integration
- PCI DSS Compliance
- Database Setup

2. Implementation

Backend "API" Layer

1. API Endpoints
 - GET /api/products
 - POST /api/orders
 - POST /api/cvc
2. Integration
 - Connect to OmniCRM
 - Integrate Stripe (pm_1A2B3C4D)
 - Connect to OmniCRM
3. Security
 - Implement OmniCRM Authentication
 - Implement 4-Step Security
 - Implement OmniCRM Security

3. Frontend

Payment Form

Input: Visa Card Number 1234

Payment Gateway

Payment Confirmation

- 11111111111111111111
- 22222222222222222222
- 33333333333333333333

1111111111

1111111111

- 1111111111
- 2222
- 3333
- 4444444444444444

11111111

1. 11111111111111111111
2. 111111 222222 : 3
3. 11 "111111"
4. 11111111

Visa1111111156781111111111111111

1111111111111111

111111

1111111111111111111111111111111111

11111111

11111111111111111111

第1部分

- 第一部分
- 第二部分 : 3
- 第三部分

第2部分

第一部分

第一部分

Visa 1234 12/2026

⚠ 警告信息

[] []

" " 部分

第3部分

部分

Stripe

部分

- 第一部分 - 第二部分
- 第一部分 - 第二部分
- 第三部分 - 第四部分

第四部分

OmniCRM

XXXXXXXXXX

XXXX XX → XXXXXXXX XXXXXXXXXXXXXXXX

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

XXXX

- XXXX - XXXX30/60/90XXXXXXXX
- XXXX - XXXX
- XXXX - XXXXXXXXXXXXXXXX

XXX

- XX - XXXXXXXXXXXXXXXXXXXX
- XX - XXXXXXXXXXXXXXXXXXXXMailjet

XXXX

XXXXMailjetXXXXXXXXXXXX

- XXX**60** - XXXX

- 30日 - 1ヶ月
- 7日 - 1週間
- 1日 - 1日

Mailjet

- 1日
- 4日
- 1日
- 1日

`integrations_mailjet`

1. 1日
2. 1日
3. 1日 → 1日
4. 1日 "1日"
5. 1日
6. 1日
7. 1日/1日

1. 1日
2. 1日 → 1日
3. 1日
4. 1日

5. 問題

6. 問題

問題

問題

- 問題
- 問題
- 問題

問題

問題

問題

1. 問題

- 問題
- 問題
- 問題
- 問題

2. 問題

- 問題
- 問題
- 問題“問題”問題

3. 問題

- 問題 - 問題
- 問題 - 問題
- 問題 - 問題

問題

1-3 問題

- 問題
- 問題

- 0000000000000000

▣4-7▣▣

- 0000000000000000
- 00000000
- 0000000000

▣8-14▣▣

- 00000000000000
- 0000000000000000
- 000000000000000000000000000000

▣15▣▣▣▣▣

- 00000000000000
- 000SIM000000000000
- 000000
- 00000000000000

▣▣▣▣▣▣▣

▣▣▣▣▣▣▣▣

- 0000300000
- 000000000000
- 00000000
- 00000000000000

▣▣▣▣▣▣▣▣

▣▣▣▣▣▣▣▣▣▣▣▣▣▣

1. 000000000000
2. 000000
3. 000000000000
4. 000000000000

5. 銀行振込

銀行振込

銀行

OmniCRM銀行振込

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

- 銀行振込
- 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
- null

payments

2. 查詢

查詢

- 查詢
- 查詢
- 查詢 `invoice_id` 查詢
- 查詢

查詢

3. 查詢

- 查詢
- 查詢
- 查詢

查詢 → 查詢

1. 查詢

查詢

- 查詢 Stripe 查詢
- 查詢 POS 查詢
- 查詢



2. 查詢

查詢 Stripe

- 查詢
- 查詢 "查詢 #1234 查詢"
- 查詢 -\$45.00 查詢
- 查詢 `invoice_id` 查詢

查詢

3. 設定

- 設定  設定
- 設定 

設定 → 設定

Stripe 設定

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

設定

1. 設定/POS/設定
2. 設定
3. 設定
4. 設定
5. 設定

設定

設定

設定 **1** 設定

設定

1. 設定
 - 設定 → 設定
 - 設定 設定 → 設定 "設定"
 - 設定 Stripe 設定

- 確認金額

2. 請求書

- 請求金額
 - 請求金額 \$50.00
 - 請求金額 \$45.00
 - 請求金額 \$95.00

3. 請求内容

- 請求内容 → 請求 "請求内容"
- 請求内容
- 請求 "請求"
- 請求 #INV-2025-001234 請求金額 \$95.00

4. 請求金額

- 請求金額 #INV-2025-001234
- 請求金額 \$0.00

5. 請求内容

- 請求内容
- 請求内容 PDF 請求内容

6. 請求内容

- 請求内容
- 請求内容
- 請求 "請求" 請求
- 請求内容
- 請求 "請求"
- Stripe 請求

7. 請求内容

- 請求内容 "請求"
- 請求内容 \$95.00

請求内容

請求書 2 請求内容

請求内容

1. 請求内容

- 2025年1月31日
- 2025年1月31日
- 2025年1月31日

2. 2025年1月31日

- 2025年1月31日
- 2025年1月31日
- 2025年1月31日
- 2025年1月31日

3. 2025年1月31日

- 2025年1月31日 → 2025年1月31日
- 2025年1月31日
 - 2025年1月31日
 - 2025年1月31日
 - 2025年1月31日
 - 2025年1月31日
 - 2025年1月31日
 - 2025年1月31日

4. 2025年1月31日

- 2025年1月31日
- 2025年1月31日 \$0.00

5. 2025年1月31日

- 2025年1月31日
- 2025年1月31日

6. 2025年1月31日

- 2025年1月31日
- 2025年1月31日/POS 2025
- 2025年1月31日

2025年1月31日

2025年1月31日

2025年1月31日

1. 2025年1月31日

- 0000 2 0
- 00000 \$10 00

2. 00000000

- 00000 → 00 000 → 00 00
- 00 "+ 0000"
- 00000 00
- 00000 00000000000000
- 000"000000"
- 000"1 0 8-9 00 2 00000"
- 0000010.00
- 00 "0000"

3. 000000

- 00000 **-\$10.00** 000000
- 00000 000
- 00000 000000 -\$10.00

4. 000000

- 00000000000000000000
 - 0000000000000000
 - 0000000000000000
- 0000000000
 - 00 00 → 00 "000000"
 - 0000000000000000
 - 0000000000000000

5. 000000

- 000000000000\$35.00
- 0000000000

000 000000000000000000000000000000000000

00000 **4**0000000000 - 000000

000 00000000000000000000 - 0000000000

1. 00000000

- 0000000000

- Stripe 查詢“發票”

- 查詢發票的步驟

2. 查詢發票

- 查詢發票 的 日期
- 查詢 日期 → 日期 “YYYY-MM-DD”
- 查詢發票的日期範圍
- 查詢

3. 查詢發票

- 查詢發票的 “發票號”
- 查詢發票的日期範圍

4. 查詢發票

- 查詢 日期 日期
- 查詢發票的日期
- 日期 “MM” 日期
- 查詢發票的日期範圍
- 日期 “YYYY”
- Stripe 查詢發票

5. 查詢發票

- 查詢發票的 “發票號”
- `payment_reference` 查詢 Stripe 發票 ID

查詢 查詢發票的日期範圍

查詢 5 查詢發票的日期範圍

查詢 查詢發票的日期範圍 - 查詢發票的日期

1. 查詢發票

- 日期 #INV-2025-001234 查詢發票的日期
- 查詢發票的日期
- 查詢

2. 查詢發票

- 查詢 日期 日期 → 日期 日期
- 查詢發票的日期
- 查詢發票 (日期)

- 0000
- 000000

3. 000000

- 000 00 00
- 0000000000000000000000 000
- 000000 00000
- 0000000000

4. 0000000000

- 00 00 → 00 "00000000"
- 000000000000
- 0000000000000000"00"0000000000
- 00 "0000"
- 0000000000000000

5. 000000000000

- 000000000
- 000000000000
- 0000000000000000

000 0000000000000000000000000000000000000000000000000

0000 **6**0000000000000000

000 0000000000000000000000

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"

- 項目 "項目名"
- 項目 #1234 項目 "項目名"

4. 項目名

- 項目 #1235 項目名
 - 項目"項目名 2025-02-10 - 項目 #001"
- 項目 #1236 項目
 - 項目"項目名 2025-02-10 - 項目 #001"

5. 項目名

- 項目 項目 項目
- 項目名
- 項目名

6. 項目名

- 項目 項目 → 項目名 項目 \$300.00
- 項目名 "項目" 項目

項目 項目名

項目名

項目名

項目名

- 項目名
- 項目名
- 項目名
- 項目名

項目名

- 項目名 1 項目
- 項目名
- 項目名
- 項目名
- 項目名 PDF

□□□□

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

□□□□

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

□□□□□□□□□□□□

□□□□

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

□□□□

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

□□□□

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

□□□□□

□□□□□

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

□□□□□□

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

□□□□□

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

□□□□□□□□□□

□□□□□□□□□□□□

□□□

- □□□□□□□□□□□□□□
- □□□□□□“□□□□□□□□□□”

□□□

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

1. 在 Stripe 网站上注册
2. 创建 Stripe 账户
3. 创建 Stripe 商户 - 商户名称和地址
4. 创建 Stripe 商户 - 商户名称和地址

Stripe

- Stripe 商户名称和地址
- Stripe 商户名称和地址
- Stripe 商户名称和地址 - Stripe 商户名称

Stripe 商户名称和地址

Stripe

- Stripe 商户名称和地址
- Stripe 商户名称和地址
- Stripe 商户名称

Stripe

1. 在 Stripe 网站上注册
2. 创建 Stripe 账户
 - Stripe
 - Stripe
 - Stripe 商户名称/地址
 - Stripe 商户名称
 - Stripe 商户名称

Stripe

- Stripe 商户名称和地址
- Stripe 商户名称和地址
- Stripe 商户名称和地址 POS 商户名称和地址
- Stripe 商户名称和地址

□□□□□□

□□□

- □□□□□□□□□□□□ → **B** □□□□□□□□□□□□

□□□

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

□□□□□

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

□□□□□





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


















□□□□□

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

CGRateS

CGRateS OmniCRM         

 OmniCRM  (CGRateS)    

     CGRateS          
Playbooks  - 

 -      CDR 

 -     

 **ID** -   “ ”  “ ” 

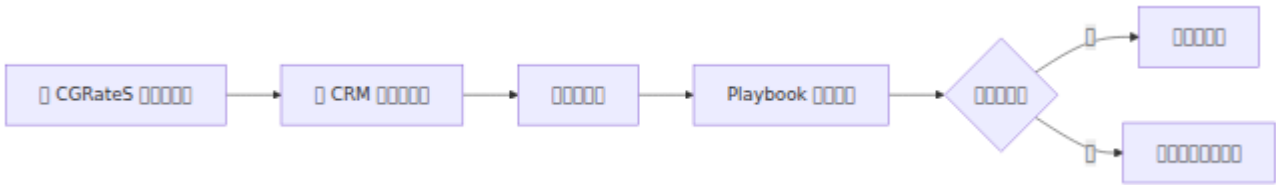
 -    

 -    “+5 ” 

 -        

    CGRateS   
 

   Python  CRM  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: "
#

```

-
-
-

配置示例

```
# 配置项  
[1]:  
  ID: "Data_5GB_5days_uuid_abc123"  
  [data]: *data  
  [value]: 5368709120 # 5GB配置  
  [start]: 2024-12-29  
  [duration]: 10  
  
[2]:  
  ID: "Data_10GB_30days_uuid_def456"  
  [data]: *data  
  [value]: 10737418240 # 10GB配置  
  [start]: 2025-01-24  
  [duration]: 10  
  
# 配置项  
# 配置项
```

配置项CGRateS - 配置项

配置项 *topup 和 *topup_reset

配置项 ID 配置项

*topup - 配置项

***topup** 配置项

配置项

- 配置项 ID 配置项
- 配置项
- 配置项
- 配置项

□□□

```
# □□□□□
□□□
  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 □□□□□□

製品 ID 取得

製品 ID 取得

製品 ID 取得

OmniCRM 製品 ID 取得 API Web UI

製品 ID

```
{製品ID}__{製品ID}
```

製品 ID

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

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

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

製品 ID 取得 - 製品 ID 製品 ID API 製品 ID

API 製品 ID

製品 ID OmniCRM API 製品 ID CGRateS 製品 ID 製品 ID ID 製品 ID _hr 製品 ID

```

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

        // 容量使用率
        "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 関数

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. 容量使用率取得

```

# 容量使用率
id_hr = descriptive_name.replace("_", " ") # "AU Data
Domestic"

```

3. 条件分岐

```
# データタイプ
if balance_type == "*data":
    original_value_hr = convert_bytes_to_gb(original_size) #
    "100 GB"

# 音声タイプ
elif balance_type == "*voice":
    original_value_hr = convert_ns_to_minutes(original_size) #
    "3000 min"

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

4. 条件分岐

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

5. 条件分岐

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

Web UI

残り容量 容量

```

// ID: "AU_Data_Domestic__107374182400"
// ID: 53687091200

// ID
<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>

```

UI

1. UI - UI "50 GB of 100 GB" UI "50 GB"
2. UI - UI
3. UI - ID UI
4. UI - UI *topup UI 70 GB UI 100 GB
 - o ID UI "AU_Data_Domestic__107374182400" UI 100 GB
 - o UI 170 GB
 - o UI "170 GB (70 GB rolled over + 100 GB new)"

UI - UI ID

UI UI UI UI

```

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

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

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

```

100GB - 1024*1024*1024

100GB*1024*1024*1024

```
# 平衡情報
{
  "BalanceId": "PAYG_Monetary_Balance",
  "BalanceType": "*monetary",
  "Units": 5000 # $50.00
}

# UI 表示情報
# "$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 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": "{{ balance_id }}",
          "Value": 5368709120,
          "ExpiryTime": "+5d",
          "Weight": 10
        }
      }]
    }
}
```

充值套餐接口配置

```
# 5GB 15GB
1:
ID: "Data_5days__5368709120_a1b2c3d4"
: 5368709120
Value_hr: "5 GB"
OriginalValue_hr: "5 GB"
Remaining_hr: "5 GB of 5 GB"
: 2024-12-29

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

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

# 15GB
# UI "Data 5days - 5 GB of 5 GB"
```

2 ID *topup

ID *topup

```
- 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 1 1 1
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 数据库连接池配置

Blocker

Blocker: true

1. 数据库连接池配置
2. 数据库连接池配置
3. 数据库配置 → INSUFFICIENT_CREDIT_BALANCE_BLOCKER

□□□□

- □□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□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 →

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 充值
      {
        "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
- 普通充值 → 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 0200 000
3. **700-1200** 000 00 Hard_Spending_Cap 0000 \$0.10 = \$50 0500 000
4. **1200** 0000 Hard_Spending_Cap 00 → 00000000

000000 1200 000000000 \$50

0000000000

1. 0000000000

```

"BalanceWeight": 9999 # 0000000000

```

2. 000000000000

```

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

```

3. 000000000000

- 00 ActionTriggers 0 80% 090% 0100% 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

数据包的列表

包 1:

ID: "Data_Package_A"

大小: 5GB

创建日期: 2024-12-25

数量: 11 # 数据包 = 数据包

包 2:

ID: "Data_Package_B"

大小: 10GB

创建日期: 2025-01-15

数量: 10 # 数据包 = 数据包

包 1 的数据包数量 11 > 包 2 10

数据包列表

包 1

包 1 数据包列表

包 1 数据包列表 "5GB 5" 数据包列表

包 1

```

- name: 新增 UUID 数据表
  set_fact:
    uuid: "{{ 99999999 | random | to_uuid }}"

- name: 新增 5GB 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",
          "Categories": "*any",
          "Balance": {
            "ID": "Data_5GB_{{ uuid[0:8] }}",
            "Value": 5368709120,
            "ExpiryTime": "+120h", # 5天
            "Weight": 10
          }
        }]
      }

```

配置

- 12月24日 00:00:00 → 新增 5GB套餐 12月29日 00:00:00
- 12月25日 00:00:00 → 新增 5GB套餐 12月30日 00:00:00
- 套餐权重配置为 10

新增 2个套餐

新增 “新增 5GB 套餐” 数据表

新增

```

- 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 充值接口 → 70GB 20GB + 50GB 充值接口 20 天 + 30 天

充值 3 充值接口 充值接口 充值接口

充值 “充值 100GB 充值” 充值接口 100GB 充值接口

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**□□□□□□□□□□□□□□

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

□□□□

```
# 000000000000
- name: 0000000000
  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
            "Weight": 20
          }
        }
      ]
    }
}
```

```
# 000000000000
- name: 0000000000
  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
            "Weight": 10 # 0000
          }
        }
      ]
    }
}
```

□□□□

- □□ 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 - 00000000
- *voice - 00000000
- *sms - 00000000
- *monetary - 0000000000
- *generic - 0000

Units - 00/000000000000000000000000

ExpiryTime - 00000000

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

Balanceld - 0000000000 ID = 000000 ID = 0000

Weight - 0000000000 = 000000000000

Blocker - 000 true000000000000

Disabled - 000 true000000000000

00 Python 000000000000

000000000000 Python 0000 cgrateshttpapi 0000000000000000
 OCS_Obj.SendData() 000000

□□□□

```
import cgrateshttpapi
import time

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

□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□ □□□□□□□

- □□□□□ - □□/SMS □□□□□□□□□□ `Dest_International_UK` □
- **PLMN** □□□ - □□□□□□□□□□□□□□ `Dest_PLMN_OnNet` □ `Dest_PLMN_US_Verizon` □

□□□□□

- □□/SMS □□ → □□□□□□□□□□□□□□□□□□ TO □
- □□□□ → □□ PLMN □□□□□□□□□□□□□□□□□□ FROM □

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- PLMN □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- PLMN □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□ **CGRateS** □□□□□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
# 1GB in bytes
1_GB = 1 * 1024 * 1024 * 1024 # 1073741824 bytes
100_GB = 100 * 1024 * 1024 * 1024

# 1 minute in seconds
1_minute = 60 * 1000000000 # 60 seconds
3000_minutes = 3000 * 60 * 1000000000

# 3000 SMS
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": [
      # 1000000000000000
      {
        "Identifier": "*reset_account",
        "Weight": 700
      },
      # 100 100GB 10000
      # 10000000000 PLMN 1000000000000000
      # 100000000000000000000000 PLMN
      {
        "Identifier": "*topup_reset",
        "BalanceId": "AU_Data_Domestic__" + str(100 * 1024
* 1024 * 1024),
        "BalanceType": "*data",
        "DestinationIDs": "Dest_PLMN_OnNet", # 10000 PLMN
[mcc505.mnc057]
        "Units": 100 * 1024 * 1024 * 1024,
        "ExpiryTime": "*month",
        "BalanceWeight": 1200,
        "Weight": 90
      },
      # 100 3000 100000000
      {
        "Identifier": "*topup_reset",
        "BalanceId": "AU_Voice_Domestic__" + str(3000 * 60
* 10000000000),
        "BalanceType": "*voice",
        "DestinationIDs":
"Dest_AU_Mobile;Dest_AU_Fixed;Dest_AU_TollFree;",
        "Units": 3000 * 60 * 10000000000,
        "ExpiryTime": "*month",
        "BalanceWeight": 1200,
        "Weight": 89
      },
      # 100 3000 10000
      {

```

```

        "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 SMS PLMN 1000000000

Python

1000000

- **ActionsId** - CRM product_slug
- **Overwrite** - True ID
- **Tenant** - CGRateS

- **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

```

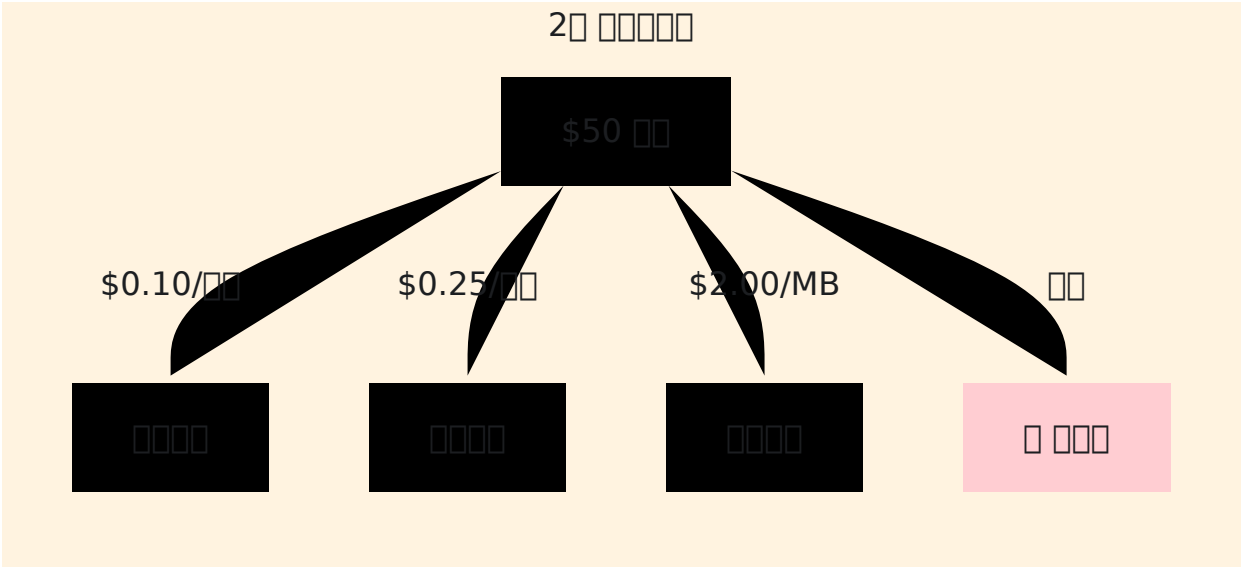
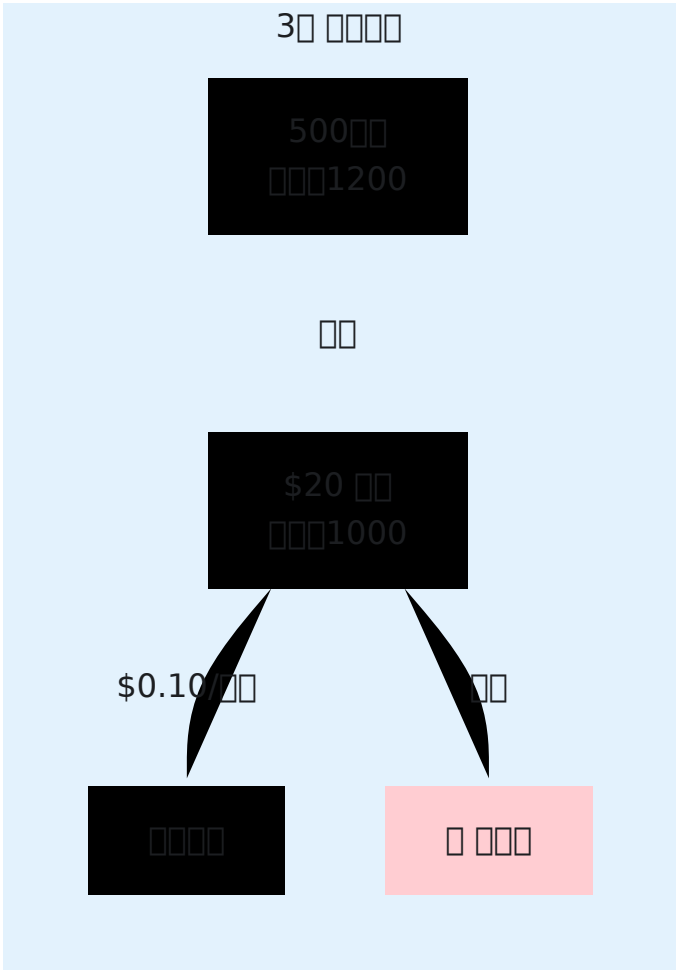
- name: 
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "APIerSv1.ExecuteAction",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "ActionsId": "Action_Name_Here"
        }]
      }

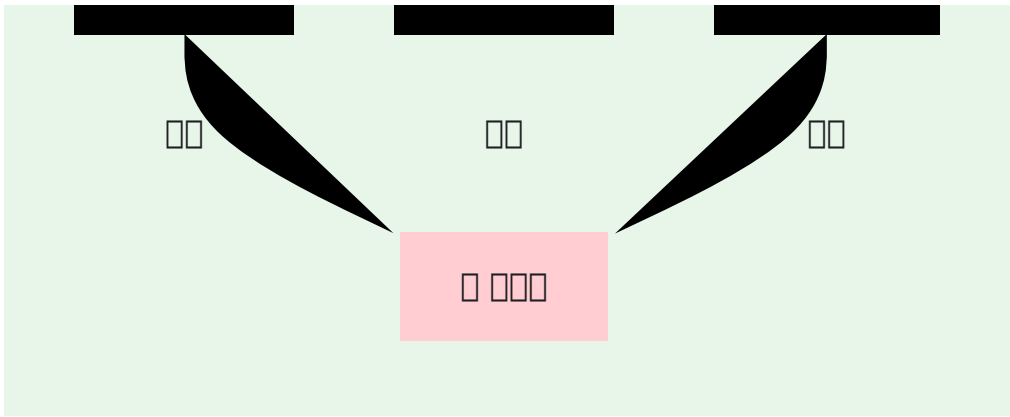
```

CGRateS

VS VS

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





RAM 1GB

RAM 1GB GB

RAM

```

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    -    
                 

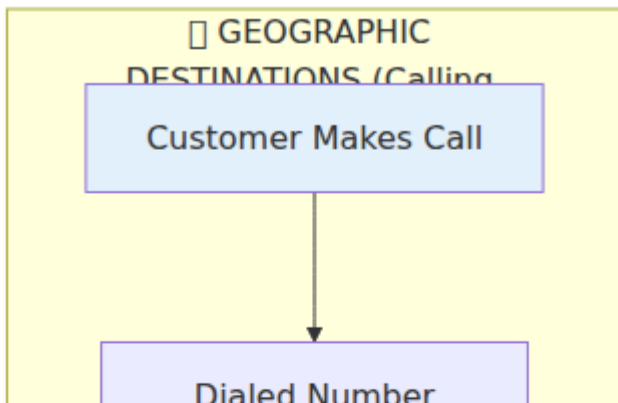
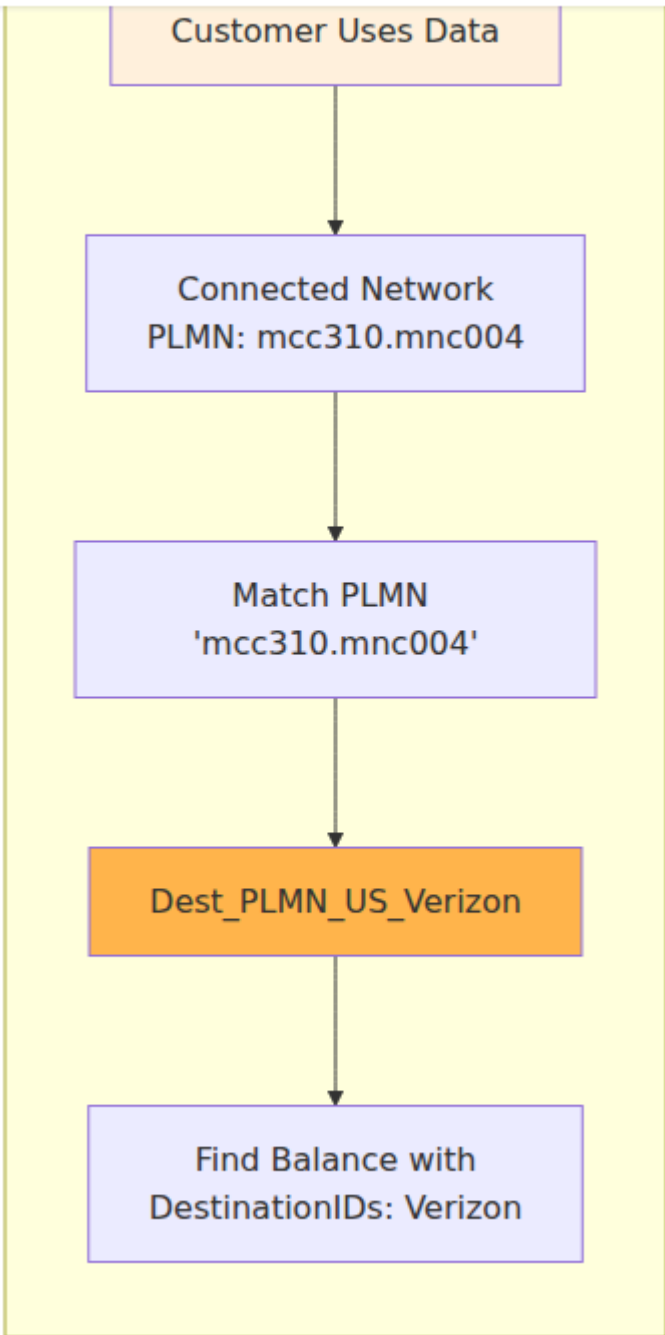


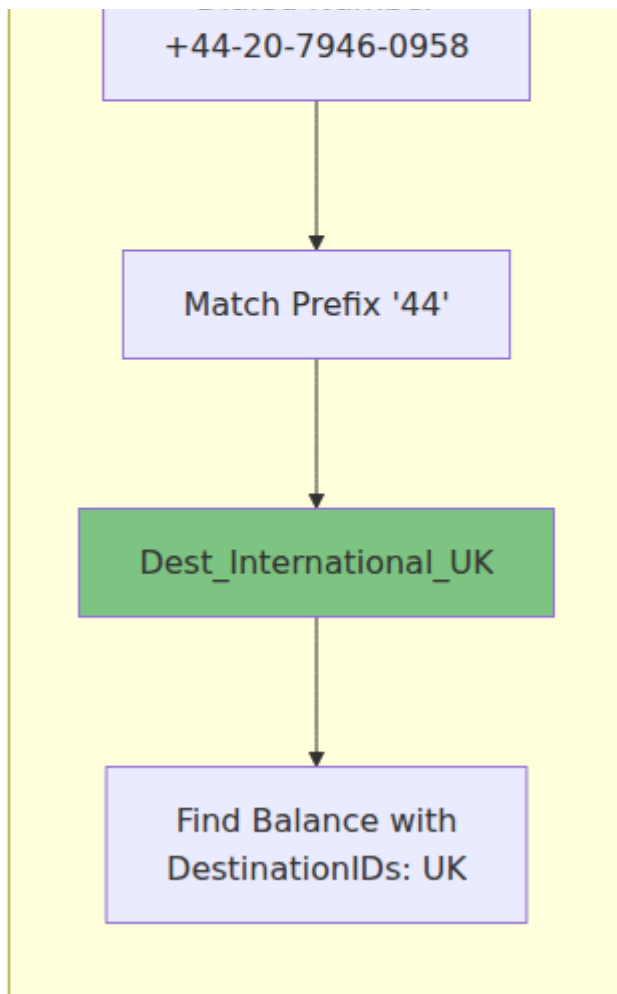
CGRateS               

-                  
- **PLMN**                

-                             
-                             





□□□□

- □□/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`

`000000000000`

`0000000000000000 SIM 000000000000`

```
# 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: Tariff Plan Data

Country	Operator	Plan Name	Features
USA	Verizon	"100GB Unlimited"	Unlimited SMS
USA PLMN	Verizon	"100GB Unlimited"	Unlimited
USA PLMN	Verizon	"Verizon Unlimited 5GB"	Unlimited/SMS
USA PLMN	Verizon	"Verizon Unlimited 5GB"	Unlimited/SMS
USA PLMN	Verizon	"10GB Unlimited"	Unlimited/SMS

Table 2: Tariff Plan Data

Table 3: Tariff Plan Data

Table 3: CGRateS Table with ID, PLMN, and TariffPlan columns. The table contains data for various countries and operators.

Table 4:

- Table 4: Data for various countries and operators.
 - Table 4: Data for various countries and operators.
 - ⚠️ Table 4: TariffPlan column - Data for various countries and operators.
- LoadTariffPlanFromStorDb Table

Table 5: Tariff Plan Data

Table 6:

```
# 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"]
  }]
})

```

[[{"method": "CacheSv1.ReloadCache", "params": [{"DestinationIDs": ["Dest_PLMN_Zone_Europe"]}]}]]

2 [[{"method": "CacheSv1.ReloadCache", "params": [{"DestinationIDs": ["Dest_PLMN_Zone_Europe"]}]}]]

[[{"method": "CacheSv1.ReloadCache", "params": [{"DestinationIDs": ["Dest_PLMN_Zone_Europe"]}]}]]

```

# Remove Germany from Europe zone
OCS_Obj.SendData({
  'method': 'ApierV2.SetTPDestination',
  'params': [{
    "TPid": tpid,
    "ID": "Dest_PLMN_Zone_Europe",
    "Prefixes": [
      "mcc208", # France
      "mcc222", # Italy
      "mcc214", # Spain
      "mcc234" # UK
      # Germany (mcc262) REMOVED
    ]
  }]
})

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

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

```

1. 在配置文件中，找到 `Dest_PLMN_Zone_Europe` 配置项，
 删除 `PLMN` 配置项

2. 发送 **3条/SMS** 短信

3. 验证 SMS 是否发送成功

```

# 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"`

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
-
-


```

#
=====
# 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

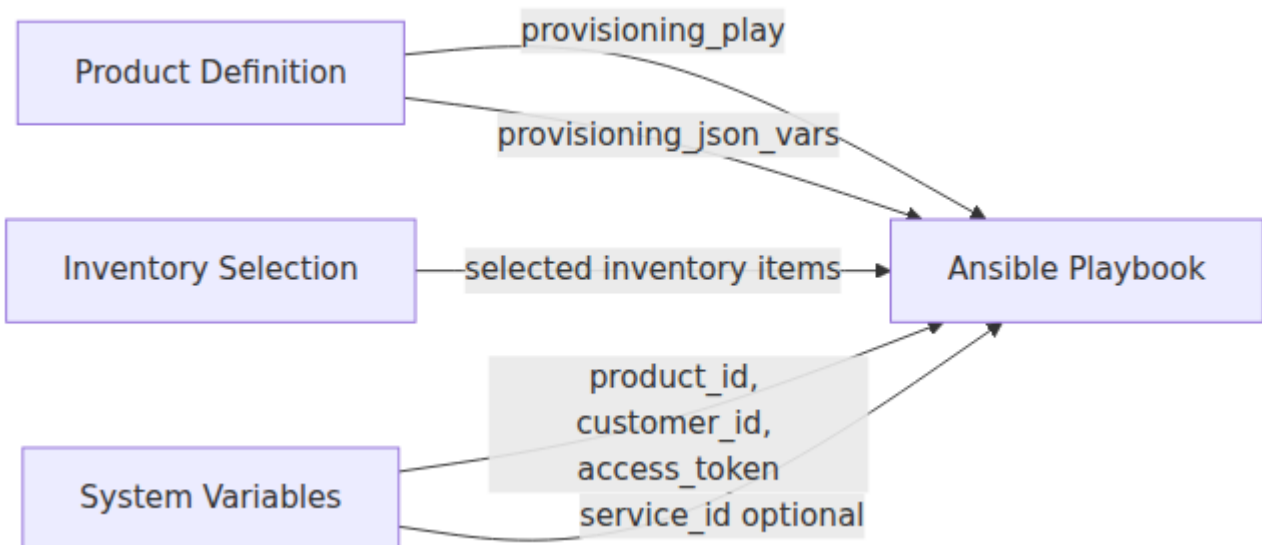
Playbooks

 Playbooks OmniCRM playbook
 -

Playbooks

 OmniCRM

- provisioning_play
- 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 Tasks

```

# 1. Provisioning SIM card 1
# provisioning_play: play_simple_service
- CRM
- OCS
- SIM
-

# 2. Provisioning bundle 3
# provisioning_play: play_bundle_internet_tv_voice
-
- IPTV
- VoIP
-
-

# 3. Provisioning topup 0
# provisioning_play: play_topup_no_charge
- service_id
- OCS
-
-

# 4. Provisioning CPE 0
# provisioning_play: play_prov_cpe_mikrotik
-
-
-
-

```

OmniCRM `playbook` tasks

Plays Tasks

OmniCRM Plays Tasks tasks OmniCRM playbooks tasks

Playbook

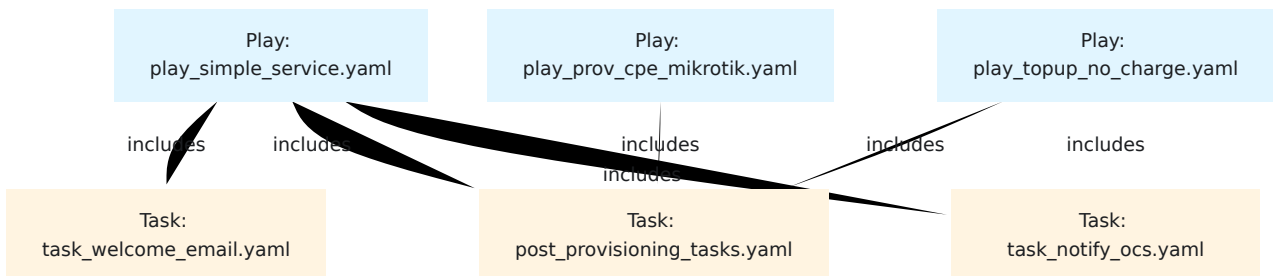
Plays `OmniCRM-API/Provisioners/plays/` playbooks

- `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 `playbooks`

`playbooks` OmniCRM playbooks `playbooks` `playbooks`

`playbooks`

`playbook` `playbook`

```

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

  tasks:
    - name: Main block
      block:
        # task_block

      rescue:
        # task_block

```

name

name

UI description `playbook_description`

hosts: localhost

OmniCRM playbooks on localhost connecting to API via SSH

gather_facts: no

Ansible will not gather facts

- `ansible.builtin.debug`
- `ansible.builtin.ping`
- `ansible.builtin.assert`

become: False

Do not use sudo to run API tasks

tasks

tasks: include vars

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

vars: `crm_config.ocs.cgrates`, `crm_config.crm.base_url`

`crm_config.yaml` content

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

        # 2. Get Product information from CRM 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
            validate_certs: no
            register: api_response_product

        # 3. Get Customer information from CRM API
        - 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. Set package facts
        - 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. 增加余额
- 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 获取当前时间
- 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: 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:

# []/[]
- 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"
```

Topup/Recharge

```

- 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. 获取服务信息
    - 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. 获取产品信息
    - 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. 设置服务事实
    - 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

送信メッセージの送信先をONTに

```

- 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. 配置 WiFi

```
- name: Set WiFi SSID
  set_fact:
    wifi_ssid: "Network_{{ cpe_inventory_id }}"

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

- name: Create WiFi PSK
```

```

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

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

# 7. 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. 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

```

□□□□□

□□□□□□□□□□ playbooks□□□□□ plays □□□□□□□□□□□□□□□□

□□□□□□□□

task_welcome_email.yaml - □□□□□□□□□□□□

```

# 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

```

□□□□□□□□

□□□□□□/□□□□

- **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') }}"`

□□□□□□□□

- **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`

□□□□□□

UUID □□□□□□

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

□□□□□□

```
- 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
```

□□

□□□□□□□□

```

# 00000000000000000000
# 000000 15 0000000000 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. 00000000000000000000
2. 000000000000000000000000000000
3. 00000000000000 API 00
4. 00000000000000000000000000
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 service
service_uuid: "{{ service_uuid }}"

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

```

tasks

```

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

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

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

```

modules

OmniTouch modules

1. create account
2. create Ansible playbooks
3. create playbooks
4. create

OmniTouch modules

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
- 安裝 Playbooks
- 配置 Playbooks
- 執行 Playbooks

配置

配置 Playbooks

1. 安裝 Playbooks

安裝 Playbooks 到本地目錄

2. 執行 Playbooks

執行 playbook 使用 `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
- 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

ignore_errors: True

测试 ignore_errors: True 测试

测试

测试 when 测试

```
when: service_creation_response is defined
```

测试

测试

1. 测试
2. 测试
3. 测试 OCS/测试
4. 测试

测试

测试

- 测试 (action == "deprovision") → 测试
- 测试 → 测试

测试

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 playbook

- 環境構築
- OCS
- 環境構築
- OCS
- 環境構築

環境構築の API

- OCS CRM
- OCS
- 環境構築
- 環境構築
- 環境構築

Ansible

Ansible `block/rescue` blocks allow you to define a sequence of tasks to run, and a set of tasks to run if the main block fails. This is useful for handling errors and cleanup.

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. Rollback changes
    # 2. Notify user
    # 3. Playbook failed
    - 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/>>

목차

OmniCRM 백엔드 애플리케이션 개발을 위한 인증 전략

1. **JWT Bearer** 방식 - 백엔드 애플리케이션 Web UI와 연동

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 2024-01-01 10:00:00

API 2024-01-01

API 2024-01-01 10:00:00

API

- API
- API
- API
- API cron

API 2024-01-01

API 2024-01-01 `crm_config.yaml` API 2024-01-01 32
API 2024-01-01 `X-API-KEY` API 2024-01-01

API 2024-01-01

API 2024-01-01 `crm_config.yaml` API 2024-01-01

```

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> Introduction

- Ansible Playbook Introduction Play <concepts_ansible> Introduction Ansible Introduction service_id Introduction ICCID

MSISDN
UI

UI

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. ⚙️ **Ansible** - **Configuration Management**
4. 📖 **Ansible** - **Ansible playbook** **Automation**
5. 📖 **Ansible** - **Ansible**

📖 **Ansible**

📖 **Ansible**

- **Ansible** - **Configuration Management**
- **Ansible** - **Ansible**
- **Ansible** - **Ansible**

📖 **Ansible**

- **Ansible** - **Configuration Management** **FontAwesome** **Icons**
- **Ansible** - **Ansible**
- **Ansible** - **Ansible**
- **Ansible** - **Ansible** \$ **Icons**
- **Ansible** - **Ansible**

□□□□

- □□□□
 - □□□□ - □□□□□□□□□□/□/□□
 - □□□□□□ - □□□□□□□□□□□□
 - □□□□ - □□□□□□□□□□□□30 □□□□□□365 □□□□□□
- □□□□□
 - □□ - □□□□□□□□□□
 - □□ - □□□□□□□□□□
- □□□□□□
 - □□□□□ - □□□□□□□□□□□□□□ ID □□□□□□
 - □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□


□□□□□□

□□□□□□ Ansible □□□□□□□□□□

□□□□□

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

□□□□□

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

쿠폰

쿠폰은 할인율을 적용하여 결제 금액을 계산합니다.

- 할인율 10% 적용 시 \$50.00 → \$5.00
- 할인율 0% 적용 시 → \$0.00
- 할인율 → 결제 금액을 계산합니다

쿠폰 옵션

- **enabled** - 쿠폰이 활성화되어 있는지 여부 (true/false)
- **residential** - 주거용 고객에게만 적용되는지 여부
- **business** - 상업용 고객에게만 적용되는지 여부
- **customer_can_purchase** - 고객이 쿠폰을 구매할 수 있는지 여부 (true/false)
- **available_from** - 쿠폰이 유효해지는 날짜/시간
- **available_until** - 쿠폰이 유효하지 않는 날짜/시간

계약

- **contract_days** - 계약 기간 (30, 365, 0 등)
- **auto_renew** - 자동 갱신 여부
 - `prompt` - 고객에게 알림
 - `true` - 자동 갱신
 - `false` - 자동 갱신 안함

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

其他属性

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

其他属性

- **provisioning_play** - 用于 Ansible playbook 的 yaml 文件 `OmniCRM-API/Provisioners/plays/`
- **provisioning_json_vars** - 用于 Ansible playbook 的 JSON 文件 `customer_id` `product_id` `service_id` `access_token`
- **inventory_items_list** - 服务项列表 ['SIM Card', 'Mobile Number'] 每个项包含 ID 属性
- **relies_on_list** - 依赖项 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

□□□□□□□□□□

```
{
  "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 + □□□□□□□□□□□□□□□□□□□□□□□□□□

API Example

```
{
  "product_slug": "Mobile-Topup-5",
  "product_name": "PAYG £5 Topup",
  "category": "addon",
  "service_type": "mobile",
  "provisioning_play": "play_topup_monetary",
  "provisioning_json_vars": "{\"service_id\": \"\"}",
  "inventory_items_list": "[]",
  "retail_cost": 5,
  "retail_setup_cost": 0,
  "wholesale_cost": 0,
  "contract_days": 0,
  "residential": true,
  "business": false,
  "enabled": true,
  "customer_can_purchase": true,
  "icon": "fa-solid fa-coins",
  "features_list": "['£5 credit', 'Valid for 180 days']",
  "terms": "Valid for 180 days or until all credit is used. See our website for full rates",
  "comment": "£5 to use for Calls, SMS & Data"
}
```

API example `service_id` parameter

Introduction to Ansible

Ansible is a `API` Ansible playbook `playbooks`

Prerequisites

Prerequisites for running Ansible playbooks

1. **python provisioning_json_vars** - python module
2. **ansible** - `API` Ansible playbooks
3. **ansible-playbook** - `API` Ansible playbooks

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 卡 ID: 789 - 客户 ID: 101

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

Playbook 交互 ID 通过 CRM API 交互 ICCID IMSI MSISDN

Ansible 交互

示例 `extravars` 通过 Ansible playbook 交互

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

示例

示例

- `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

UI

UI configuration for `playbook` configuration

- `product_id` - ID
- `customer_id` - ID
- `service_id` - ID
- Modem Router ID
- `access_token` - API token

UI configuration for `play_seniors_package` IPTV configuration

UI

UI configuration

UI configuration for `OCS` configuration
OCS configuration: `CGRateS` configuration
ActionPlans configuration: `ThresholdS`

UI

UI configuration

UI

UI configuration

1. Residential - Business
 - Residential: `residential = true` | `business = false`
 - Business: `business = true`
2. Standalone - Bundle
 - Standalone: `category = standalone` | `bundle`
 - Addon: `category = addon`
3. Addon - Bundle

- `enabled = true` - 可購
- 開始日期 `available_from` 及 結束日期 `available_until` 欄位 - 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

- 查詢所有業務/非業務 `business = true` `category != "addon"`
- 查詢所有業務 `business = true` `category = "addon"` `service_type = "mobile"`
- 查詢所有業務/非業務

查詢

查詢所有業務/非業務

查詢所有業務

- **service_id** - 業務ID
- **customer_id** - 客戶ID
- **product_id** - 產品ID
- **service_name** - 業務名稱
- **service_type** - 業務類型 VoIP IPTV
- **service_uuid** - OCS/CGRateS 業務唯一識別碼
- **icon** - 業務圖示 FontAwesome

查詢業務狀態

- **service_status** - 業務狀態
- **service_provisioned_date** - 業務生效日期
- **service_active_date** - 業務啟用日期
- **service_deactivate_date** - 業務停用日期
- **contract_end_date** - 合約結束日期

查詢業務成本

- **retail_cost** - 零售成本
- **wholesale_cost** - 批發成本
- **service_billed** - 業務已開票 `true`
- **service_taxable** - 業務可課稅 `true`
- **invoiced** - 業務已開票
- **promo_code** - 業務優惠碼

Step 1: Overview

API Parameters

- `provisioning_play` - Ansible playbook (default: `play_psim_only`)
- `provisioning_json_vars` - Provisioning JSON variables
- `customer_id` - Customer ID
- `product_id` - Product ID
- `service_id` - Service ID
- `initiating_user` - Initiating user ID

Step 2: Request

Request Parameters

- `provisioning_json_vars` (required)
- Headers
- Body
 - `access_token` - API CRM JWT token
 - `initiating_user` - Initiating user ID
 - `customer_id`, `product_id`, `service_id`
- URL - `{inventory_type: inventory_id}`

Example

```
{
  "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 JWT token
  "initiating_user": 5      # Initiating user ID
}
```

Step 3: Response

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
- `access_token` - CRM API 的 JWT
- `initiating_user` - 發起用戶
- 服務 ID `{{ inventory_type }}: inventory_id`
- 提供 `provisioning_json_vars` 參數
- 其他參數

Playbooks 參照

- 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 }}`
- 提交 `POST /crm/transaction/`
- 更新 `PUT /crm/service/`
- 更新 `PATCH /crm/service/{{ service_id }}`
- 更新 `PATCH /crm/inventory/inventory_id/{{ inventory_id }}`

如何編寫 **playbook**

參照

1. Playbook 參數 `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. 其他

參照 `Provision` 或 `Provision_Event` 參照

OCS 參照

OCS 查詢 `CGRateS` 獲取 CRM 查詢 OCS 獲取

- `number` - `number`DID `number`
- `number` - `number` MB `number`
- `number` - `number` GB `number`
- `number` - `number` \$30 `number` 10GB `number`
- `number` - `number`

CRM `number` OCS `number`

`number`

`number` OCS/CGRateS `number`

`number`

`number` UI `number` API `number`

1. **API** `number` - `number` `GET /crm/service/{service_id}` `number` UI `number`
2. `number` - API `number` `service_uuid`
3. **CGRateS API** `number` - `number` CGRateS `number`
 - i. **Get_Balance(service_uuid)** - `number` `BalanceMap`
 - `number` `number`
 - `number`ID `number` ID
 - `number` `custom_Name_hr` `custom_Expiration`
`custom_Description_String`
 - ii. **Get_ActionPlans(service_uuid)** - `number`
4. `number` - CGRateS `number`

```

{
  "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

Web UI

- 3 ServiceUsage
- GET /crm/service/{service_id} CGRateS
-

API /

DID 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

API

Action

[[ActionPlans]] [[Ansible]] Action [[ExecuteAction]]

[[]]

[[\$10 PAYG]]

[[\$0]]

[[/]]

[[ThresholdS]]

[[PAYG / \$0]]

[[CRM]] [[OCS]]

[[CRM API]] [[OCS]] [[Balances]] [[ActionPlans]] [[Actions]] [[ThresholdS]]

[[CRM API]] [[ActionPlans]] [[Ansible Playbooks]] [[CRM]] [[ActionPlans]] [[Ansible Playbooks]]

[[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
-

- `ocs` - `ocs` / `ocs` `ocs`
- `ocs` - `ocs`

ocs **ActionPlans** ocs

- `ocs` "ocs"
- `ocs`

ocs **ActionPlans**

ocs UI ocs **ActionPlans**

ocs **ActionPlan** ocs

1. ocs **ActionPlansTable** ocs "ocs"
2. ocs "ocs"
3. ocs DELETE /crm/oam/remove_action_plan/{action_plan_id}
4. API ocs `ocs.Remove_ActionPlan()` ocs **CGRateS** ocs **ActionPlan**
5. ocs "ocs {service_id} ocs **ActionPlan** {ActionPlanId}"
6. **ActionPlan** ocs

ocs

1. ocs **ActionPlansTable** ocs "ocs"
2. ocs "ocs"
3. ocs
 - ocs **ActionPlanId** ocs `product_id`
 - ocs
 - ocs `playbook`
 - ocs
4. ocs
5. ocs

ocs

ocs `auto_renew` ocs

- `auto_renew = "true"` ocs - ocs **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“”
-

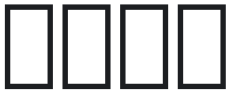
ActionPlans

CRM

GET /crm/service/expiring?threshold=7

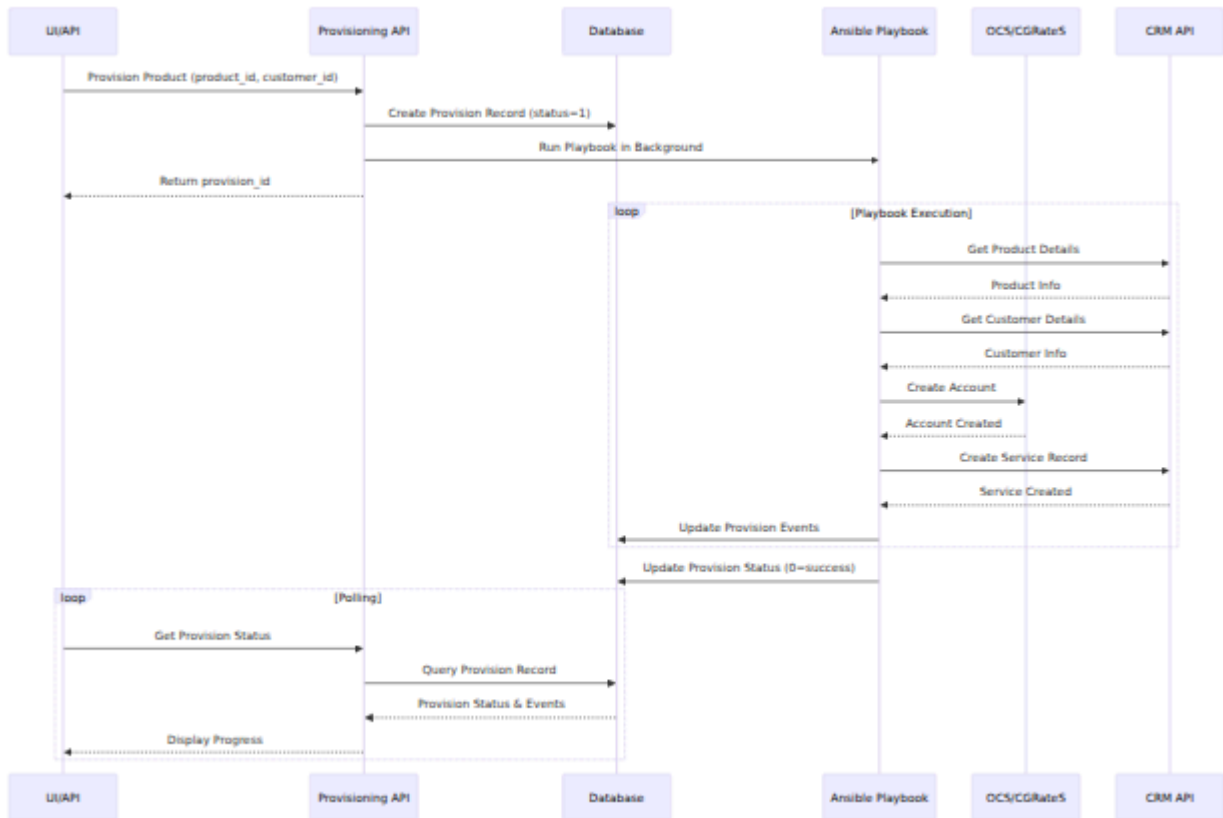
-
- ActionPlans
- ActionPlans

-
-
-
-



OmniCRM **Ansible** SIM Provisioning

SIM Provisioning <concepts_sim_provisioning> SIM Provisioning eSIM



00

Ansible `provisioning_play`
<code>provisioning_play</code>

00

OmniCRM Ansible
OCS/CGRateS API

1. UI/API
2. OCS

00000

- 0 -
- 1 -
- 2 -
- 3 - `ignore_errors: true`

OmniCRM

0000000

0000000

- `provisioning_play` - Ansible .yaml

- `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 测试

测试环境

- 测试 UI 环境
- 测试环境
- API 测试 `PUT /crm/provision/`

测试“API”

测试环境“API”测试环境

1. UI 测试

测试环境

- `provisioning_play` - 测试 Ansible 脚本
- `inventory_items_list` - 测试数据 ['SIM Card', 'Mobile Number']
- `provisioning_json_vars` - 测试数据

2. 测试环境

测试 `inventory_items_list` 测试数据

3. 测试数据

JavaScript 测试 `PUT /crm/provision/` 测试

```
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

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12 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. □□□□□□ □□ □□□□
3. □ OCS □□□□□□□□
4. □□ CRM API □□□□□□
5. □□□□□□□□
6. □□□□□□□□ rescue □□□□ □□□□ □□ □□

□□□□□□□□□□

□□□□◆◆◆◆ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□ □□□□□□□□ □□□□□□□□

□□□□□□□□

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□□□□□□□□□□□□□□

- 通过 SMS/email 通知
- 配置项 `ignore_errors: True`
- 配置项

通知

配置项

- 通过 YAML 配置
- 配置项
- 配置项
- 通过 Ansible 配置

配置项

- Ansible 配置
- 通过 `stdout` 配置
- 通过 `stderr` 配置
- 通过配置项
- 配置项

配置项

配置项 2 配置项 `crm_config.yaml` 配置项 `provisioning.failure_list` 配置项

配置项

- 配置项
- 配置项/配置项
- 配置项
 - 配置项
 - 配置项
 - 配置项
- 配置项/配置项
- 配置项 Ansible 配置项



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": "{...}"  
    }  
  ]  
}
```

API 调用

API 调用示例

```
GET /crm/provision/?  
page=1&per_page=20&sort=provision_id&order=desc  
Authorization: Bearer <token>
```

API 调用

```
GET /crm/provision/?filters={"provisioning_status":  
[2]}&search=Mobile  
Authorization: Bearer <token>
```

API 调用示例 2 搜索“Mobile”

API 调用

API 调用

- 调用 **block/rescue** 接口
- 调用 **ignore_errors** 接口
- 调用 **debug** 接口
- 调用 **assert** 接口 API
- 调用接口

API 调用

- 调用 **refresh_token** 接口
- 调用 **IP** 接口 API
- 调用 **refresh_token** 接口

□□□□

- □□□□□□□□□□□□ customer_id□service_id □□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□ Provision_Event □□□□□□□□□□□□□□□□

□□

- □□□□□□□□□□□□ customer_id□product_id□service_id
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□
- **IP** □□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□

- □□□□□□□□□□□□ API □□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

- `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□□□□

□□□□□□□□

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

□□□□□□

- □□□□□□□□□□□□□□□□

- 000000000000000000000000
- 000000“00”0000“0000”

00 4 0000000000000000

00000000000000000000000000

000000000000000000

0000000000000000000000000000000000000000000000000000000

[0] [0]

00000000

- 000000000000000000
- 000000000000000000000000

000000 00000000000000“0”000000000000000000000000000000“0”0

00 5 00000000

00000000000000000000000000

000John Smith 00ID 00123 000000 - 20GB00

000 • SIM 00SIM-00123 00ICCID 008944...0001 • 000000+44 7700 900123

000 • 00000£0.00 • 00000£15.00

00000000

[00] [000000]

00**“000000”**0000000000

00 6 00000000

00000000000000000000000000

✓ 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 網路
- 雲端服務

5G 網路與雲端服務的應用

5G 網路

5G 網路的優勢與挑戰

- 高速度 - 峰值速率可達 20Gbps
- 低延遲 - 延遲可低至 1ms
- 大連接 - 每平方公里可連接 100 萬個設備
- 高可靠 - 99.999% 的可靠性

雲端服務

雲端服務的類型

1. IaaS (Infrastructure as a Service)
2. PaaS (Platform as a Service)
3. SaaS (Software as a Service)

雲端服務的安全性

雲端服務的安全性挑戰

- 數據安全 - 數據在傳輸和存儲過程中被竊取
- 身份驗證 - 用戶身份被冒用
- 服務中斷 - 雲端服務提供者發生故障
- 隱私 - 用戶數據被濫用

雲端服務的安全性解決方案

5G 網路與雲端服務的結合

5G 網路與雲端服務的應用場景

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>

項目名

項目番号: "001" の項目

項目

5GB の項目

2025年1月10日 から 2025年1月17日 まで £5.00

WiFi 6 の項目

2024年12月1日 から Modem-12345 まで £10.00

項目名

項目番号: "002" の項目

- 項目
- "項目" の項目
- 項目 "項目" の項目
- 項目

項目

- 項目 - 項目
- 項目 - 項目
- 項目 - 項目
- 項目 - 項目

項目

項目 CPE の項目

項目名

項目 1の項目

- 項目 "項目"
- 項目

3. 如何“**???**”
4. 如何“5GB ”
5. 如何
6. 如何

如何 2 如何 IP

1. 如何 VPN 如何 IP
2. 如何
3. 如何“如何 IP ”
4. 如何 IP
5. 如何
6. 如何 IP 如何

如何 3 如何

1. 如何
2. 如何
3. 如何“WiFi 6 ”
4. 如何
5. 如何
6. 如何 £10.00 如何

如何

“如何”

- 如何
- 如何 `service_type` `residential/business` 如何

如何

- 如何
- 如何
 - 如何
 - 如何

- 管理員用戶

管理員用戶

- 管理員用戶
- 管理員用戶
 - 管理員用戶
 - 管理員用戶
 - 管理員用戶 `inventory_items_list` 管理員

管理員用戶

- 管理員用戶
- 管理員用戶
 - 管理員用戶 `auto_renew: true`
 - 管理員用戶
 - 管理員用戶

管理員用戶

- `guide_product_lifecycle` - 管理員用戶
- `csa_add_service` - 管理員用戶
- `concepts_products_and_services` - 管理員用戶
- `administration_inventory` - 管理員用戶

□□□□

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

□□□□□□ □□□□ <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/網路VoIP
- 網路 - 網路服務
- 網路 - 網路服務
- 網路 - 網路/網路
- 網路 - 網路
- 網路 - 網路服務

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 - 網路服務

網路服務

網路

網路服務

網路服務

網路

- 網路 - 網路 SIM 網路
- 網路 - 網路DSL 網路
- **IPTV** - 網路服務
- **VoIP** - 網路服務

□□□□□

□□□□□□□□□□

□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□

□□□ □□□□□□□□□□□□□□

□□

□□□□□□□□□□□□□□□□

□□□□


- □□□□
- □□ 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** 消息 - 接收/发送消息
- 通讯录 - 联系人/群组

消息

- 消息 - 接收/发送消息
- 消息 - 接收/发送消息
- 消息 - 接收/发送消息

消息

消息

消息

消息

消息

消息

消息

消息

消息

- 消息 - 接收/发送消息
- 消息 - 接收/发送消息
- 消息 - 接收/发送消息
- 消息 - 接收/发送消息
- 消息 - 接收/发送消息
- 消息 - 接收/发送消息
- 消息 - 接收/发送消息
- 消息 - 接收/发送消息

消息

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` `API` `CRM` `API`

Response

Example response:

- Account ID (string)
- Actions

Example Action: `Acme Corp` `FixedWireless_75628fa5` `Mega` `$89.99`

Example Date: `2025-11-11 10:45`

[] []

- Example "Action" name

4. `CGRateS`

5. `CGRateS`

Properties

- `CGRateS`
- `CGRateS`
- `CGRateS`
- `CGRateS`
- `CGRateS`

Methods

- `CGRateS`
- `CGRateS`
- `CGRateS`
- `CGRateS`

Example

Code Snippet

1. `CGRateS` (Example) `CGRateS`

2. `CGRateS`

```
CGRateS
```

```
Acme Corp FixedWireless_75628fa5 2025-11-11 10:45
```

```
CGRateS
```

```
[CGRateS]
```

3. `CGRateS` `CGRateS`

4. `ActionPlan` `CGRateS`

- 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
 - 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 Ansible

CGRateS Ansible

- CGRateS
- IMSI/MSISDN
-
-

- - MaxBitrateDL/UL
- - QoS PcefPolicyName
- - MSISDN
- -
- -

- -
- -
- -
- -

CGRateS

CGRateS

CGRateS

- **CGRateS** -
- **CGRateS** -
- -

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

CGRateS 設定

設定項目

ID 設定 ATTR_ACCOUNT_{service_uuid}

設定項目 UUID 設定 Mobile_SIM_c2880638 設定項目 ID 設定 ATTR_ACCOUNT_Mobile_SIM_c2880638

設定項目

1. 設定項目
2. 設定 “CGRateS 設定” 項目
3. 設定 “設定” 項目
4. 設定項目/設定項目
5. 設定 “設定”

□□□□

- □□ - □□□□□□□□□□ *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 規則
- PcefPolicyName 規則
- 其他規則

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

□□□

- □□□□□□□□□□
- □□□□□□□□

- 測試環境
- 測試數據
- 測試時間

測試

測試環境測試數據測試時間

測試

測試環境測試數據 `UUID` `*string:~*req.Account:{service_uuid}` 測試

CGRateS API 測試

測試 CGRateS 測試環境測試數據 OmniCRM API 測試

測試 `POST /crm/ocs/proxy`

測試

- `method` - CGRateS API 測試 `APIerSv1.GetAttributeProfile`
- `params` - 測試數據
- `customer_id` - 測試 ID 測試數據
- `service_id` - 測試 ID 測試數據

測試

- `tenant` - CGRateS 測試環境

測試

```
{
  "method": "APIerSv1.GetAttributeProfile",
  "params": [{"ID": "ATTR_ACCOUNT_Mobile_SIM_c2880638"}],
  "customer_id": 123,
  "service_id": 456
}
```

00000

000000 OmniCRM 000000 `crm_config.yaml` 0000 `ocs.ocsTenant` 0000000000
CGRateS 000000000000000000000000000000

000000

00 `cgrates_api_access` 0000000000000000 `admin` 0000

0000???

0000 GET 0 CGRateS API 00000000000000000000000000000000

- 0000 API 000
- 000000
- 0000000000
- 0000000000 ID
- 0000000000
- 0000

00000 CGRateS 0000000000000000

00000 **CGRateS** 000

000 **CGRateS** 000000“000000”

- 0000 000000 `cgrates_api_access` 000
- 0000 000000000000000000000000000000

000000000000

- 0000 CGRateS 0000000000000000
- 0000
 - 0000000000 CGRateS 0000000000
 - 0000000000000000
 - 0000000000000000 API 000
 - 000000000000000000000000000000

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

□□□□□□

□□□□□□□□□□□□□□□□□□□□

□□□

□□□□□□□□□□□□□□□□

□□□□

- 100GB - 1000GB
- 100GB - 1000GB
- 100GB - 1000GB/100
- 1000GB - 1000GB

100GB

1000GB

1000GB

- 1000GB - 1000GB
- 100GB - 1000GB
- 100GB - 1000GB "12.5GB / 20GB"
- 1000GB
 - 100GB > 50%
 - 100GB 20-50%
 - 100GB < 20%
- 100GB - 1000GB

1000GB

1000GB

1000GB

100GB (GB) 1000GB (MB)

1000GB

1000GB 12.5GB / 20GB (62%)

1000GB

- 1000GB - 5GB 10GB 20GB
- 1000GB - 500GB 1TB

- 00 - 000000000000
- 000000 - 0000000000

0000

000000000000

- 0000000000
- 00000000000000
- 000000000000000000

0000

0000000000

000 00 (min)

00000

000125 min / 500 min (25%)

00000000

- 00000000
- 0000
- 00000000000000
- 000000

000

0000000000000000000000000000000000000000000000000

0000

0000000000

000 00 (msgs)

00000

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
- 1000000000000

1000

- 1000000000000
- 1000000000000000
- 1000000000

1000000000000000000

10000

1000000000000

- 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

オプション

接続オプション

- 接続速度
- 接続タイプ

その他

`dongle=true` 接続オプション

接続

接続速度 0 / 0

- 接続速度 OCS ~ CGRateS
- 接続タイプ
 - 接続速度 OCS
 - 接続 OCS API
 - 接続 UUID

接続

- 接続速度 OCS
- 接続タイプ
 - 接続速度
 - 接続速度
 - 接続 OCS API

接続

- 查詢 OCS 服務狀態
- 查詢
 - 查詢 OCS 服務
 - 查詢服務
 - 查詢 OCS 服務

查詢服務

- 查詢 服務狀態
- 查詢
 - 查詢服務狀態
 - 查詢 OCS 服務

查詢服務

- 查詢 服務狀態
- 查詢
 - 查詢服務狀態
 - 查詢服務
 - 查詢服務

■ OCS/CGRateS 查詢

查詢 OCS 服務狀態 CGRateS 查詢服務狀態
 查詢 [CGRateS Actions and Topup Behaviors](#)

查詢

1. 查詢服務
2. OmniCRM 查詢 `GET /crm/service/{service_id}`
3. API 查詢 UUID 查詢 OCS
4. OCS 查詢服務

```
{
  "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 0000000000000000 → GB0000 → 0000

6. 0000000000

7. 0 3 000000

OCS 00000000

OCS 資料取得のUI

- `*data` → 資料取得
- `*voice` → 音声取得
- `*sms` → SMS取得
- `*monetary` → 課金取得

資料取得 + 課金 + 課金履歴取得

単位

- 1GB → 1024^3
- 1分 → 60×10^9
- 1円
- 100円

課金履歴

OCS 課金履歴取得のAPIは [CGRateS Actions and Topup Behaviors](#)

課金履歴

CGRateS 課金履歴取得のAPI

- 課金履歴取得
- 課金履歴
- 課金履歴
- 課金履歴

課金履歴

1. 課金履歴

- 課金履歴 `auto_renew = true`
- 課金履歴 OCS 課金履歴
- 課金履歴取得のAPI

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
- 失败 — 失败返回 JSON 格式
 - `impersonating_user_id` — 冒充者 ID
 - `target_user_id` — 目标用户 ID
 - `impersonation_start` — 开始时间
- 消息 — UI 显示的消息
 - "成功 [消息] 完成"
 - "失败 [消息]"
- 日志 — 记录冒充操作 ID
 - 记录
 - 记录
 - 在 `ImpersonationLog` 表中记录

🔐🔐🔐🔐

🔐🔐 Web UI 🔐

1. 🔐🔐🔐 — 🔐🔐🔐🔐🔐🔐🔐🔐
2. 🔐🔐 — 🔐🔐🔐🔐🔐🔐🔐🔐“🔐🔐🔐”
3. 🔐🔐 — 🔐🔐🔐🔐🔐🔐🔐🔐🔐🔐

🔐🔐 API 🔐

🔐🔐🔐 `POST /auth/stop_impersonation`

🔐🔐🔐 🔐🔐🔐🔐🔐🔐🔐🔐🔐🔐🔐

🔐🔐🔐

```
{
  "success": true,
  "impersonation_end": "2025-01-04T15:45:00Z",
  "duration_seconds": 900,
  "access_token": "your_original_token",
  "refresh_token": "your_original_refresh_token"
}
```

🔐🔐🔐🔐🔐🔐🔐

🔐🔐🔐🔐🔐🔐

🔐🔐🔐🔐🔐🔐 `ImpersonationLog` 🔐🔐🔐🔐

- **impersonating_user_id** — 🔐🔐🔐🔐🔐🔐
- **target_user_id** — 🔐🔐🔐🔐🔐🔐
- **impersonation_start** — 🔐🔐🔐🔐
- **impersonation_end** — 🔐🔐🔐🔐🔐🔐🔐🔐
- **impersonation_duration** — 🔐🔐🔐🔐🔐🔐🔐

🔐🔐🔐🔐🔐🔐🔐🔐🔐🔐🔐🔐

- `impersonation_logs` — `impersonation_logs`
- `impersonation_logs` — `impersonation_logs`
- `impersonation_logs` — `impersonation_logs`
- `impersonation_logs` — `impersonation_logs`

API

API `impersonation_logs`

```
GET /auth/impersonation_logs?user_id={user_id}&start_date={date}&end_date={date}
```

Response

- `impersonation_logs`
- `impersonation_logs`
- `impersonation_logs`
- `impersonation_logs`

Fields

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`

Fields

- `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 Plays 📄 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 Successful")
```

Part 2 API Data Retrieval

Retrieve data from the CRM API using the token from the previous step.

```

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
- `BalanceId` `BalanceType` `DestinationIDs` `Units` `ExpiryTime` `Weight`
-
-

□□□□□□

- ***topup** - □□□□□□□□□□□□□□□□□□□□□□□□
- ***topup_reset** - □□□□□□□□□□□□□□□□□□□□□□□□

□□ **6□□ CRM** □□□□□

□□CGRateS □□□□□□□□ CRM □□□□□□□□

□□□□ **SIM** □□

```
product_list_new = []

product_list_new.append({
    "category": "standalone",
    "provisioning_play": "play_psim_only",
    "relies_on_list": "",
    "contract_days": 0,
    "retail_cost": 0,
    "retail_setup_cost": 0,
    "product_slug": "Mobile-SIM",
    "product_name": "□□□□ SIM",
    "comment": "□□□□□□□□□□ SIM □",
    "provisioning_json_vars": "{\"iccid\" : \"\", \"msisdn\" : \"\"}",
    "terms": "□□□□ 6 □□□□□□□□□□□□ 12 □□□□□□□□□□□□□□□□□□",
    "service_type": "mobile",
    "residential": True,
    "business": True,
    "enabled": True,
    "inventory_items_list": "['SIM Card', 'Mobile Number']",
    "icon": "fa-solid fa-sim-card",
    "features_list": "['□□□□□□□□ (61xxx)', '□□□□□□', '□□□□□□']",
    "wholesale_cost": 3,
    "wholesale_setup_cost": 1,
    "customer_can_purchase": True
})
```

□□□□□□□□□□

```

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", # Action_au-premium-
plan-1
    "product_name": "Premium Plan 1",
    "comment": "100GB 3000 SMS 6GB",
    "provisioning_json_vars": "",
    "terms": "30",
    "service_type": "mobile",
    "residential": True,
    "business": True,
    "enabled": True,
    "icon": "fa-solid fa-mobile",
    "features_list": ["100GB", "3000 SMS", "6GB"],
    "wholesale_cost": 45.00,
    "wholesale_setup_cost": 0,
    "auto_renew": "True",
    "customer_can_purchase": True
})

```

100GB 3000 SMS 6GB

```

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", # CGRateS Action_au-data-
addon-20gb
    "product_name": "20GB",
    "comment": "20GB",
    "provisioning_json_vars": "",
    "terms": "",
    "service_type": "mobile",
    "residential": True,
    "business": True,
    "enabled": True,
    "icon": "fa-solid fa-mobile",
    "features_list": "['20GB']",
    "wholesale_cost": 22,
    "wholesale_setup_cost": 0,
    "auto_renew": "False",
    "customer_can_purchase": True
})

```

CGRateS

- **category** -
 - standalone -
 - addon -
 - bundle -
- **provisioning_play** - Ansible playbook
 - play_topup_charge_then_action
 - play_simple_service
- **relies_on_list** - product_slugs
- **contract_days** - 0 = 30 =
- **retail_cost** - /
- **retail_setup_cost** -
- **product_slug** - CGRateS Action_{product_slug}
- **product_name** -

- **comment** - 字符串
- **provisioning_json_vars** - 在 playbook 中定义的 JSON 变量
- **terms** - 字符串
- **service_type** - 字符串
- **residential** - 字符串
- **business** - 字符串
- **enabled** - 字符串
- **inventory_items_list** - 在 Python 中定义的列表
- **icon** - Font Awesome 图标
- **features_list** - 在 Python 中定义的列表
- **wholesale_cost** - 字符串
- **wholesale_setup_cost** - 字符串
- **auto_renew** - 字符串 "True" 或 "False"
- **customer_can_purchase** - 字符串

字符串

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}")
```



```
# play_topup_charge_then_action.yaml
- name: cgr_action_name
  set_fact:
    cgr_action_name: "Action_{{
api_response_product.json.product_slug }}"

# CGRateS
- 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 "CGRateS" CGRateS

Ansible Playbooks

CGRateS

"50GB CGRateS"

1. CGRateS

CGRateS 5 CGRateS 50GB CGRateS

- **ActionsId:** "Action_50gb-data-pack" CGRateS product_slug
- **BalanceType:** "*data"
- **DestinationIDs:** "Dest_PLMN_OnNet" CGRateS
- **Units:** 50 * 1024 * 1024 * 1024 50GB CGRateS

- Identifier: `*topup` `XXXXXXXX`

2. CRM `XXXXXXXX`

```
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 XXXXXXXX
    "Action_"
    "product_name": "50GB XXXX",
    "comment": "XXXX 30 XX 50GB XXXX",
    "provisioning_json_vars": "",
    "terms": "XXXX 30 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
    "service_type": "mobile",
    "residential": True,
    "business": True,
    "enabled": True,
    "icon": "fa-solid fa-database",
    "features_list": "['50GB XXXX', '30 XXXX', 'XXXX']",
    "wholesale_cost": 20.00,
    "wholesale_setup_cost": 0,
    "auto_renew": "False",
    "customer_can_purchase": True
}

# CRM XXXX
response = session.put(
    crm_url + '/crm/product',
    json=product_50gb_data,
    headers=headers
)
assert response.status_code == 200
print("XXXXXXXX50gb-data-pack")
```

3. `XXXX`

`XXXXXXXXXXXX`

1. 创建 "50GB 数据" 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 接口 ID 为 Action_50gb-data-pack
5. CGRateS 配置
6. 创建 50GB 数据

创建 Playbooks

1

CRM 配置 CGRateS 配置

1

SMS

CRM

```
{
  "product_slug": "premium-monthly-plan", # 调用 Action_premium-monthly-plan
  "product_name": "数据",
  "category": "addon",
  "retail_cost": 59.00,
  "auto_renew": "True",
  "contract_days": 30,
  "provisioning_play": "play_topup_charge_then_action",
  "features_list": "['100GB 数据', '5000 数据', '5000 SMS', '数据']"
}
```

*topup_reset 配置 Python

2

SMS

CRM 配置

```
{
  "product_slug": "10gb-addon", # 配置 Action_10gb-addon
  "product_name": "10GB 配置",
  "category": "addon",
  "retail_cost": 15.00,
  "auto_renew": "False", # 配置
  "contract_days": 0,
  "provisioning_play": "play_topup_charge_then_action",
  "features_list": "['10GB 配置', '30 配置']"
}
```

配置 *topup 配置 CGRateS 配置 - 配置

配置 3配置

配置 配置

CRM 配置

```
{
  "product_slug": "fixed-50gb-monthly", # 配置 Action_fixed-50gb-monthly
  "product_name": "配置 50GB 配置",
  "category": "addon",
  "retail_cost": 35.00,
  "auto_renew": "True",
  "contract_days": 30,
  "provisioning_play": "play_topup_charge_then_action",
  "features_list": "['50GB 配置', '配置', '配置']"
}
```

配置 *topup_reset 配置 CGRateS 配置 - 配置

配置

配置 配置CGRateS 配置

- 2G/3G 网络支持 MO 及 MT 短信/SMS 业务
- 2G/3G 网络支持 CAMEL 及 MO-MT SMS 及 USSD 业务
- 4G/5G 网络支持 Diameter 业务
- 支持 3GPP 定义的鉴权及计费功能
- 支持 3GPP 定义的计费功能

支持 OmniCRM 及 PAYG 业务

支持 3GPP 定义的鉴权及计费功能

支持 3GPP

支持 SMS 业务

1. CGRateS 支持 GB
2. 支持鉴权及计费功能
3. 支持 CGRateS 鉴权及计费功能
4. 支持鉴权及计费功能 \$0.10 / MB
5. 支持鉴权及计费功能

□□□□□□□□□□□□□□□□

```
# 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，Subject必須設定為 **\$0.10** 或 **MB**
- 設定APIOptions時，APIOpts必須設定為 `*monetary`
- 設定APIOptions時，APIOpts必須設定為 `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

```
# 100GB - 單位
Units = 100 * 1024 * 1024 * 1024 # 100GB 單位
Units = 3000 * 60 * 1000000000 # 3000 單位

# 107374182400
Units = 107374182400
```

6. 單位

單位 = 單位

```

# 重置账户权重 = 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. 计费策略配置

计费策略配置

```

# 100MB
Units = 100 * 1024 * 1024 # 100MB 策略

# 100GB
Units = 100 * 1024 * 1024 * 1024 # 100GB

```

策略

策略“策略名称”

策略 Playbook 策略名称 “策略名称” 或 “ActionId 策略”

策略 策略 CGRateS 策略名称

策略

1. 策略名称 CGRateS 策略名称
2. 策略名称 `ActionsId = "Action_" + product_slug`
3. 策略名称

□□□□“□□□□□□□□”

□□□□ □□□□□□□□□□□□

□□□□ DestinationIDs □□□□□□□□□□

□□□□□□

1. □□□□ CGRateS □□□□□□□□
2. □□ *any □□□□□□□□
3. □□□□□□ ID □□□□

□□□□□□□□□□□□□□

□□□□ □□□□□□□□□□□□□□

□□□□ □□□□□□□□□□□□□□

□□□□□□

1. □□□□□□□□□□ *topup □□ *topup_reset □□
2. □□□□□□□□□□□□□□ 0□
3. □□□□□□□□□□□□

□□□□□□□□

□□□□ □□□□□□□□□□□□□□

□□□□ □□□□ *topup_reset □□□□ *topup

□□□□□□

- □□ Identifier □□ *topup_reset □□□□ *topup
- □□ Balancelid □□❖❖□□□□□□□□

□□□□□□ CGRateS □□□□□□□□

□□

□□□□□□ OmniCRM □□□□□□ CGRateS □□□□□□□□□□□□□□□□□□□□□□□□□□
□MVNO□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□ CGRateS □□□□□□□□□□□□□□□□□□□□□□□□□□

- □□□□□ SMS □□□□ MO/MT □□
- □□ 2G/3G □□□□□□□ CAMEL □□□□
- □□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□ QoS □□□□
- □□□□□□□□□□□□□

...□□□□□□ CGRateS □□□□□□□□ CRM □□□□□□□□□□□□□□

□□□□

- CGRateS □□□□□□□□ - □□□□□□□□□□□□□□□□□□□
- Ansible Playbooks □□ - Playbooks □□□□□□□□
- □ Playbooks □□□□□□□□ - □□□□□□□□□□
- □□□□□□□□ - □□□□□□□□□□□□
- □□□□□□□□□□ - □□□□□□□□
- CGRateS □□□□□ - □□□□ CGRateS □□□□



OmniCRM 系統與 AMBER 系統

系統 SMS 系統

OmniTouch 系統與 OWL 系統

OmniTouch 系統與 OWL 系統

- **CBE** - Web 系統
- **CBC** - 2G/3G/4G/5G 系統

OWL 系統 Web 系統 Chrome/Firefox/Safari/Edge 系統



系統 PWS 系統

- 系統
- 系統
- 系統 500 系統
- 系統
- 系統 CBC 系統
- 系統 TOTP 系統
- 系統

系統 SMS 系統 10 系統

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

???

-
-

- 認證 - 認證碼
- 認證 - 認證碼
- 認證 - 認證碼

認證

認證CBC認證碼

- 認證 - ISO認證碼en es fr zh
- 認證 - 認證碼1395

認證碼

認證

認證碼

- 認證 - 認證碼ID
- 認證 - 認證碼MCC-MNC
- **RAT** - 認證碼LTE 5G UMTS GSM

認證碼

認證碼

認證Web UI

1. 認證碼 - 認證碼CBC
2. 認證“認證碼” - 認證碼
3. 認證碼
 - 認證碼4370
 - 認證碼
 - 認證碼5-60
 - 認證碼999
 - 認證碼
 - 認證碼“認證碼”

4. 网络制式

- 网络“制式”
- 网络制式
- 网络制式 GSM7 和 1395 网络制式 Unicode 网络
- 网络制式

5. 网络制式

- 网络“制式”
- 网络制式
- 网络制式 MCC-MNC 网络
- 网络 RAT 网络 LTE 5G 网络
- 网络制式

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 MNO CBC

OWL CBC

CBC URL `crm_config.yaml`

```
cbc_url: "http://cbc.example.com:8080"
```

Steps

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

Steps

OmniCRM CBE CBC

Steps

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. OWL
- 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`

`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□□□

```
VITE_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
```

Map Data

- Map Data
- Map Data
- GPS Data
- RNC Data

Map Data

mapStyles.js

- Map Data
- Map Data
- Map Data
- Map Data

Map Data

Map Data

- Map Data
- Map Data
- Map Data
- Map Data

Map Data

- Map Data
- Map Data
- Map Data
- Map Data

Map Data

- 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(大陸)

繁體中文(台灣)

- 繁體中文(台灣) 360 繁體中文(台灣)
- 繁體中文(台灣) 繁體中文(台灣)
- 繁體中文(台灣) “繁體中文” “繁體中文”
- 繁體中文(台灣)
- 繁體中文(台灣) 繁體中文(台灣)

繁體中文(大陸)

- 繁體中文(大陸) 繁體中文(大陸)
- 繁體中文(大陸) 繁體中文(大陸)
- 繁體中文(大陸) 繁體中文(大陸)
- 繁體中文(大陸) Unicode 繁體中文(大陸)

繁體中文(香港)

- 繁體中文(香港) 繁體中文(香港)
- 繁體中文(香港) 繁體中文(香港)
- 繁體中文(香港) 繁體中文(香港)
- 繁體中文(香港) 繁體中文(香港)

繁體中文(澳門)

- 繁體中文(澳門) 4379 4380-4381 繁體中文(澳門)
- 繁體中文(澳門)
- 繁體中文(澳門) CBC 繁體中文(澳門)

- 2024년 1월 1일부터
- 2024년 1월 1일부터

2024년 1월 1일부터 Omnitouch 2024년 1월 1일부터

2024년

- 2024년 1월 1일부터
- 2024년 1월 1일부터
- MNO 2024년 1월 1일부터
- 2024년 1월 1일부터

2024년

2024년 Omnitouch 2024년 1월 1일부터

2024년

2024년 1월 1일부터

- 2024년 **K8s** - Kubernetes 2024년 1월 1일부터
- 2024년 - VMware Proxmox HyperV
- 2024년 - 2024년 1월 1일부터
- 2024년 - AWS GCP
- 2024년 - 2024년 1월 1일부터

2024년

2024년 1월 1일부터

- 2024년 1월 1일부터 OWL CBE 2024년 1월 1일부터
- 2024년/2024년 1월 1일부터
- 2024년 1월 1일부터 NDMO 2024년 1월 1일부터

2024년 MNO 2024년 1월 1일부터 BSC/MME 2024년 1월 1일부터

CBE CBC 2024년

Windows Server 2012 R2 MNO CBC

CBE CBC

- CBE CBC
- Windows Server 2012 R2
- CBE CBC TLS TCP 443
- MNO NDMO CBE

Windows

CBE Citrix VPN

- Windows Server 2012 R2
- Citrix
- Windows Server 2012 R2

Windows

CBE NDMO / Windows

3 Windows

- **2 CBE**
- **1 Windows**

Windows

- Windows 50GB
- **CPU** 2 CPU
- **RAM** 8GB
- Windows Omnitouch
- Windows TCP 443 CBC TLS

CBC MNO

3

- 2 CBC
- 1

- 50GB
- CPU 2 CPU
- RAM 8GB
- Omnitouch
- MME RNC BSC
- CBE TCP 443 TLS

OWL

1. -
2. -
3. CBE - CBE
4. / -
5. -
6. IP - NDMO MNO IP
7. VPN - Omnitouch
8. CBE NDMO -
9. CBC MNO -
10. - CBC
11. CBE CBC -
12. CBC - BSC/MME/AMF
13. API NMS -
14. -
15. / -

16. □□□□□ - □□□□□□□□□□

□□□□

□□□□□□□□□□□□□□□□□□

GSM 7□□□□□□□□□□□□□□□□

- □□□93□□□
- □□□15□ × 93 = □□1395□□□

Unicode UCS-2□□□□□□□□□□□□□□□□

- □□□41□□□
- □□□15□ × 41 = □□615□□□

OWL□□□

- □□□□□□□500□□□
- Web UI□□□□□□□□□□□□□□□□□□

□□□□□

□□□□□□□□□□□□□□

- □□□□□□□□□□□□□□
- □□□□□□□
- □□□□□□□□□
- CBC API□□□□□□
- □□□□□□□□□



□□□□□□□□□□□□□□□□

```
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□□□□□□□□□□

- 4379
- 4380
- 4381
- 4382

4379-4382

4379

4380

4381-4382

- 4379
- 4380
- 4381
- 4382

4383

- 4379-4381
- 4382
- 4383
- 4384

4385

4386-4387 / Omnitouch 4388-4389

4390

OWL 4391

- NetAct
- U2000 / U2020
- NetNumen / ZXPOS

- ENM

OmniTouch

NDMO

GIS Google Earth OmniTouch

NDMO

-
-
-
-

OmniTouch

-
-
-

- 2FA 的優點

2FA / 雙重認證

OmniTouch 雙重認證

- 2FA 的優點
- 簡便易用/快速/無須安裝
- 安全

優點

雙重認證可防止未經授權的用戶訪問您的系統，即使您的密碼被洩露，您的系統也是安全的。

優點

1. 簡便易用
2. 快速
3. 無須安裝
4. 安全

API 安全

API 安全是保護您的系統免受未經授權的訪問，CBE 提供 API 安全

API 安全

- 身份驗證 - 確保只有授權的用戶可以訪問您的系統
- 授權 - 確保用戶只能訪問他們應該訪問的資源
- 數據加密 - 確保您的數據在傳輸過程中是安全的
- 安全更新 - 確保您的系統是最新的，並受到保護

API 安全是保護您的系統免受未經授權的訪問

API 安全

CBC 提供 API 安全

□□□□

PUT /crm/cbc/

□□□□□□

GET /crm/cbc/

□□□□

PATCH /crm/cbc/{cbc_message_id}

□□□□

DELETE /crm/cbc/{cbc_message_id}

□□□ /crm/docs/ □□Swagger□□□□□□□□API□□□

□□□□

□□□□

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** - □□□□□□□□□□
- □□ - □□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□

□□□□□□

□□ **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** □□□□

□□□□□□□□ □□□□□□□□□□□□□□

| □□□ • 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

GET 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 0000000000000000000000000000000000000000

00000000

- 000 000“00000000”000
- 00000 0000000000

00000

- basics_customers - 0000
- basics_navigation - 0000
- administration_inventory - 0000

OmniCRM Self-Care Portal

OmniCRM <self_care_portal> Self-Care Portal

- SIM Card Provisioning
- WiFi Dongle Provisioning
- Device Provisioning

SIM Card Provisioning <concepts_sim_provisioning> SIM Card Provisioning

Stripe

Stripe

Stripe

- Stripe
- Stripe 1-30
- Stripe
- Stripe
- Stripe
- Stripe
- Stripe

CRM

CRM

CRM

- CRM SMS
- CRM

- 0000000000
- 000000

0000000000 IP 000 IMSI 0000000000

0000

00000 **4** 000 000

00 **1**000000

0000000000000000

000

- 0000 - 00 1 0 30 0
- 000000 - 0000000000
- 000000 - 000000000000
- 00000000 - 0000000000/00

00000

00000

- 000000000000 `VITE_TOPUP_PRICE_PER_DAY` 00
- 000000 \$10 00
- 0000 `VITE_CURRENCY_CODE` 00

00 **2**000000

000000000000

- 00
- 00
- 000000

00000000

- 信用卡
- 移动支付
- 虚拟货币
- 数字货币

第 3 章

本章将介绍 Stripe Elements、PayPal SDK 等支付网关

支付网关

- 支持 Visa、Mastercard、Amex
- 支持
- 支持 Apple Pay、Google Pay、PayPal 等支付方式

支付网关

- 支持 PCI 支付标准
- OmniCRM 支付网关
- 支持 3D Secure 支付
- 支持

支付网关

1. 支付网关
2. 支付网关
3. 支付网关
4. 支付网关
5. 支付网关/支付

支付

支付网关 OCS 支付网关

第 4 章

支付网关

2025年1月17日

客户 <customer@example.com> ID: TXN-123456

摘要

服务订购成功

服务名称: 移动热点

订购日期: 2025-01-17

交易ID: TXN-123456

详情

服务订购成功

1. 成功

响应:

- 状态: `succeeded`
- 金额: `days × price_per_day`
- 交易ID: TXN-123456

2. 配置

- API 端点: `POST /oam/topup_dongle`
- 参数: `service_uuid` 和 `IMSI`
- 参数: `OCS/CGRateS`
- 参数: `play_topup_hotspot`

3. 备注

服务订购成功

- **HotspotTopup** 参数 - 支付意图ID
 - `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 - 数据
 - 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 playbook 数据

```
- name: OCS
  hosts: localhost
  tasks:
    - name: OCS
      uri:
        url: "{{ ocs_api_url }}/add_balance"
        method: POST
        body:
          imsi: "{{ imsi }}"
          days: "{{ days }}"
          service_uuid: "{{ service_uuid }}"
```

000000

000000000000000000000000

SMS

0 OCS 000000 Action_Balance_Low Action_Balance_Out
Action_Balance_Expired

000000

0 OCS/CGRateS 00000000000000000000

000000

- Action_Balance_Low - 000000000000 2
- Action_Balance_Out - 0000
- Action_Balance_Expired - 0000

000000000000000000000000

□□□□

□□□□

“□□□□□□”

- □□□ Stripe □□□□□□□□□□□□□□
- □□□□□
 - □□ VITE_STRIPE_PUBLISHABLE_KEY □□□□□□
 - □□ Stripe □□□□□□□□□□
 - □□□□□□□□□□ JavaScript □□

“□□□□□□□□...”

- □□□ □□□□□□□ OCS □□□□□□□□□□□□□
- □□□□□
 - □□□□□□□ GET /crm/provision/provision_id/<id>
 - □□ OCS/CGRateS API □□□□□□
 - □□ playbook play_topup_hotspot.yaml □□□□□□
 - □□ Ansible □□

“□□□□□□”

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

“□□□□□□□□”

- □□□ UI □□□□□□□□□□□□□□
- □□□□□
 - □□ VITE_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 Level 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 **playbooks** 查詢 - 查詢 playbooks 查詢
4. 查詢 **OCS** 查詢 - 查詢 OCS API 查詢
5. 查詢 - 查詢 `VITE_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

- `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` - 产品 UI 图标 FontAwesome 图标
- `comment` - 产品备注

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

- 設定する
- 設定する
- 2025年1月1日
- 設定する

5 設定

```
{
  "auto_renew": "prompt",
  "allow_auto_renew": true
}
```

- "prompt" - 設定する
- "true" - 設定する
- "false" - 設定する
- allow_auto_renew: true - 設定する/設定する

6 設定

設定する `<administration_inventory>` 設定する

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

設定

設定する OmniCRM 設定する

- SIM - 設定する
- MAC - 設定する
- 設定する
- 設定する

設定

inventory_items_list Python `<administration_inventory>`

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

# Modem Router
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"
    }

```

- “”
-
-

○○○○○○○○

○○○“○○”○○○○○○○○

1. ○○○ ID○○○○○○○○○○○ service_id ○○○
2. ○○○○○○○○○○○○○○○○○○OCS ○○○○○○○○○○○○○○○
3. ○○○○○○○○○○○○○○○○○○○○○

○○○○○○

○○○○○○○○○○○○○○○○○○○○

```

rescue:
  - name: ○○○○○
    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 # ○○○○○

```

○○○○○○○○○○○○○○○○○○○○“○○”○○

○○○○○○

○○ inventory_items_list: "[]" ○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○

- ○○○ - ○○○○○VPN ○
- ○○○○○ - ○○○○○○○○○○○○○
- ○○○ - ○○○○○○○

○○ “5GB ○○○”○○○○○○ inventory_items_list: "[]" ○○○○○○○○○○○○○○○○○○○○○○○
○○

□□□□□□

□ `inventory_items_list` □□□□□□□□□□□□□□□□□□□□

1. □□□ □□ → □□ → □□
2. □□□□□□□□□□□□□□□□ "SIM □"□
3. □□□□□□
 - `itemtext1_label`: "ICCID"
 - `itemtext2_label`: "IMSI"
 - `itemtext3_label`: "PUK Code"
4. □□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□ □□□□ `<administration_inventory>`□

□□□□□□□□□□

□□ `inventory_items_list` □□□□□□□□□□□□□□□□□□□□ `"['SIM Card', 'SIM Card']"`□□
□□□□□□□□□□□□ UI □□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□

□□ **1**□□□□□□□□□□□□□□□□

```
# □ SIM □□□□  
inventory_items_list: ["'Primary SIM Card', 'Secondary SIM Card',  
                        'Mobile Number']"
```

□□□□□□□□□□ "Primary SIM Card"□□ "Secondary SIM Card"□□□□□□□□□□□□

□□ **2**□□□□□□□□□□□□□□□□

```
# □ SIM □□  
inventory_items_list: ["'Dual SIM Kit', 'Mobile Number']"
```

□□ "Dual SIM Kit"□□□□□□□□□□ SIM □□□□□ `itemtext1`: □ ICCID□ `itemtext2`: □ ICCID□
□□□□

□□□□□□

VoIP

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

- SIM Card ICCID/IMSI eSIM
- MSISDN

Modem Router

```
inventory_items_list: ["Modem Router', 'Static IP Address']"
```

- MAC CPE
- IP IPv4

Fixed Wireless CPE

```
inventory_items_list: ["Fixed Wireless CPE', 'IPv4 Address', 'IPv6 Prefix']"
```

- CPE
- IPv4 IP
- IPv6 /56 /64

VoIP SIP

VoIP

```
inventory_items_list: ["DID Number']"
```

- DID

SIP

GPON

```
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
- 設備名稱設備名稱設備名稱設備名稱

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
- 設備名稱設備名稱設備名稱

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
- 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. service_id 为 NULL 的
3. service_id 为 NULL 的 → 查询 → 插入
4. service_id 为 NULL 的“”“”
5. service_id 为 NULL 的

SQL 语句

SQL

- inventory_items_list 为 []
- inventory_items_list 为 NULL
- service_id 为 NULL

SQL

- inventory_items_list: ["Type1", "Type2"]
- service_id
- API 返回 inventory_items_list

SQL 语句“”

SQL

- service_id
- ID
- service_id

SQL

- hostvars[inventory_hostname]['SIM Card']
- {{ hostvars[inventory_hostname]['SIM Card'] | int }}
- service_id

SQL 语句 <administration_inventory>

SQL 7

SQL 语句

```
{
  "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/属性/卖点

1. 标题 - 关键词
 - o “20GB”“30GB” - 关键词
 - o 卖点/属性
2. 标题 - 关键词
 - o “30天”关键词
 - o “30天”关键词
 - o 卖点/属性
3. 标题 - 关键词
 - o 关键词“30天”→关键词❓❓❓
 - o 关键词
 - o 关键词
4. 标题 - 关键词
 - o 关键词
 - o 关键词
 - o 关键词
5. **SEO** 关键词 - 关键词
 - o 关键词“30天”→关键词
 - o 关键词

标题 - 关键词

描述/属性/卖点

1. 标题 - 关键词
 - o “30天” - 关键词 31 天
 - o “30天” - 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. 0000000000

0000000000000000000000000000

0000000000

terms 00000000000000000000

000 **UI** 00000000

1. 00000000

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 20GB",
  "product_slug": "prepaid-mobile-20gb",
  "category": "standalone",
  "service_type": "mobile",
  "enabled": true,
  "icon": "fa-solid fa-sim-card",
  "comment": "SIM 20GB",

  "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',
    '',
    '',
    '',
    '30'
  ],
  "terms": "30",

  "provisioning_play": "play_local_mobile_sim",
  "provisioning_json_vars": "{
    \"days\": 30,
    \"data_gb\": 20,
    \"voice_minutes\": \"unlimited\",
    \"sms_count\": \"unlimited\"
  }
```



```
{
  "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 - 0000000000000000000000000000
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 days

3. Item state

- `item_state = "In Stock"`
- `item_state = "Damaged"`
- ...

4. Price

- 14 days £150
- `item_state = "Lost"`
- £45+ ...

...

...

- ...
- ...
- `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

- Price £0.00
- Price £15.00
- Price £15.00
- Duration 30

Implementation

30

4. API

API

- API "status=1" Provision
- + +
- 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 = IMSI000000
- 000000000000000000000000

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] }}"

```

0000000

- 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

```

•••••

- CGRateS JSON-RPC API
- service_uuid
-
-

••••

- 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“移动”
- 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"
    }

```

•

- SIM
- service_id SIM
- “”“”
-

•

- 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"

```

套餐

- 套餐管理
- 删除OCS 套餐
- 套餐“名称”
- 套餐管理

套餐

- OCS 套餐
- 套餐
- 套餐

套餐管理

套餐管理

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

- 20GB - 20GB

20GB

- 2GB 25
- 5GB 7
- 7GB
- 20GB

4GB

```
- name: 20GB
  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
      }
}
```

20GB

- £5.00
- £1.50
- 20GB

20GB

1. 20GB
2. 20GB
3. 20GB OCS

4. CRM 設定

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”
- []“[]”[]
- []“[]”[]

第 4 章

本章主要介绍了...

学习目标

本章学习结束后，应能...

1. 了解...
2. 掌握...
3. 理解...
4. 能够...

本章重点

1. 重点一

- 概念
- 定义
- 性质
- 应用

2. 重点二

本章重点二...

- 概念
- 定义
- 性质
- 应用

3. 重点三

本章重点三...

```
{
  "action": "deprovision",
  "service_id": 1234,
  "customer_id": 123,
  "service_uuid": "Local_Mobile_SIM_a3f2c1d8"
}
```

[[service.deprovisioning_play]]

4. Ansible []

```

- name: 0000000000
  hosts: localhost
  tasks:
    - name: 00 OCS 00
      uri:
        url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
        method: POST
        body:
          {
            "method": "ApierV2.SetAccount",
            "params": [{
              "Account": "{{ service_uuid }}",
              "ExtraOptions": { "Disabled": true }
            }]
          }

    - name: 00000000000000000000
      uri:
        url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
        method: POST
        body:
          {
            "method": "ApierV1.RemoveActionPlan",
            "params": [{
              "Id": "ServiceID_{{ service_uuid }}__*"
            }]
          }

    - name: 00 CRM 00000000
      uri:
        url: "http://localhost:5000/crm/service/{{ service_id }}"
        method: PATCH
        body:
          {
            "service_status": "Deactivated",
            "service_deactivate_date": "{{ current_datetime }}"
          }

    - name: 0000000000
      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

- 000000
- 0000 OCS 00
- 0000
- 000000

```
- name: 00 OCS 00
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body:
      {
        "method": "ApierV2.RemoveAccount",
        "params": [{
          "Account": "{{ service_uuid }}"
        }]
      }
}
```

00000000000000

0000000000

00000

- `enabled: true` - 開
- `enabled: false` - 閉

設定

- `enabled: false` 設定
- 設定
- 設定/設定
- 設定

設定

設定

- `New` - 設定
- `In Stock` - 設定
- `Assigned` - 設定
- `Decommissioned` - 設定
- `Damaged` - 設定

設定

設定

- SIM 設定“設定”
- 設定30 設定
- 設定“設定”

設定

設定

- 設定
- 設定
- 設定
- 設定

Mailjet & OmniCRM

OmniCRM & Mailjet integration

Overview

Mailjet integration

- Integration -
- Integration - /
- Integration - 10 Mailjet
- Integration - CRM CSV
- Integration -

OmniCRM & Mailjet integration **Customers → All Contacts** UI `GET /crm/contact/export`

Configuration

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

Example

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

API

1. <https://www.mailjet.com>
2. → API
3. API

4. `crm_config.yaml`

Configuration

OmniCRM `10` configuration parameters

- **from_email** - email address
- **from_name** - sender name
- **template_id** - Mailjet ID
- **subject** - email subject

Example

Configuration

Example

```
api_crmCommunicationCustomerWelcome:  
  from_email: "support@yourcompany.com"  
  from_name: "Your Company Support"  
  template_id: 5977509  
  subject: "Welcome to our company"
```

Parameters

- `from_email`
- `from_name`
- `template_id`

Variables

- `{{ var:customer_name }}` - customer name
- `{{ var:email }}` - customer email
- `{{ var:company_name }}` - company name
- `{{ var:login_url }}` - login URL
- `{{ var:support_url }}` - support URL

□□□□□□□□

□□□□□□□□□□□□□□

```
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 }}` - 用户名
- `{{ var:email }}` - 邮箱地址
- `{{ var:role }}` - 角色
- `{{ var:login_url }}` - 登录 URL
- `{{ var:temp_password }}` - 临时密码
- `{{ var:support_email }}` - IT 支持邮箱

模板内容

模板配置

```
api_crmCommunicationUserPasswordReset:
  from_email: "noreply@yourcompany.com"
  from_name: "Your Company Security"
  template_id: 5977511
  subject: "密码重置"
```

模板变量

- 模板内容中的“密码重置”
- 模板配置

模板内容

- `{{ var:user_name }}`
- `{{ var:reset_url }}` - 重置 URL 1 分钟
- `{{ var:expiry_time }}` - 有效期

模板配置

模板配置

模板内容

模板配置

```
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のお知らせ"
```

本文

- パスワードリセット完了のお知らせ
- パスワードリセット完了のお知らせ

本文

- {{ var:user_name }}
- {{ var:change_date }} - 年/月/日
- {{ var:ip_address }} - IP アドレス
- {{ var:support_email }} - サポートメール

本文

本文

```
api_crmCommunicationUserPasswordChange:
  from_email: "noreply@yourcompany.com"
  from_name: "Your Company Security"
  template_id: 5977513
  subject: "パスワード変更完了のお知らせ"
```

本文

- パスワード変更完了のお知らせ
- パスワード変更完了のお知らせ

本文

- {{ var:user_name }}
- {{ var:change_date }}
- {{ var:changed_by }} - “”

- `{{ var:support_email }}`

□□□□□□

□□□□□□□□◆◆□□□□□□

```
api_crmCommunicationEmailVerification:  
  from_email: "noreply@yourcompany.com"  
  from_name: "□□□□"  
  template_id: 5977514  
  subject: "□□□□□□□□□□"
```

□□□□□□

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

□□□□□□□□

- `{{ var:user_name }}`
- `{{ var:verification_url }}` - □□□□□□□□□□
- `{{ var:verification_code }}` - □□□□□□□□□□□□□□

□□□□□□□□

□□□□□□□□□□/□□□□□□□□□□

```
api_crmCommunicationsBalanceExpired:  
  from_email: "support@yourcompany.com"  
  from_name: "Your Company Support"  
  template_id: 5977515  
  subject: "□□□□□□□□□□"
```

□□□□□□

- □□□□□□□□
- □□□□□□□□□□□□

- 00000000

00000000

- {{ var:customer_name }}
- {{ var:service_name }} - 00000000
- {{ var:expiry_date }}
- {{ var:balance_type }} - “00”“00”“00”“00”
- {{ var:renewal_url }} - 00/000000

000000

00000000000000000000

```
api_crmCommunicationsBalanceLow:
  from_email: "support@yourcompany.com"
  from_name: "Your Company Support"
  template_id: 5977516
  subject: "000000"
```

000000

- 00000000000000 20%
- 000000 OCS 000
- 0000000000

00000000

- {{ var:customer_name }}
- {{ var:service_name }}
- {{ var:current_balance }}
- {{ var:threshold }}
- {{ var:balance_type }}
- {{ var:topup_url }} - 00000000

eSIM 000000

0 eSIM 0000000000000000

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

📄📄📄📄

- eSIM 📄📄📄📄📄📄
- eSIM 📄📄📄📄📄

📄📄📄📄📄📄

- `{{ var:customer_name }}` - 📄📄📄📄
- `{{ var:iccid }}` - eSIM ICCID
- `{{ var:activation_code }}` - LPA 📄📄📄📄📄
- `{{ var:has_qr_code }}` - 📄📄📄📄📄📄 QR 📄📄📄📄📄
- `{{ var:has_activation_code }}` - 📄📄📄📄📄📄📄📄📄📄

📄📄📄📄

📄 `template_id` 📄 0 📄📄📄📄📄📄📄📄📄📄📄 OmniCRM 📄📄📄📄 HTML 📄📄📄📄 eSIM QR 📄📄
📄📄📄📄📄📄📄📄📄📄 `crm_config.yaml` 📄📄 `branding` 📄📄📄 — 📄📄📄📄📄📄 📄📄📄📄

📄📄 Mailjet 📄📄📄📄📄📄

📄📄📄📄📄📄📄📄📄📄📄 Mailjet 📄📄📄📄📄📄📄📄

📄📄 1📄📄 Mailjet 📄📄📄📄📄

1. 📄📄 Mailjet 📄📄📄📄
2. 📄📄📄 📄📄📄📄📄 → 📄📄📄📄📄
3. 📄📄 📄📄📄📄
4. 📄📄 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

00 2000000

00 Mailjet 0000000 HTML 00000000000000

000000

- 00 - 00000000
- 00 - 00 `{{ var:customer_name }}` 0 `{{ var:user_name }}` 000
- 00 - 0000000
- 00 - 0000000000000000
- 0000 - 00000000/00
- 00 - 000000000000000000

000000000000

```

<!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 Business St, City, Country
  </p>
</body>
</html>

```

○○ 3○○○○○○ ID

1. ○ Mailjet ○○○○
2. ○○ ○○ ID○○○○○○○ 5977509○
3. ○○ ID ○○○ crm_config.yaml

4. Mailjet

1. Mailjet API 키를 생성합니다.
2. Mailjet 계정에서 ESP를 설정합니다.
3. Mailjet 계정에서 ESP를 설정합니다.
4. Mailjet 계정에서 ESP를 설정합니다.

5. OmniCRM 설정

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: "비밀보안"
```

OmniCRM API 설정

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

CRM

Mailjet ESP를 사용하여 CRM 데이터를 Mailjet에 전송합니다.

UI

Customers → All Contacts (/contacts)에서 데이터를 가져옵니다.

marketing_consent_status, customer_status 필드를 CSV로 내보냅니다.

API

VIEW_CUSTOMER_CONTACT

Method	Endpoint	Parameters
GET	/crm/contact/search	search, page, per_page, sort, order, filters (JSON)
GET	/crm/contact/export	CSV, max 50,000 rows

CSV fields: contact_id, customer_id, customer_name, customer_status, contact_types, marketing_consent_status, marketing_consent_granted_at, created, last_modified

Filter: marketing_consent_status=granted. CSV output for Mailjet ESP.

OmniCRM-API/services/contact_service.py search_contacts, export_contacts_csv

XXXXXXXXXXXXXXXXXXXX

Mailjet integration with OmniCRM. "XXXXXXXXXXXXXXXXXXXX"

OmniCRM-API/services/marketing_consent_service.py marketing_consent_status == 'granted'

XXXX

XXXXXX

- contact.marketing_consent_status Enum: none | pending | granted | revoked
contact.marketing_consent_granted_at - timestamp

- `marketing_consent` - 用于接收来自 Mailjet 的 Webhook 通知。该配置项指定了用于验证消息的 ID 和 IP 地址，以及用于生成确认链接的 Mailjet ID。

“数据”在 `crm_config.yaml` 中 `marketing_consent.consent_text_version` 键值对下。

□□

□ `marketing_consent` 键值对 `crm_config.yaml`

```
marketing_consent:
  consent_text_version: "v1-2026-05"
  confirmation_url_path: "/marketing/confirm"
  token_expiry_days: 30
  re_permission_batch_size: 200
  re_permission_rate_per_second: 5
  re_permission_skip_pending_within_hours: 72
```

□□□□□□□□□□ Mailjet 用于生成 HTML 模板

```
mailjet:
  api_crmCommunicationMarketingConfirm:
    from_email: "support@example.com"
    from_name: "Norfone Support"
    template_id: 0 # 0 = 默认 HTML 模板
    subject: "订阅通知"
```

Mailjet 模板数据 `firstname` `confirmation_link` `consent_text_version`

□□ **A**□□□□

□□□□□□□□“Norfone 支持”□□□□□□□□□□□□□□□□□□□□□□□□

1. `routes/auth.py /self-signup-partial` 中的 `customer.subscribe_to_newsletter`

2. `marketing_consent_service.request_consent` `source='signup'` `IP`
3. `marketing_consent` `granted=True` `confirmed_at=NULL` `URL` `marketing_consent_status` `pending`
4. Mailjet `{public_url}{confirmation_url_path}?token=...`
5. `GET /crm/marketing-consent/confirm` `confirmed_at` `marketing_consent_status` `granted`

B

“” (/self-care/preferences)

GET	/crm/marketing-consent/me	+
POST	/crm/marketing-consent/me/subscribe	
POST	/crm/marketing-consent/me/unsubscribe	

`granted=False` `revoked` OmniCRM

C

`marketing_consent_status='none'`

Method	Endpoint	Description
GET	/crm/oam/marketing-consent/stats	Consent counts + Status counts
POST	/crm/oam/marketing-consent/re-permission?dry_run=true	Re-permission (Dry Run)
POST	/crm/oam/marketing-consent/re-permission?dry_run=false	Re-permission (Live)

Consent status: `none`, `revoked`, `pending`

`re_permission_skip_pending_within_hours` (Hours to skip pending)

`Closed` (Status)

`request_consent` (Request consent)

`re_permission_rate_per_second` (Rate per second)

Mailjet `limit` (Limit)

Implementation

1. `marketing_consent` (Marketing consent)
2. UI **Email Marketing → Marketing Consent** (`/email-marketing/marketing-consent`) — `none`
3. `limit` (Limit)
4. `limit` (Limit) 50 (Limit 50)
5. `limit` (Limit)
6. **Customers → All Contacts** `marketing_consent_status = granted` **Export CSV** Mailjet ESP

Prometheus

メトリック名	ラベル	値
<code>omnicrm_marketing_consent_events_total</code>	<code>event</code>	<code>confirmation_sent</code> <code>confirmation_faile</code> <code>granted</code> <code>revoked</code> <code>token_redeemed</code> <code>token_invalid</code> <code>token_expired</code>
<code>omnicrm_marketing_consent_status_total</code>	<code>status</code>	<code>none`</code>

Mailjet API `omnicrm_mailjet_api_*` `omnicrm_mailjet_emails_sent_*` のメトリックを監視する

インストール

前提条件

- Mailjet API キーと Mailjet アカウント ID を取得する
- 設定
 - `crm_config.yaml` に `api_key` と `api_secret` を設定する
 - Mailjet アカウント ID を取得する
 - Mailjet API キーを取得する

インストール

- OmniCRM のインストール
- 設定
 - 環境変数 `OMNICRM_API_KEY` と `OMNICRM_API_SECRET` を設定する
 - Mailjet のインストール

PDF

- PDF
- -
 -
 - PDF 15MB Mailjet
 - PDF

Mailjet

- OmniCRM Mailjet
- - granted
 - **Customers → All Contacts** GET
`/crm/contact/export?filters={"marketing_consent_status":["granted"]}`
 - UI CSV Mailjet

- `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 で監視するためのエンドポイント `/crm/metrics` を Prometheus に登録する

API

監視対象

- API - 監視対象/エンドポイント
- API - 監視対象
- API - OCS/CGRateS, Stripe, Mailjet API
- API - 監視対象
- **HTTP** API - API 監視対象

URL

URL: `http://your-omnicrm-api:5000/crm/metrics`

API: Prometheus 監視

API: Prometheus 監視対象

API

1. API

Ansible を使用して Prometheus を監視するための [Provisioning System](#) `Ansible Playbooks`

omnicrm_provision_jobs_total

API: 監視

📄:

- `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` - 可能値: `fatal`, `task_failed`, `timeout`
- `playbook` - Ansible の名前

例: 実行ログを出力する

2. 設定項目

OmniCRM の SQLAlchemy 設定項目の概要を [System Architecture](#) で確認してください。

omnicrm_db_query_duration_seconds

例: 1000

例:

- `operation` - SQL 操作: `SELECT`, `INSERT`, `UPDATE`, `DELETE`
- `table` - テーブル名

例: `[0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0, 5.0, 10.0]`

例: SQLAlchemy のクエリ実行時間

omnicrm_db_pool_size

例: 10

例: データベース接続プールサイズ

omnicrm_db_pool_overflow

例: 10

例: データベース接続プールオーバーフロー

omnicrm_db_pool_connections_checked_out

例: 10

例: データベース接続プールチェックアウト

omnicrm_db_errors_total

📄: 📄

📄:

- `error_type` - 📄
- `operation` - 📄: `connection_error`, `query_error`, 📄

📄: 📄

📄: 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

3. OCS/CGRateS 📄

OCS📄📄📄📄📄📄📄📄📄📄📄 CGRateS 📄📄📄📄📄📄📄📄📄📄📄📄 [Billing Overview 📄](#)
[CGRateS Integration](#)

omnicrm_ocs_api_calls_total

📄: 📄

📄:

- `method` - OCS API 📄: `GetBalance`, `SetBalance`, `SetAccount`, `RemoveAccount`, 📄
- `status` - 📄: `success`, `failed`

📄: OCS/CGRateS API 📄📄📄📄📄📄📄📄 API 📄📄 OCS 📄📄📄📄

omnicrm_ocs_api_duration_seconds

📄: 📄

📄:

- `method` - OCS API 📄: `GetBalance`, `SetBalance`, `GetAccount`, 📄

📄: `[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 Error Summary Table OCS API Error Summary Table

omnicrm_ocs_balance_queries_total

Table: Table

Table:

- `query_type` - Query Type: `single_account`, `multiple_accounts`

Table: OCS Balance Query Summary Table

omnicrm_ocs_action_plan_operations_total

Table: Table

Table:

- `operation` - Operation: `create`, `remove`, `query`

Table: CGRateS Action Plan Operations Summary Table

4. Stripe Integration

Stripe Integration [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 錯誤統計

統計期間: 2023-01-01 ~ 2023-01-31

omnicrm_api_errors_total

項目: 件数

項目:

- `endpoint` - API 名称
- `error_type` - エラー種別
- `status_code` - HTTP ステータスコード

項目: 発生時刻

omnicrm_api_auth_failures_total

項目: 件数

項目:

- `auth_method` - 認証方法: `jwt`, `api_key`, `ip_whitelist`
- `reason` - 失敗理由

項目: 発生時刻

9. API バージョン管理

omnicrm_api

項目: 件数

項目:

- `version` - API バージョン
- `environment` - 環境: `production`, `staging`, `development`

項目: OmniCRM API 使用状況

□□□□

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 Database Query Duration: {{ $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 }}" 🚫🚫🚫🚫

```

🚫🚫🚫🚫

🚫🚫🚫🚫

1. 🚫🚫🚫🚫 - 🚫🚫🚫🚫🚫🚫🚫
 - 🚫🚫🚫🚫 > 10%
 - 🚫🚫🚫🚫🚫🚫
 - OCS/CGRateS API 🚫🚫
 - Stripe 🚫🚫🚫🚫

2. Prometheus - KPI

- Stripe
-
-

3. Prometheus -

- API
-
- API

4. Prometheus -

-
-
- API

Prometheus:

- 15 - 15
- 90 - 5
- 1 - 1

Thanos Cortex VictoriaMetrics

/metrics nginx

:

```
curl http://localhost:5000/crm/metrics
```

Prometheus 配置:

```
# Prometheus 配置  
http://prometheus:9090/targets
```

配置

配置 Prometheus 监控目标

- Prometheus 配置
- Stripe 配置
- OCS 配置

配置

配置 Prometheus 监控目标

```
# Prometheus 配置  
count by (__name__) ().__name__=~".+")
```

配置 >10,000 配置 Prometheus 监控目标

配置

配置: 3431 配置 + 3 配置 Flask 配置

配置:

- 配置: 20
- 配置: 5
- 配置: 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 [Stripe](#) [PayPal](#)



OmniCRM

- [Stripe](#) [PayPal](#) OmniCRM
- [Stripe](#) [PayPal](#)
- [Stripe](#)
- [PayPal](#)
- [Stripe](#)
- [Stripe](#) [Self-Care Portal <self_care_portal>](#)
- [Stripe](#)

OmniCRM

- [Stripe](#) [PayPal](#)
 - [Stripe](#) [PayPal](#) [Visa](#) [Mastercard](#) [American Express](#) [Discover](#)
 - [Stripe](#)
 - [PayPal](#)
- **PayPal** [PayPal](#)

OmniCRM

OmniCRM

- [stripe](#) [paypal](#)
- [Visa](#) [Mastercard](#)

- 00000
- 0000/00
- 00000/00
- 00000000000000

0000000

000000

1. 000 **Customers** → [00000]
2. 00 **Billing** 00
3. 000 **Payment Methods** 00

00000

00000000000000

0000000000000000

0000000000000000 60 000000000000

0000000

0000000000000000

□□□□

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

□□□□□□□□

□□□□□□□□□□□□ : □□□□□□□□

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

□□□□□□□□

□□ **"Add Payment Method"** □□□□□□□□□□□□

□□□□□□□□□□□□□□□□

□□□□□□□□□□□ Stripe Elements □ PayPal SDK □□□□□□□□

□□□□□□

- 数字 - 数字 CVC Stripe 数字
- 数字 - 数字
- 数字/数字 - 数字

数字

- 数字 - 数字

数字

- 数字 iframe Stripe Elements / PayPal SDK
- OmniCRM 数字
- PCI DSS 数字
- 数字

数字

数字 "Add Payment Method" 数字

1. 数字
 - 数字
 - 数字
 - 数字 CVC 数字
2. 数字
 - 数字 OmniCRM
 - 数字 pm_1A2B3C4D 数字 Stripe
 - 数字 OmniCRM
3. 数字
 - OmniCRM 数字
 - 数字
 - 数字 OmniCRM 数字

数字

数字

数字 Visa 数字 1234 数字

□□□□□□□□□□□□

□□□□□□

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

□□□□□□□□

□□□□□□□□

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

□□□□□□□□

1. □□□□□□□□□□□□□□□□
2. □□□□□□□ □□□□□ : □
3. □□ **"Set as Default"**
4. □□□□□□

Visa □□ 5678 □□□□□□□□□□□□

□□□□□□□□□□□□

□□□□□□

□□□□□□□□

□□□□□□□□□□□□□□□□

如何删除支付方法

删除支付方法

删除支付方法

1. 选择要删除的支付方法
2. 点击“删除”按钮
3. 确认删除

删除支付方法

删除支付方法

删除支付方法

删除 Visa 1234 12/2026

⚠️ 删除支付方法后，该支付方法将无法用于支付。

[删除] [取消]

删除“Delete Payment Method”

删除支付方法

删除支付方法

删除支付方法

删除支付方法

- 删除支付方法后，该支付方法将无法用于支付。
- 删除支付方法后，该支付方法将无法用于支付。
- 删除支付方法后，该支付方法将无法用于支付。

XXXXXXXXXX

OmniCRM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXX

XXX **Billing** → **Expiring Cards** XXXXXXXXXXXXXXX

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

◆◆ XXX

- XXXX - XXXX 30/60/90 XXXXX
- XXXX - XXXXX
- XXXX - XXXXXXXXXXXXX

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. วัตถุประสงค์
2. วัตถุประสงค์ **Billing → Payment Methods**
3. วัตถุประสงค์
4. วัตถุประสงค์
5. วัตถุประสงค์
6. วัตถุประสงค์

วัตถุประสงค์

วัตถุประสงค์

- วัตถุประสงค์
- วัตถุประสงค์
- วัตถุประสงค์

วัตถุประสงค์

วัตถุประสงค์

วัตถุประสงค์

1. วัตถุประสงค์
 - วัตถุประสงค์
 - วัตถุประสงค์
 - วัตถุประสงค์
 - วัตถุประสงค์
2. วัตถุประสงค์
 - วัตถุประสงค์
 - วัตถุประสงค์
 - วัตถุประสงค์
3. วัตถุประสงค์
 - วัตถุประสงค์ - วัตถุประสงค์
 - วัตถุประสงค์ - วัตถุประสงค์
 - วัตถุประสงค์ - วัตถุประสงค์

วัตถุประสงค์

□ **1-3** □□□□□□

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

□ **4-7** □□

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

□ **8-14** □□

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

□ **15** □□□□□

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

□□□□□□□□

□□□□□□□□

- □□□□ **30** □ □□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□
- □□□□□□□□□□□□□□

□□□□□□□□

□□□□□□□□□□□□

1. □□□□□□□□□□

2. 商家
3. 商家代辦
4. 商家代辦
5. 商家代辦

商家代辦

商家

OmniCRM 商家代辦

1. 商家代辦 → 商家代辦
2. 商家代辦 → 商家代辦
3. 商家代辦 **OmniCRM** → 商家代辦
4. 商家代辦 → 商家代辦

OmniCRM 商家代辦

```
{
  "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

- 商家代辦
- CVV/CVC 代辦
- 商家代辦
- PIN 代辦

PCI 0000

00000000000000000000

- 00 **PCI** 00 - 00000000 OmniCRM 0000
- 00000000 - 00000000000000000000 iframe 0
- 00000 - 0000000000000000
- 00000 - 00000000 HTTPS/TLS

0000 `Payment System Guide <payment_system_guide>` 00000000000000000000

00000000

00000 **1**00000000000000000000

0000 00000000

1. 00000000
2. 00000000
3. 0000000000000000
4. 0 Stripe 0000000000000000
5. 0000000000
6. 00000000
7. 00000
8. 00000000

00000 **2**00000000000000000000

0000 00000000

1. 00000000000000000000 60 00
2. 0000000000
3. 0000 **Billing → Payment Methods**
4. 0000000000 12/2025
5. 00 **"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. 0000000 Mastercard 00 2222000000
4. 0000000 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 📌
- 📌
 - 📌
 - 📌
 - 📌
 - 📌
 - 📌

□□□□□□□□□□□□

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

□□□□□□□□□□“□□□□□□□”□□□□

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

□□□□□□□□

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

□□□□□□□□□□

- □□□ □□□□□ SDK □□□□□API □□□□□□□□□□□□□□
- □□□□□
 - □□□□□□□□□□□□□□
 - □□□□□□□□□□□□□□
 - □□□□□□□□□□□□□□□□
 - □□□□□□ SDK □□□□□□□□□□□□□□
 - □□□□□/□□□□□□□□

□□□□□□□□□□

- □□□ Mailjet □□□□□□□□□□□□□□
- □□□□□

- 編輯 crm_config.yaml 編輯 Mailjet 編輯
- 測試發送電子郵件
- 測試發送電子郵件
- 編輯 Mailjet 測試發送電子郵件

測試發送

測試發送

- 測試發送電子郵件
- 測試發送 2 測試發送電子郵件
- 測試發送 30 測試發送電子郵件
- 測試發送/測試發送電子郵件
- 測試發送電子郵件“測試 Visa”“測試 Amex”
- 測試發送電子郵件

測試發送

- 測試發送電子郵件
- 測試發送電子郵件/SMS/測試發送電子郵件
- 測試發送電子郵件
- 測試發送電子郵件
- 測試發送電子郵件
- 測試發送電子郵件 £0.01 測試

測試發送

- 測試發送電子郵件
- 測試發送 60/30/7 測試發送電子郵件
- 測試發送電子郵件/測試發送電子郵件
- 測試發送 Mailjet 測試發送電子郵件
- 測試發送電子郵件
- 測試發送電子郵件

□□□□

- `Payment System Guide <payment_system_guide>` - □□□□ API □□□□
- `payments_process` - □□□□□□□□□□
- `payments_invoices` - □□□□□□□□□□
- `features_topup_recharge` - □□□□□□□□□□
- `basics_payment` - □□□□□□□□
- `customer_care` - □□□□□□□□□□□□□□

OmniCRM 支付 API

简介

OmniCRM 支持 Stripe、PayPal、Square、Adyen、Braintree 支付 API 集成。

本文档介绍了如何配置 API 集成。

[Ansible Playbook](#) 配置 Playbooks 集成 API [Playbooks](#)

目录

1. 配置
 2. 支付
 3. 配置
 4. 配置 API
 5. 配置 API
 6. 配置 API
 7. API 配置
 8. 配置
-

配置

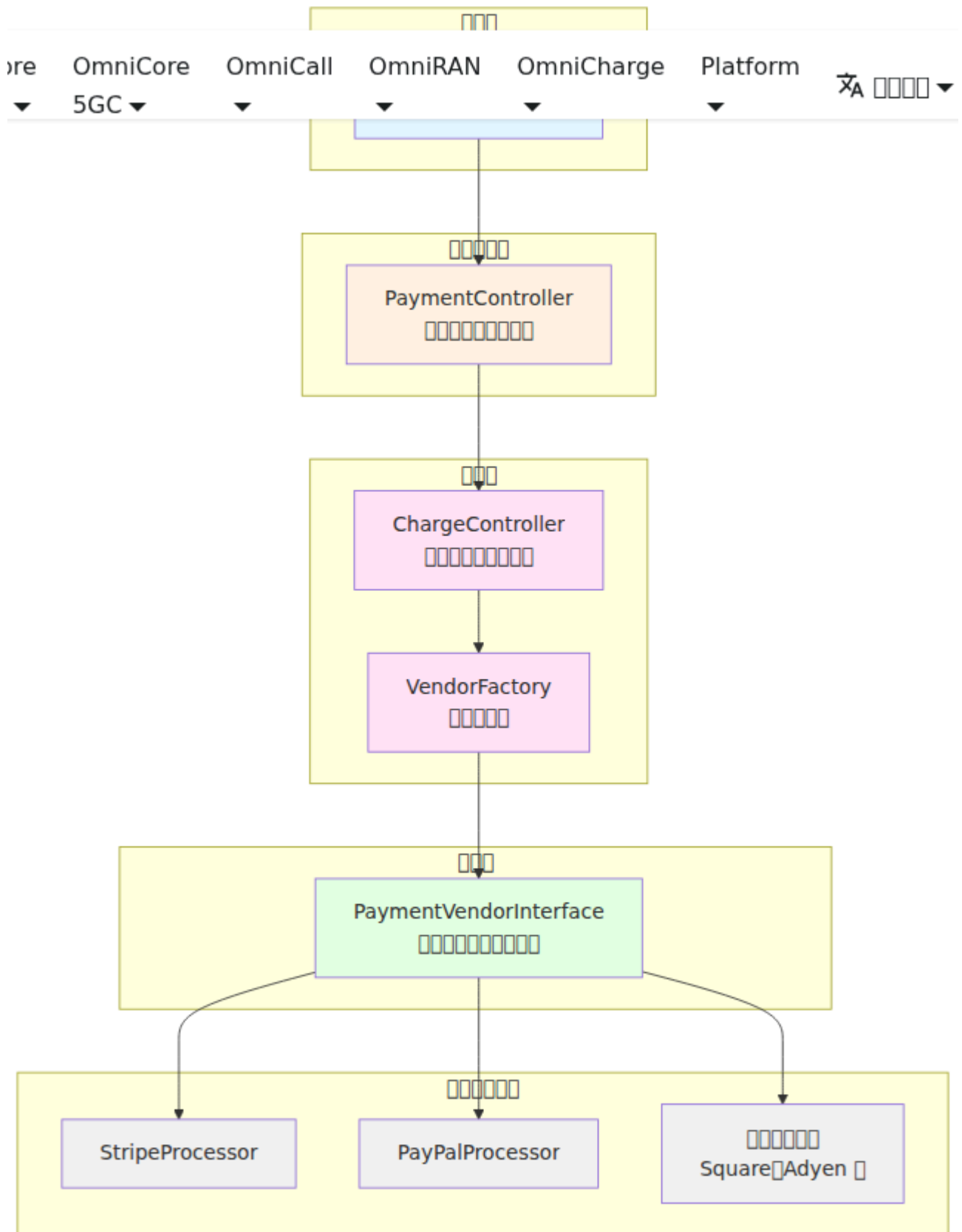
配置

本文档介绍了如何配置支付 API 集成。

- 配置支付 API 集成
- 配置支付 API 集成

- □ □□□□□□□
- □ □□□□□□□API □□

□□□



Payment Gateway

Payment Gateway: Square, Adyen, Braintree, OmniCRM

Payment Gateway: Stripe, PayPal

Payment Gateway

Payment Gateway

Payment Gateway

Payment Gateway

Payment Gateway

Payment Gateway

Payment Gateway API

Payment Gateway

- PaymentVendorInterface
- VendorFactory
-
- API

Payment Gateway

```
# Square
curl -X POST /api/payments/methods \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "vendor": "square",
    "payment_token": "sq_xxxxx"
  }'
```

Payment Gateway API

Payment Gateway 1-2

```
stripe:
  secret_key: 'sk_live_YOUR_SECRET_KEY_HERE'
  publishable_key: 'pk_live_YOUR_PUBLISHABLE_KEY_HERE'
  currency: 'aud'
  statement_descriptor_suffix: 'YOUR_COMPANY'
```

API `crm_config.yaml` Stripe API

3. `.env`

UI `.env`

```
VITE_STRIPE_PUBLISHABLE_KEY=pk_live_YOUR_PUBLISHABLE_KEY_HERE
```

PayPal

PayPal PayPal

1. PayPal API

- <https://developer.paypal.com/dashboard/> REST API
- **ID**
- `mode: 'sandbox'` # 'sandbox' 'live'

2. `crm_config.yaml`

`crm_config.yaml`

```
paypal:
  client_id: 'YOUR_PAYPAL_CLIENT_ID'
  client_secret: 'YOUR_PAYPAL_CLIENT_SECRET'
  mode: 'sandbox' # 'sandbox' 'live'
```

API `crm_config.yaml` PayPal API

3. 設定

UI `.env` 設定

```
VITE_PAYMENT_VENDORS="stripe, paypal"  
VITE_PAYPAL_CLIENT_ID=YOUR_PAYPAL_CLIENT_ID
```

`VITE_PAYPAL_CLIENT_ID` は `crm_config.yaml` の `paypal.client_id` 設定

設定表

	開発	本番
Stripe API	<code>sk_test_ / pk_test_</code>	<code>sk_live_ / pk_live_</code>
PayPal API	<code>api-m.sandbox.paypal.com</code>	<code>api-m.paypal.com</code>
PayPal mode	<code>sandbox</code>	<code>live</code>
備考		

Stripe API の `sk_test_ / pk_test_` は `crm_config.yaml` の `stripe.api_key` 設定
UI `.env` の `VITE_PAYPAL_CLIENT_ID` は PayPal の Client ID 設定

PCI 対応

OmniCRM の PCI 対応

- Stripe Elements / PayPal Card Fields を iframe で読み込む
- OmniCRM の PCI 対応
- PCI 対応のフレームワーク
- PCI 対応のライブラリ

□□□□□□

OmniCRM □□□□□□□□□□□□□□□□ Stripe □□□□□□□□□□□□ API □□□□□□□□□□□□
□□□□□□ □□□□□□

□□□□□□□□□□□□

□□□□□□□□□□□□□□□□



□□ - □□ Square □□

```
{  
  "vendor": "square",  
  "vendor_payment_method_id": "sq_card_abc123",  
  "payment_type": "card"  
}
```

□□□□□□□□□□□□□□□□

□□□□

□□□□

PaymentMethod

□□□□□□□□□□□□□□□□

```
{
  "payment_method_id": 789,
  "customer_id": 123,
  "vendor": "stripe", // 'stripe', 'paypal', []
  "vendor_payment_method_id": "pm_xxx", // [] 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", // 商家交易 ID
  "amount": 200.00,
  "currency": "USD",
  "status": "succeeded", // 'succeeded', 'failed',
  'refunded'
  "captured_at": "2025-12-27T10:30:00Z",
  "vendor_response": {}, // 商家响应
  "meta": {}
}
```

WalletAccount

商家钱包账户

```
{
  "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

商家钱包流水

```
{
  "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", // 消費
  "reference_id": 103,
  "meta": {},
  "created_at": "2025-12-27T10:35:00Z"
}
```

□□□□

□□□□□□□□□□□□□□□□ □□□□

□□

□□□□□□□□ □□□□□□□□□□□□□□□□ API □□□□□□□□□□□□ ID □□□□□□□□□□□□

`invoice_id` □□□□□□“□□”□

□□□□□□□□□□□□□□□□ □□□

□□□□□

```
{
  "invoice_id": 12345,
  "invoice_number": "INV-2025-000001", // 发票号 INV-YYYY-NNNNNN
  "customer_id": 123,
  "title": "商品采购",
  "paid": true, // 是否已付
  "void": false, // 是否作废
  "payment_reference": "ch_xxxxx", // 支付/收款 ID
  "payment_type": "stripe_capture", // 支付方式
  "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, // 零售成本
  "wholesale_cost": 250.00
}
```

发票号 INV-YYYY-NNNNNN 包含 1 个 1 个商品

商品

零售成本 retail_cost 包含 500.00

包含

- 商品名称 "商品"
- 商品 ID 12345
- 商品数量 1
- 商品零售成本 500.00

包含

- 商品 ID 12345
- 商品名称 "商品"
- 商品零售成本 - 商品零售成本

包含

```

// 1. 成功交易
{
  "transaction_id": 7001,
  "invoice_id": 12345,
  "retail_cost": 100.00, // 零售 = 零售
  "title": "咖啡"
}

// 2. 退款交易
{
  "transaction_id": 7002,
  "invoice_id": 12345,
  "retail_cost": -100.00, // 零售 = 零售
  "title": "咖啡退款 (12345)",
  "payment_type": "stripe_capture",
  "payment_reference": "ch_xxxxx"
}

// 零售$0 零售 → 零售

```

payment_reference payment_type
 payment_time

- invoice_id Hayden
- /
-

- 1.
- 2.

3. `POST /api/payments/`

4. `POST /api/payments/`

- `POST /api/payments/`
- `POST /api/payments/`

Stripe Stripe

API

URL `/api/payments/`

POST

POST

POST

POST `/api/payments/methods`

POST

```
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"
}'
```

POST

```

{
  "customer_id": 123,
  "vendor": "stripe", // 'stripe', 'paypal', etc
  "payment_token": "pm_xxxxx", // SDK token
  "is_default": false, // default flag
  "card_nickname": "My Visa Card" // nickname
}

```

201 OK

```

{
  "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}

Success

```
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}

200 OK

GET /api/payments/methods/default?customer_id={id}

200 OK

```
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

PUT

PUT /api/payments/methods/set-default

PUT

```
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": "成功",
  "data": {
    "payment_method_id": 790,
    "is_default": true
  }
}
```

成功

成功

DELETE /api/payments/methods/{payment_method_id}?customer_id={id}

```
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": "成功"
}
```

API

1. 成功

POST /api/payments/charge

```
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

Request

1. Request body
2. Headers - Authorization
3. Content-Type
4. Example request

Request

```
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"
  }
}'
```

Response

```
{
  "customer_id": 123,
  "amount": 200.00,
  "currency": "USD",
  "payment_method_id": 789,           // ID - payment_method_id
  "vendor": "stripe",                // vendor payment_token
  "payment_token": "pm_xxxxx",      // ID - payment_token
  "save_method": false,
  "use_wallet": true,                // use_wallet 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_to_use
    "card_amount": 50.00,           // card_amount
    "message": "Success $50.00 captured from $200.00"
  }
}
```

2. Capture

POST /api/payments/capture/{authorization_id}

Request

1. Authorization ID
2. Amount
3. Currency
4. Card/Token

Response

```
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": "□□□□",
    "description": "3□□□",
    "wholesale_cost": 100.00,
    "contract_days": 3,
    "send_email": true
  }
}'
```

□□□□

```
{
  "amount": 200.00, // □□ - □□□□□□□□□□
  "metadata": {
    "invoice": true, // □□□□□□□□□□
    "create_transaction": true, // □□□□□□□□
    "title": "□□□□",
    "description": "3□□□",
    "wholesale_cost": 100.00,
    "contract_days": 3,
    "send_email": true // □□□□□□□□□□
  }
}
```

□□□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": { // 成功
      "ledger_id": 401,
      "amount": 50.00
    },
    "wallet_debit": { // 成功
      "ledger_id": 402,
      "amount": 200.00
    },
    "transaction": { // 成功 invoice=true
      "transaction_id": 7001
    },
    "invoice": { // 成功 invoice=true
      "invoice_id": 12345,
      "invoice_number": "INV-2025-000001"
    }
  }
}

```

3 成功

POST /api/payments/release/{authorization_id}

成功

成功

```

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"  
  }  
'
```

HTTP/1.1 200 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/true  
  }  
}
```

5. 請求の送信

APIクライアントから送信する

APIクライアントから送信する `POST /api/payments/rental`

curl

```
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"
  }
}'
```

APIクライアントから送信する

```
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"
  }
}'
```

APIクライアントから送信する `200 OK`

```
{
  "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"
}'
```

响应

```
{
  "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": "成功充值 100.00 元"  
}
```

返回数据

成功充值 100.00 元

```
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` customer ID
- `limit` number of items (max 50)
- `offset` starting point (max 0)
- `type` transaction type: 'credit', 'debit', 'refund', 'adjustment'

status 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 一覧

決済方法

メソッド	URL	レスポンス
POST	<code>/api/payments/methods</code>	決済方法
GET	<code>/api/payments/methods?customer_id={id}</code>	決済方法一覧
GET	<code>/api/payments/methods/default?customer_id={id}</code>	デフォルト決済方法
PUT	<code>/api/payments/methods/set-default</code>	デフォルト決済方法設定
DELETE	<code>/api/payments/methods/{id}?customer_id={id}</code>	決済方法削除

決済

メソッド	URL	レスポンス
POST	<code>/api/payments/charge</code>	決済実行
POST	<code>/api/payments/invoice</code>	請求書発行
POST	<code>/api/payments/topup</code>	チャージ
POST	<code>/api/payments/authorize/hold</code>	予約決済
POST	<code>/api/payments/capture/{id}</code>	決済確定
POST	<code>/api/payments/release/{id}</code>	予約解除
POST	<code>/api/payments/rental</code>	リース決済
POST	<code>/api/payments/refund</code>	返金

API

Method	Endpoint	Response
GET	<code>/api/wallet/balance?customer_id={id}</code>	Balance
GET	<code>/api/wallet/info?customer_id={id}</code>	Wallet Info + Card
POST	<code>/api/wallet/topup</code>	Transaction ID
GET	<code>/api/wallet/transactions?customer_id={id}</code>	Transaction List

PayPal Integration

Method	Endpoint	Description
POST	<code>/api/payments/paypal/vault/setup-token</code>	Integrate PayPal SDK and setup token
POST	<code>/api/payments/paypal/vault/finalize</code>	Finalize PayPal integration
POST	<code>/api/payments/paypal/vault/update-setup-token</code>	Update 3DS/SCA token

API

API 1

API 30

```
# curl command to create a payment invoice
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-day service"
    }
  }'
```

Response

```
{
  "success": true,
  "message": "Payment successful",
  "data": {
    "service_amount": 99.99,
    "wallet_portion_used": 50.00,           // wallet $50
    "card_portion_used": 49.99,          // card $49.99
    "final_balance": 0.00
  }
}
```

2. Payment Gateway Integration

Integration with a payment gateway involves several steps:

```
# 請求のcurl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

請求内容の取得方法として、WiFi 接続環境で請求内容を確認する方法

```
# 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
- 充值失败

返回 **400** 错误 + 消息

错误消息

```
# 1 authorize - hold
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"
    }
  }'
```

```
# {"authorization_id": 301, "status": "authorized"}
```

```
# 2a capture - invoice
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 - 3",
      "description": "302 12 27 30",
      "send_email": true
    }
  }'
```

```
# 2b 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 100.00 元"
}
```

成功

成功

成功400 元

```
{
  "error": "余额不足 50.00 元"
}
```

成功400 元

```
{
  "error": "余额不足"
}
```

成功400 元

```
{
  "error": "customer_id 或 amount 错误"
}
```

404 Not Found

```
{  
  "error": "999"  
}
```

500 Internal Server Error

```
{  
  "error": "Internal Server Error"  
}
```

Payment Refund

Refund

How to Refund a Payment

1. Stripe/PayPal

Refund a Payment

Refund a Payment via PayPal

POST /api/payments/refund

Refund

□□□□

□□□□

□□□

□

□□□□□

□

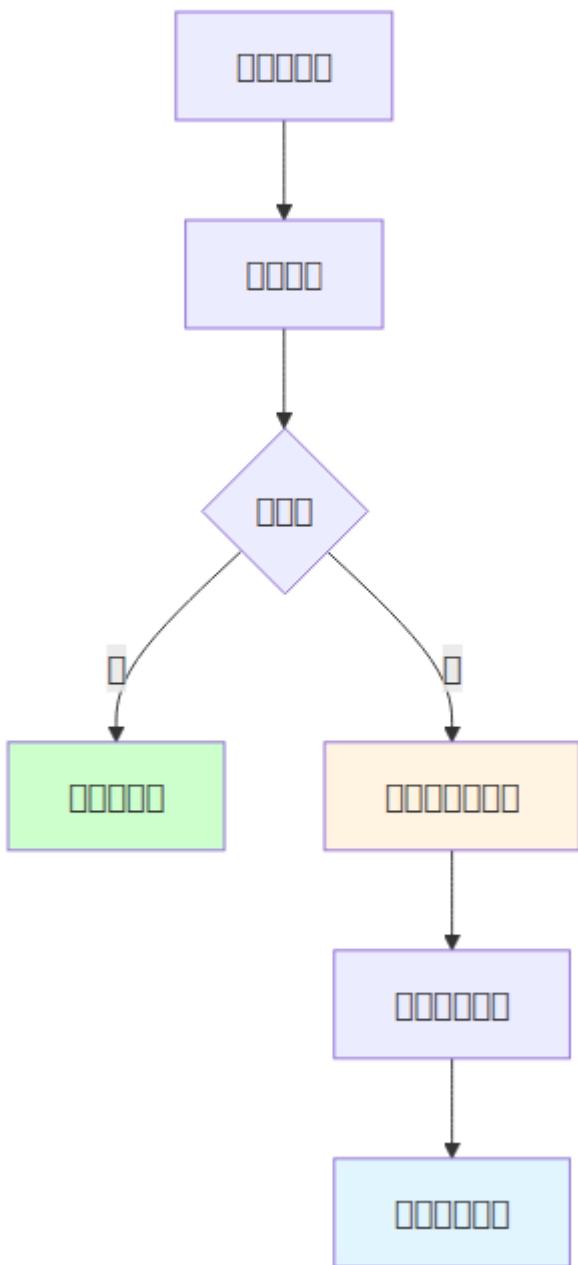
□□□□□□□

□□□□□□

□□□□□□

□□□□□

1. □□□□ \$100 □□□□□
2. □□□□□□□
3. □□□□□
4. □□□□ \$100 □□□□□5-10 □ + □□□□□□□
 - □□□ \$100 □□□□□□
 - □□□□□□□□□□□□
 - □□□□□□
 - □□□□□□□□



API 호출

```
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 返回

```
{
  "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, // 钱包 $1
    "card_portion_used": 9.00, // 卡片 $9
    "charged_amount": 9.00, // ← 卡片
    "wallet_credited": 9.00, // ← 钱包
    "wallet_debited": 10.00, // ← 钱包
    "final_balance": 0.00
  }
}
```

2 * \$50 钱包 + \$30 卡片

钱包 \$50 卡片 \$30

钱包

□□□□\$50

□□□□
\$30 - \$50 = \$0

□□□

□□□□

□□□□

□□□□-\$30

□□□□\$20
□□□□\$0
□□□□\$30 □

API □□□

```

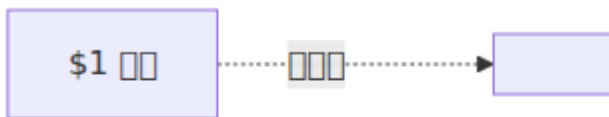
{
  "success": true,
  "data": {
    "service_amount": 30.00,
    "initial_balance": 50.00,
    "charged_amount": 0, // ← 0000
    "wallet_debited": 30.00,
    "final_balance": 20.00,
    "payment_method_used": false // ← 0000
  }
}

```

3. 결제 승인

카드 결제 \$500, 현금 결제 \$150

1. 결제 승인 POST /api/payments/authorize/hold



API 응답

```

{
  "authorization_id": 301,
  "amount": 500.00,
  "status": "authorized",
  "wallet_balance": 150.00,
  "wallet_to_use": 150.00,
  "card_amount": 350.00,
  "message": "카드 결제 $350, 현금 결제 $150 결제 승인"
}

```

2. 결제 승인 POST /api/payments/capture/301

□□□□

□□□□\$350

□□□□+\$350
□□□\$500

□□□□-\$500
□□□□

□□□□\$0
□□□□\$350

□□□□

- □ □□□ \$350 “□□” □□□□□□□□ \$500
- □ □□□□□□□□□□□
- □ □□□□□□□□□
- □ □□□□□□□

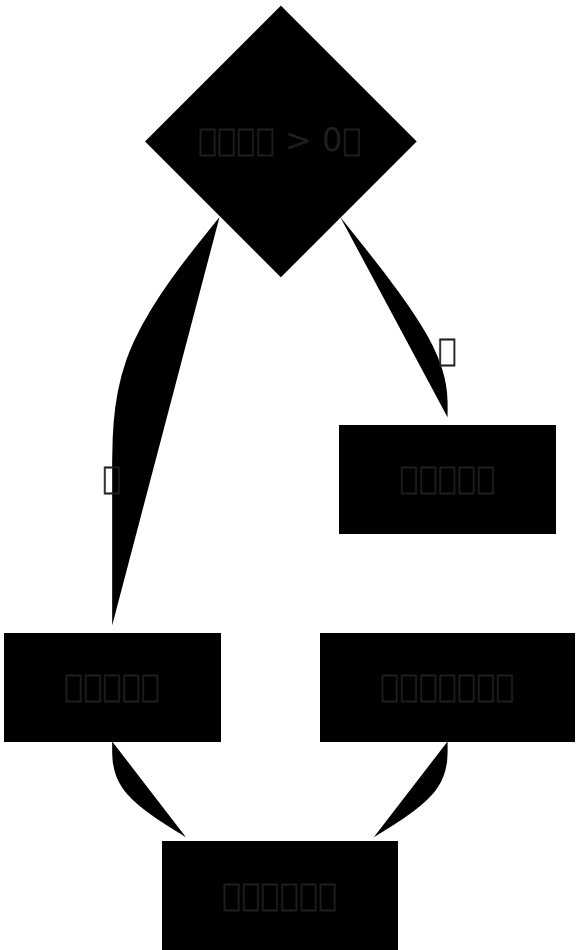
□□□□

□□□□□□□□□□

□□□□

□□□□□□

□□□
wallet_portion = min
balance, amount
card_portion = amount
- wallet_portion



00

00000

1. 0000000000
2. 00000 amount - wallet_balance
3. 0000 > 0 0000000000
4. 0000000000
5. 0000000000
6. 000000000000000000

0000000

000000000000000000

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 \$10000000 \$10000000 \$1000000\$00

000000000000000000/000000000000

□□□□

1. □□□□□□□□ □□□□□□□□□□□□
2. □□□□□□□□□□□□□□□□□□□□□□□□
3. □□□□□□□□ □□□□□□
4. □□□□□□□□□□□□□□□□□□□□
5. □□□□□□ □□□□□□□□□□
6. □□□□□□ □□□□□□□□□□
7. □□□□□□□□□□□□ 7 □□
8. □□□□□□ □□□□□□□□

Playbook □□□□□□

□ Ansible playbooks □□□□□□□□□□

1. □□□□□/□□□□ - □□□□□□ try/catch □□□□□□□□
2. □□ **authorization_id** - □□□□□□□/□□□□
3. □□ **API** □□ - □□□□□□□□□□
4. □□□□□□□□ - □□□□ 2 □□□□
5. □□□□□□□ - □□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□ **Playbook** □□□□□□



□□ API □□□□□□□□ API □□□□□□□□□□□□ **Authorization** □□□□

```
Authorization: Bearer YOUR_API_KEY
```

□□□□□□□□□□□□□□ API □□□□

□□□□□

□□□□

- **Stripe** - □□□□□□□□ACH□
- **PayPal** - □□□□□□□□□□□□□□□□ SDK □□□

□□□□□□

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

□□□□□

□□□□□

- **Playbooks** □□□□□□□□ - Ansible playbook □□
 - □□□□□□□□
 - □□/□□□□□
 - □□□□□□□
 - □□□□□□

□□□□□□

- □□□□□ - □□□□□□□□□□□□□□
- □□□□□ - □□□□□□□□□□□□□□□□ PayPal □□
- □□□ - □□□□□□□□□□□□□□
- □□□□□ - □□□□□□□□□□□□□□
- □□□□□□□ - □□□□□□□□

□□□□□

- □□□□□□ → □□□□ □□□□□□□
- □□ **Stripe/PayPal**□ → □□□□ □□□□□□□

- 000000 → 000 0000
 - 000000 → 000 **Playbook** 00
-



API API API API

Transactions </payments_transaction>

Mailjet <integrations_mailjet> PDF

OmniCRM API

OmniCRM API HTML templates Jinja2 templates

Templates

OmniCRM API `OmniCRM-API/invoice_templates/`

Templates

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

Example

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

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

Configuration

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. 模板目录

```
<!-- Placeholder URL -->
 (https://yourcompany.com/logo.png)

<!-- 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> <a href="https://yourcompany.com/pay">https://yourcompany.com/pay</a></p>
</div>
<div class="payment-info">
  <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 정보

```

<div class="terms">
  <h4>결제 조건</h4>
  <p>결제 기한: 30일 이내</p>
  <p>결제 수수료: 2%</p>
  <p>결제 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

1. 테스트
2. 테스트

3. PDF 生成

4. テンプレートエンジン

テンプレート

エンジン

Jinja2 のインストール

```

{% if total_amount > 1000 %}
  <div class="high-value-notice">
    <p><strong>注意</strong> 高価な商品です - 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 %}

```

インストール

テンプレート

```

invoice_template_en.html
invoice_template_es.html
invoice_template_fr.html

```

テンプレート

エンジン

```

<!-- 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 商品总额 tax_percentage 税率 0%

表头

表体

```

<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 引入字体

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. □□□□□□
 - □□□□□□□□
 - □□□□□□□□

URL

- URL 格式
- HTTPS URL

Table Layout

- Table layout
- `table-layout: fixed`

Font

- Font family
- Arial, Times New Roman

PDF Generation

- HTML to PDF using WeasyPrint
- PDF generation options

PDF

PDF generation process: OmniCRM PDF generation PDF generation

PDF Cache

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

2. 查詢 - 查詢所有已下載
 - 查詢 invoice_id 查詢 PDF
 - 查詢所有已下載的 PDF
 - 增加 `last_accessed` 查詢所有已下載
3. 查詢 - 查詢所有已下載 PDF 查詢
 - 查詢所有已下載/查詢所有已下載
 - 查詢所有已下載
 - 查詢所有已下載

查詢

- 查詢 - 查詢所有已下載 PDF 查詢
- 查詢 - 查詢所有已下載 PDF 查詢
- 查詢 - 查詢 PDF 查詢 CPU 查詢
- 查詢 - 查詢所有已下載

查詢

查詢 PDF 查詢所有已下載

- 查詢所有已下載 90 查詢
- 查詢所有已下載 30 查詢
- 查詢所有已下載

API 查詢

查詢 API 查詢 UI 查詢

- 查詢所有已下載 PDF 查詢
- 查詢所有已下載
- 查詢/查詢

查詢 查詢所有已下載

```
-- 查詢所有已下載 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" "00" 000000000000
- 0000000000000000
- 0000000000000000
- 000000000000

00000

0000000000000000

- **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

📌

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
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
 - 退款 `refund_reference`
6. "退款"

7. 銀行口座

🔍🔍🔍 銀行口座

- 銀行口座の種類
- 銀行 "口座"
- 銀行口座
- 銀行口座の種類 3-7 銀行口座
- Stripe 銀行口座

銀行

- 銀行口座 POS 銀行口座
- 銀行口座
- 銀行口座

銀行口座

銀行

銀行口座

- 銀行 ID
- 銀行
- 銀行口座

銀行

銀行口座

銀行

- 銀行 - 銀行口座
- 銀行 - 銀行口座

銀行

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.
 - "+
 - - 2025 1
 -
 - " "
 - "
- 6.

3

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

4

- 1.
- 2.
- 3.
 -

- **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. 0000000000000000000000 "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` "hello"
10. `return` "hello"
11. `return`
12. `return` "hello"

9. `return`

1. `return`
2. `return` Stripe `return`
3. `return`
4. `return`
5. `return`
6. `return` "hello" `return` Stripe `return`
7. `return`
8. `return` Stripe `return`
9. `return` "hello"
10. `return` 3-7 `return`
11. `return`

9. `return`

9. `return` - `return`

- `return` `return`
- `return` `return`

9. **PDF** `return`

- `return` `WeasyPrint` `return`
- `return` `HTML` `return` `API` `return`

9. `return` **Stripe** `return`

- `return` `Stripe API` `return`
- `return` `Stripe` `return`

Stripe

- Stripe API
- Stripe API

Mailjet

- Mailjet API
- Mailjet API

Stripe

- Stripe API
- Stripe API

PDF

- PDF
- PDF

Stripe

- Stripe API
- Stripe API

Stripe

- Stripe API
- Stripe API

Stripe

- Stripe API
- Stripe API

Stripe

- Stripe API
- Stripe API



□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

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

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

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

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

□□□□

□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

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

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

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

□□□□

□□□□□□□□□□□□□□□□

□□□□□□

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

□□□□□□□□

□□□□

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

□□□□□

□□□□□□□□□□□□ : □□□□□□□□

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

□□□□□

□□□□□□□□□

□□□□□□□□

□□□□□□□□□□□□□□

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

□□□□□□□/□□□□

□□□□□□□□□□□□

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

□□□□□□□□

□□ " + □□□□ " □□□□□□□□□□□□

□□□□□□□□□□

□□□□□□□□/□□□□

□□□□

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

□□□

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

□□□□□□

1. □□□□□□□□□□
2. □□□□□□□□□□

3. 0000“000000”0000
4. 0000000000
5. 0000000000

0000000000

00

00000000000000000000

- 00ID
- 00
- 00
- 00ID

0000

0000000000000000

000000

- 0000 - 0000000000
- 0000 - 0000000000

000000

- 000000 - 000000000000
- 000000 - 0000000000000000

00

00000000000000

- **ID** - 0000ID000000/000000
- 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 - 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
- 0000000000 (✓)
- 000000

000000

0000000000000000

1. 00000000000000000000
2. 000000000000
3. 000000000000
4. 0000 invoice_id 000000
5. 00000000"0000"

0000 payments_invoices 00000000000000000000

00000000

000000 **1**0000000000000000

1. 000000"00000020"
2. 00000000£100000
3. 000000 00 00
4. 00 "+ 000000"
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 **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. 0000000000000000 : 0
4. 00 "0000"
5. 0000000000
6. 000000000000
7. 00000000000000

□□□□

- `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
# ...
```

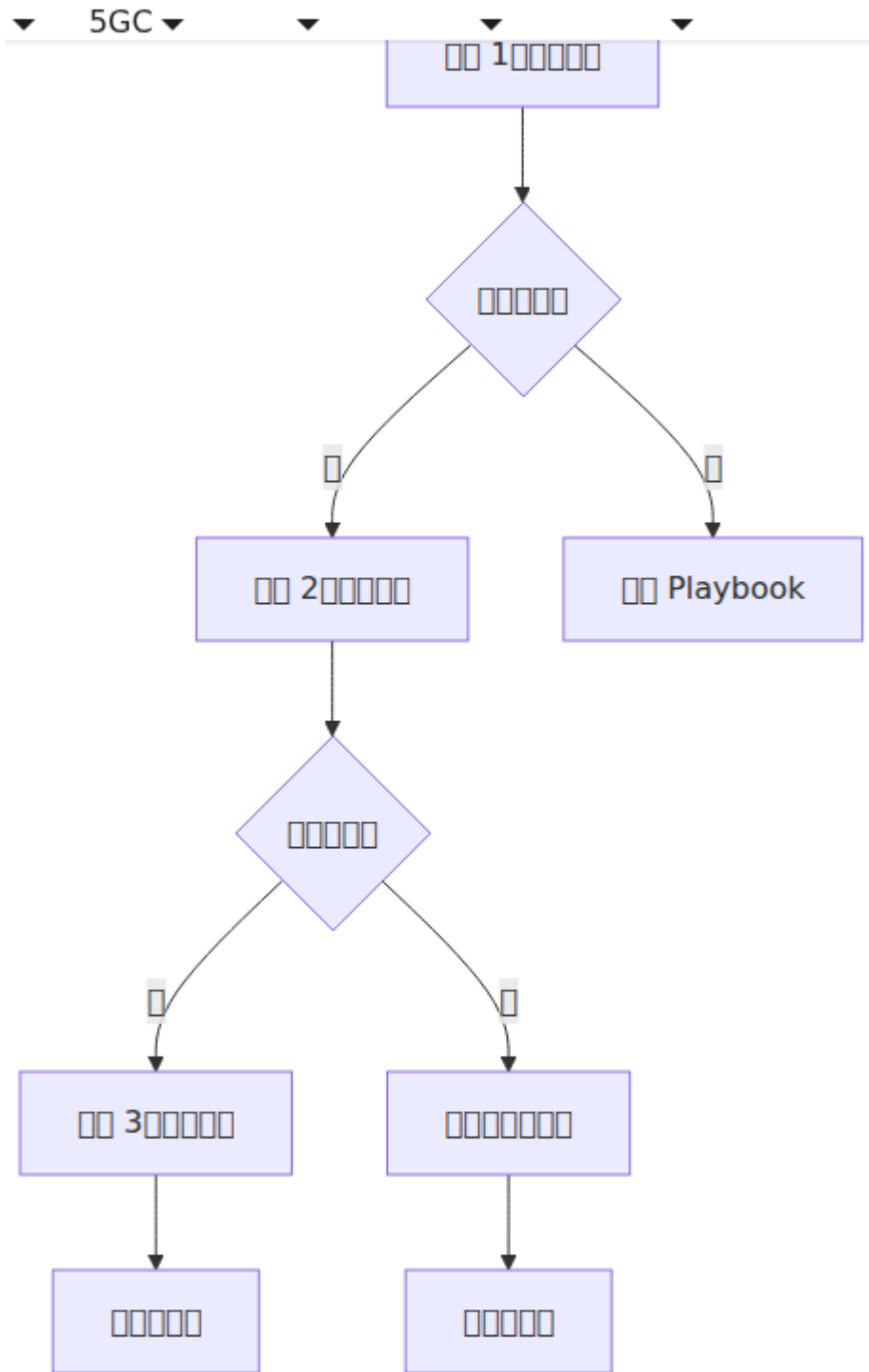
□□□□□□□□

```
# □□□□□□□□□□□□
- 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) }}"
```


□□□□



□□□□

□□□□□□□□□□

□□ **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: OCS playbook
  fail:
    msg: "OCS provisioning {{ authorization_id }} failed"

```

API

1. Provisioning failed for \$1 and \$10 payments
 2. \$9 provisioning failed

1. /

POST /crm/payments/authorize/hold

\$150 \$500 \$350
 1

```
{
  "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

□□□

```
{
  "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 호출 시 반환되는 응답 구조를 살펴보면 다음과 같습니다.

예시

```
{
  "metadata": {
    "release_reason": "provisioning_failed",
    "error_details": "OCS provisioning failed"
  }
}
```

예시

```
{
  "success": true,
  "message": "Success"
}
```

참고 사항

1. API 호출 시 반환되는 응답 구조를 살펴보면 다음과 같습니다.
2. **???** provisioning failed
3. provisioning failed
4. Success
5. Success

API 호출 시 반환되는 응답 구조를 살펴보면 다음과 같습니다.

예시

API 호출 시 반환되는 응답 구조를 살펴보면 다음과 같습니다. API 호출 시

API 호출 시

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: []
  uri: ...
  register: api_response_authorization

- name: []
  assert:
    that:
      - api_response_authorization.status == 200
      - api_response_authorization.json.success == true
    fail_msg: "[]{{ api_response_authorization.json }}"

```

3. [] ID

[] authorization_id []/[]

```

- name: [] authorization_id []/[]
  set_fact:
    authorization_id: "{{
api_response_authorization.json.data.authorization_id }}"

```

4. []

[][][][][]

```

metadata:
  description: "[]"
  service_id: "{{ service_id | int }}"
  product_id: "{{ product_id }}"
  user_id: "{{ initiating_user | int }}"
  title: "[]"
  wholesale_cost: "{{ wholesale_cost | float }}"
  invoice: true # []
  send_email: true # []

```

5. []

[][][][][]

```
- name: 000000000000
  set_fact:
    final_cost: "{{ (base_cost | float * multiplier | float) |
round(2) }}"
```

6. 000000000000

0000000000000000000000000000000000000000

```
- 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

0000000000000000

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`□□□

- 0000000000000000

0000000000

000 0000000000000000

00000

- 0000000000000000/0000/00000
- 00000   0000000000
- 0000000

00000

- 0000000000000000000000000000
- 0000000000000000
- 0000000000
- 000000000000

0000000000

0000

0000000000000000

1. 00000000 - 000000000/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 - 請求パラメータを指定する

請求パラメータを指定する

- 請求
- 請求パラメータ
- 請求

請求パラメータを指定する

請求

請求パラメータを指定する

- Stripe ACH
- PayPal PayPal

請求パラメータを指定する playbooks 請求パラメータを指定する Square Adyen Braintree

請求

- 請求 - 請求パラメータ
- 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` - □□□□□□□□□□

-  — `Admin` `Support` `Finance`
- `admin` — `admin`

JWT API IP /

OmniCRM

OmniCRM Admin Support Finance CRM

admin

admin - "admin"

admin - Admin Support Finance CRM

admin - admin

OmniTouch

- TempP@ssw0rd!
-
-
-

API

POST /auth/users

admin

```
{
  "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"
}
```

```
{
  "id": 123,
  "username": "john.smith",
  "email": "john.smith@company.com",
  "first_name": "John",
  "last_name": "Smith",
  "roles": ["Support"],
  "created": "2025-01-04T10:30:00Z"
}
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

```
{  
  "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
  ]
}
```


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. ユーザー名 → パスワード
2. ユーザー名
3. ユーザー名/パスワード
4. ユーザー名**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}

POST /auth/forgot_password

```
{
  "email": "user@example.com"
}
```

Response

2. Reset Password

POST /auth/reset_password

```
{
  "token": "abc123...",
  "new_password": "NewSecureP@ssw0rd!"
}
```

Response

Response

Web UI

1. Login →
2. Password**"****"***
3. Password
4. **"****"
5. Password
6. Password

API

POST /auth/users/{user_id}/admin_reset_password

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. 存储验证信息

□□□□

- 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
- バックアップコードを再生成
- 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
- 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

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

PUT /auth/users/{user_id}

```
{  
  "deleted": false  
}
```

deleted

API

API

API

UI GDPR

API

- API - API
- API - API
- API - API
- API - API
- API - API
- API - API

API

CRUD

- view_* — API
- create_* — API
- update_* — API

- delete_* — 删除/删除

权限名称**“权限名称...”**权限名称/权限名称

删除/删除

- `admin` — 权限名称/权限名称
- `can_impersonate` — 权限名称/权限名称
- `grafana_access` — Grafana权限名称/权限名称
Grafana权限名称

权限名称/权限名称

- 权限名称
 - view_customer[] create_customer[] update_customer[] delete_customer
 - 权限名称 权限名称
- 权限名称
 - view_customer_site[] create_customer_site???
update_customer_site[] delete_customer_site
 - 权限名称 权限名称
- 权限名称

□□

- 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，它决定了谁可以做什么。权限管理通常分为角色权限和细粒度权限。角色权限是指将权限分配给不同的角色，而细粒度权限是指将权限分配给具体的用户或组。

权限管理

角色	权限	备注
管理员 角色	admin 角色 can_impersonate 角色 CRUD 权限	系统/管理/权限管理
普通 用户	view_customer 角色 view_customer_service 角色 view_communication 角色 view_provision 角色	普通用户 can_impersonate
普通 用户	view_customer_invoice 角色 view_customer_transaction 角色 view_product 角色 create_customer_invoice 角色	普通用户权限
普通 用户 角色 角色	“权限管理...” 权限管理是系统中最重要的一部分，它决定了谁可以做什么。	普通用户权限
普通 用户 角色	普通用户 view_*	系统/管理/权限

权限管理是系统中最重要的一部分，它决定了谁可以做什么。

API 权限管理

权限管理 admin 角色

权限管理

□□□ 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 impersonation permissions
- Returns "impersonated..." in the response
- Requires impersonation permissions
- Requires `admin` or `can_impersonate`

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

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

〇〇〇〇〇〇〇

- 〇〇〇 - 〇〇Stripe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 - 〇〇〇〇4〇〇〇〇〇〇〇〇〇〇〇

〇〇〇 `<payment_methods>` 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

1. 〇〇〇 〇〇 → 〇〇〇〇 → “〇〇”
2. 〇〇〇〇〇〇〇〇〇〇

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 <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. 00000000000000000000
3. 000000000000 “000000”
4. 0000000000 + 0000
5. 00000
6. 0000000000

□□□ □□□□ - □□□□ <csa_modify> □□□□□□□□□□□□□□□□

□□□□□

□□□□□□□□□□□□□□□□

□□□□□□□

- □□□□□□□□□□
- □□□□□□□
- □□□□□□□□7□□3□□1□□
- □□□□□□□□□□□□□□□□
- □□□□□/□□
- □□□□□□□
- □□□□□□□□□□□□□□

□□□□□□□

- □□□□□□□
- □□□□□□□□
- □□□□□□□□□□10%□
- □□□□□□□□
- □□□□□□□□□□□□□□□□□□□□

□□□□□

□□□□□□□



- □□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□□□<20%□□□□
- □□□□□□□□□□□□□□

□□□

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

□□□□□□

- □□ `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
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8000
- 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

API

1. API
2. "API"
3. API
4. API
5. API

API

- API
- API
- API
- API
- API

API <concepts_api> API

API

API

API

API

API

API

API "API"

