

OmniCRM (2FA)

OmniCRM 2FA 設定

OmniCRM には RBAC (2FA) 機能が搭載されており、TOTP (Time-based One-time Password) を利用して、ユーザーのログインを強化することができます。

RBAC <rbac> 2FA 設定 Authentication Flows <authentication_flows> を設定する必要があります。

手順

2FA 設定

- 設定 — 設定メニューを開く
- RBAC — RBAC を有効にする
- 2FA — 2FA を有効にする
- Authenticators — Google Authenticator、Authy、Microsoft Authenticator をインストールし、TOTP コードを入力

2FA 設定

2FA 設定

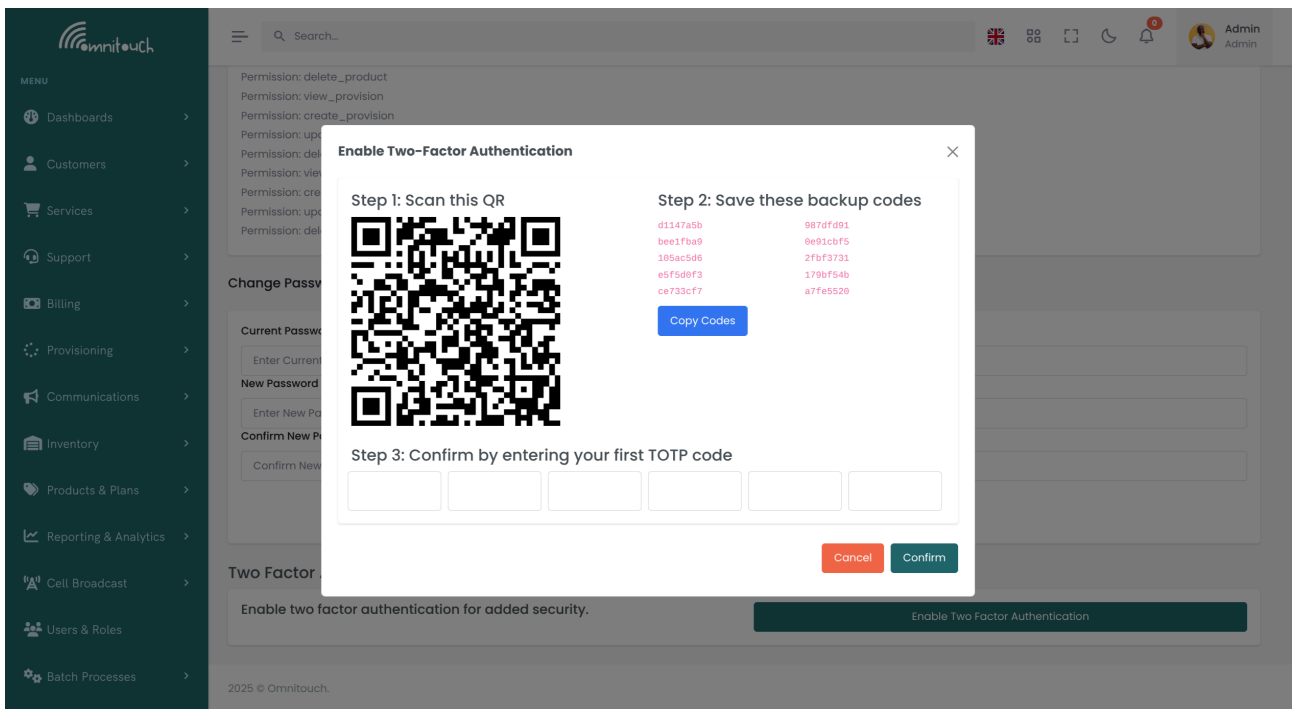
- 設定 — 設定メニューを開く
- RBAC — RBAC を有効にする
- 2FA — 2FA を有効にする
- Authenticators — Google Authenticator、Authy、Microsoft Authenticator をインストールし、TOTP コードを入力

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 Web 应用程序的重要组成部分。

- 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，是 Web 应用程序的重要组成部分。

API

- API 是应用程序编程接口，是 Web 应用程序的重要组成部分。

- **API** - **API**
- **API** - **API**
- **API** - **API**
- **API** - **API**
- **API** - **API**
- **API** - **API**

API

API

API

- **API** - **API**
- **API** - **API**
- **API** - **API**
- **API** - **API**

API

1 "API"

API +

2

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

1. 调用地址 : 调用地址
2. 调用 “调用地址”
3. 调用

调用

调用地址

调用地址

调用地址

- 调用地址
- 调用地址
- 调用地址
- 调用地址

API 调用

API 调用

1. 调用地址 : 调用地址
2. 调用 “调用地址”
3. 调用

调用地址

- 调用地址
- 调用地址 401 Unauthorized
- 调用地址
- 调用地址 - 调用地址

调用地址

- 调用地址

- 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
- 00000000

□□□□

- □□□□□□□□ 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 안전하게 저장
- API 키는 `sk_live_`로 시작

API 키 관리

CRM 관리

- CRM 관리
- CRM 관리
- CRM 관리
- CRM 관리

API 키 관리

- API 키 관리
- API 키 관리 (AWS Secrets Manager, HashiCorp Vault)
- API 키 관리
- API 키 관리 (90-365)

API 키 관리

- API 키 관리 - API 키 관리
- API 키 관리 - API 키 관리

- 認證/授權
- 認證

認證

- API 認證
- 認證
- “認證”
- 認證

認證

認證

1. 認證
2. 認證
3. 認證
4. 認證

認證

API 認證“401 Unauthorized”

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

認證“API 認證”

- 認證
- 認證
 - API 認證

- 0000000000000000
- 0000000000

API 00000000

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

00000 API 00

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

API 0000000000000000

API 00000000 API 0000000000000000

00 API 00

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

00 API 00

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

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

□□□

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

□□ **API** □□

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

□□ **API** □□

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

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


4. API 概念

概念

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



Placeholder text line 1

Placeholder text line 2: `Customer Tags <administration_tags>` Placeholder text line 3: `Customers, Contacts, Sites & Services <basics_customers>`

Placeholder text line 4



Placeholder text line 5

1. Placeholder

Placeholder text line 6: API

- `external_crm_id` = "SF-12345" Placeholder text: Salesforce ID
- `legacy_system_id` = "OLD-CRM-789" Placeholder text: Legacy System ID
- `hubspot_contact_id` = "12345678" Placeholder text: HubSpot ID

2. Placeholder

Placeholder text line 7

- `billing_method` = "quarterly" Placeholder text: Billing Method
- `auto_provision` = "true" Placeholder text: Auto Provision
- `support_tier` = "premium" Placeholder text: Support Tier
- `credit_limit` = "10000" Placeholder text: Credit Limit

3. Placeholder

Placeholder text line 8

- `gdpr_consent_date` = "2025-01-01" Placeholder text: GDPR Consent Date
- `tax_exempt` = "true" Placeholder text: Tax Exempt

- `regulatory_entity = "FCC-123456"`

4. `Account`

`Account`

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

5. `Radius`

`Radius`

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

`Radius`

`Radius`

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

`Radius`

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

UI 說明

說明

說明

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

說明

說明

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

說明

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

說明

說明

POST create_customer_attribute

Request

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

Response

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

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"
  }
]
```


API 호출

API 호출을 위한 Python 코드 예시

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

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

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

결과

API 호출 시 발생할 수 있는 에러

요청

1. 헤더

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

2. 본문

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

3. []

- []
- []
- []

4. []

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

5. []

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

[][][]

[][]

[][][][]

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

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

更新属性

更新属性

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

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

更新属性

更新属性

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

API

API endpoints

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

API uses `rbac` for permissions

UI

UI components

- API endpoints
- UI components
- `view_customer_attribute` UI

UI components

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

UI components

- UI components

- 𐀀𐀁𐀂𐀃𐀄𐀅𐀆𐀇𐀈𐀉𐀊𐀋𐀌𐀍𐀎𐀏𐀐𐀑𐀒𐀓
- 𐀔𐀕𐀖𐀗𐀘𐀙𐀚𐀛𐀜𐀝𐀞𐀟

𐀀𐀁𐀂𐀃𐀄

- 𐀅𐀆𐀇150𐀈𐀉𐀊𐀋𐀌
- 𐀍𐀎𐀏𐀐𐀑𐀒𐀓𐀔𐀕𐀖𐀗𐀘𐀙𐀚𐀛𐀜𐀝𐀞𐀟𐀠𐀡𐀢𐀣𐀤𐀥𐀦
- 𐀧𐀨𐀩𐀪𐀫𐀬𐀭𐀮𐀯𐀰𐀱𐀲𐀳𐀴𐀵

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`

データベース

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

環境変数

- `username` - MySQL ユーザー名
- `password` - MySQL パスワード
- `server` - データベースサーバー IPアドレスlocalhost

接続文字列

- データベース名 `omnicrm`
- データベースポート `3306` MySQL データベース
- データベース接続文字列 `mysql+pymysql://username:password@server/omnicrm`

環境変数と接続文字列を環境変数ファイルに設定します。

環境変数ファイル `.env` を作成し、データベース接続文字列を `db` Docker 環境変数に設定します。

サービスタイプ

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

サービスタイプを環境変数に設定します。

デフォルト値

- デフォルト値

- 設定する必要がある項目
- 設定方法
- 注意点

設定する必要がある項目

HSS 設定

HSS 設定

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

設定

- `hss_peers` - HSS の URL
- `apn_list` - APN ID のリスト

設定方法

Mailjet 設定

OmniCRM から Mailjet 設定

```
mailjet:
  api_key: your_mailjet_api_key
  api_secret: your_mailjet_api_secret

# 000000
api_crmCommunicationCustomerWelcome:
  from_email: "support@yourcompany.com"
  from_name: "Your Company Support"
  template_id: 5977509
  subject: "Welcome to YourCompany"

# 000000
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"

# 0000000000/0000
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"
```

```
# 日本語
```

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

```
# 英語
```

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

```
# 日本語
```

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

```
# 英語
```

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

Mailjet

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

Variables

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

Example

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

Variables

- `failure_list` - Ansible list of email addresses
- `template_filename` - Invoice template filename
- `template_directory` - Invoice template directory

Example

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

Variables

Variables PDF templates Jinja2 HTML

Variables `/OmniCRM-API/invoice_templates/`

Variables PDF templates

CRM URL

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

URL

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

URL

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

URL

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

OCS CGRates

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

URL

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

URL

SMSC 設定

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

説明

- SMS 認証 (2FA) 設定
- `source_msisdn` - 送信元 ID
- `smc_url` - SMSC API URL
- `api_key` - SMSC API キー

設定

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

CBC API 設定

```
features_cell_broadcast
```

JWT 設定

```
jwt_secret: 'CHANGE_ME_ON_DEPLOYMENT' # Ansible 固有
```

説明

- JWT 設定
- 秘密鍵 - 必須
- 署名アルゴリズム
- 有効期限

Ansible 固有

Ansible JWT secrets

1. Generate Ansible secrets 64 characters long
2. Store Ansible secrets
3. Use secrets in Ansible - Playbooks

Steps

- Git repository
- Ansible playbooks
- Ansible vault
- Ansible secrets

Ansible secrets

Ansible secrets

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

crm_config.yaml

```
jwt_secret: 'your_generated_secret_here'
```

Notes

- Change `CHANGE_ME_ON_DEPLOYMENT`
- Ansible secrets - Playbooks
- Ansible vault
- Ansible secrets

Stripe 設定

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

設定

- `secret_key` - Stripe API 接続用キー
- `publishable_key` - Stripe 公開用キー
- `currency` - ISO 4217 コード (aud, usd, gbp, eur など)
- `statement_descriptor_suffix` - 明細書に追加されるサフィックス

テスト

- テスト用キー: `sk_test_...` と `pk_test_...`
- 本番用キー: `sk_live_...` と `pk_live_...`

その他の設定

- 明細書に追加されるサフィックス "YOURCOMPANY"
- 22 桁の数字
- 4桁の数字
- PDF 明細書のファイル名: `YOURCOMPANY_12345.pdf`

Payment Vendor Integrations
<integrations_payment_vendors>

API IP 設定

API IP 設定

```

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

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

```

□□□

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

□□ **API** □□□

```
openssl rand -base64 48
```

□□□

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

□□□□□□□

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

□□□□□

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

本地 IP 地址 `administration_api_keys` 和 `concepts_api`

环境变量 (.env)

React UI 项目 `OmniCRM-UI/.env` 文件

路径 `/OmniCRM/OmniCRM-UI/.env`

API 配置

```
# Google Maps API 配置
VITE_GOOGLE_API_KEY=your_google_api_key

# Stripe 配置
VITE_STRIPE_PUBLISHABLE_KEY=pk_test_xxxxx

# 本地 npm start 配置
BROWSER=none
```

配置

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

环境变量

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

环境变量 UI 配置

- `COMPANY_NAME` - 公司名
- `PORTAL_NAME` - 门户/管理平台名称“ShellManager”
- `SELF_CARE_NAME` - 自助服务名称“ShellCare”
- `COMPANY_TAGLINE` - 公司标语

环境变量

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

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

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

语言支持

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

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

Moneda

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

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

Colores

```
# Color primario
VITE_PRIMARY_COLOR=#405189

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

Definición

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

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

`#RRGGBB`

`#RRGGBB`

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

`#RRGGBB`

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

OmniCRM-UI

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

`#RRGGBB`

`#RRGGBB`

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

Font families

- Merriweather, Lora, Playfair Display

Font families

- System - default system fonts

Font families

Font families

Font families

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

```
//
// 思源宋体 - 宋体
//

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

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

3. `.env` 配置

```
VITE_FONT_FAMILY=Your Font Name
```

配置

- 300 - 思源宋体
- 400 - 思源宋体
- 500 - 思源宋体
- 600 - 思源宋体
- 700 - 思源宋体

Web 配置

配置 6 个 Web 配置

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

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

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

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

GitHub Xero Monday.com Gmail MailJet Slack

Grafana

OmniCRM Grafana `grafana_access` `/grafana`
Grafana OmniCRM

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

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

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

Dashboard ID

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

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/faqs
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. Measurement ID

Google Analytics 4

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

Google Tag Manager

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

2. .env

```
VITE_GOOGLE_TAG_ID=G-XXXXXXXXXX
```

3. UI

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

4. Deployment

-
-
- Google Analytics

Deployment -

-

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

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

00 - checkout

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

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

00 - form_submission

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

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

00000000

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

Implementation

Google Analytics Implementation

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

Configuration

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

Implementation

- 環境変数
- 環境変数 < 50KB
- 環境変数
- `VITE_GOOGLE_TAG_ID` 環境変数

環境変数

- `OmniCRM-UI/src/utils/googleAnalytics.js`
- `OmniCRM-UI/src/utils/usePageTracking.js`
- `App.js` `index.js` 環境変数

環境変数

```
VITE_TOPUP_PRICE_PER_DAY=10
```

環境変数/環境変数

環境

- `VITE_TOPUP_PRICE_PER_DAY` - 環境変数

環境 `10` USD `$10` 環境

環境 `features_topup_recharge`

環境

OmniCRM 環境

環境

環境 `/OmniCRM-UI/src/assets/images/omnitouch/` 環境

環境

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

100x100px

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

100x100px

100x100px

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

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

100x100px

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

100x100px

1. 100x100px

- PNG 100x100px
- 100x100px
- 100x100px

2. OmniCRM

```
# omnitouch
cp /path/to/your/logoSm.png OmniCRM-UI/src/assets/images/omnitouch/
cp /path/to/your/logoDark.png OmniCRM-UI/src/assets/images/omnitouch/
cp /path/to/your/logoLight.png OmniCRM-UI/src/assets/images/omnitouch/
```

3. UI

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

4. Assets

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

Assets

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

Assets

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

```

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

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

<div class="invoice-details">
    <p><strong>Invoice Number:</strong> {{ invoice_number }}</p>
</div>

    <p><strong>Date:</strong> {{ date }}</p>
    <p><strong>Due Date:</strong> {{ due_date }}</p>
    <p><strong>Billing Period:</strong> {{ start_date }} to {{
end_date }}</p>
</div>

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

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

```

```

</table>

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

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

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

```

📌

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

📌

1. 📁📁📁📁

```

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

```

2. 設定

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

3. 設定

例 `crm_config.yaml`

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

4. 実行

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

5. テスト

実行コマンド

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

PDF 生成

OmniCRM から PDF 生成

実行

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

□□□□

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

□□□□□□

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

session = get_db_session()

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

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

API □□□

□□/□□□□ PDF□

GET /invoice/pdf/{invoice_id}

□□□ PDF □□□□□□□□□□ Stripe □□□□□

□□□□

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



□□□□□□

□□ (crm_config.yaml)

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

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

□□□□□□□□□□

□□ (.env)

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

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

4. □□ UI □□□ Web □□□

□□□□□

□□□ npm start □□□□□□□□□□□□□□□□□□□□

□□□□□□

□□□

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


□□

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

□□

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

□□

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

□□□□

□□□□

□□□□

- □□□ □□□□□□ UI □□□
- □□□ □□□□□□□□ API/UI □□

YAML □□□□

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

□□□□□□□

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

Stripe □□□□□

- □□□ □□□□□□□□□□□□□□
- □□□ □□□□□□□□ `publishable_key` □□

□□□□□□□

- □□□ □□□ Mailjet □□□□□ ID
- □□□ □□ Mailjet API □□/□□□□□□□□ ID □□□□

PDF □□□□□

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

□□□□□□ **PDF** □□

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

UI

- CSS `display: flex; justify-content: space-between; align-items: center; width: 100%;`
- CSS `flexbox` `grid` `flexbox` `grid`
- `display: flex; justify-content: space-between; align-items: center; width: 100%;`
- `display: flex; justify-content: space-between; align-items: center; width: 100%;` `div`

API

- `invalidate_invoice_cache()`
- `invalidate_invoice_cache()`
- `Invoice_PDF_Cache` `Invoice_PDF_Cache`

API

OmniCRM `OmniCRM`

API

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

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

Dependencies

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

Assets

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

- UI `npm run build`

UI

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

Backend

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

Infrastructure

- Docker
- MySQL/PostgreSQL
- CGRates

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

API

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

OmniCRM

OmniCRM

IP GPON ONT CPE

Customer Attributes <administration_attributes>

Customer Tags <administration_tags>

CPE

inventory_items_list

UI Ansible Complete Product Lifecycle Guide - Inventory

Requirements <guide_product_lifecycle>

OmniCRM

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

SIM

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

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

SIM SIM SIM SIM SIM SIM SIM SIM SIM SIM

SIM SIM SIM SIM

GPON

GPON GPON GPON GPON GPON GPON GPON GPON

GPON GPON GPON GPON GPON GPON GPON GPON

GPON GPON GPON GPON GPON GPON GPON GPON

OmniCRM GPON GPON GPON GPON GPON GPON GPON GPON

InventoryTemplate

InventoryTemplate InventoryTemplate InventoryTemplate InventoryTemplate InventoryTemplate InventoryTemplate InventoryTemplate InventoryTemplate

- MAC itemtext1 itemtext2
-
- itemtext1 itemtext2
- IP

20 itemtext1 itemtext20

itemtext1_label itemtext20_label

itemtext1_label "MAC" itemtext2_label ""


InventoryTemplates InventoryTemplates InventoryTemplates InventoryTemplates InventoryTemplates InventoryTemplates InventoryTemplates InventoryTemplates

□□□□□□

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
 - State
 - Item - Inventory
 - Item - Service
 - Item - Customer
 - Item - Location
 - Item - Service
 - Item - Customer
 - Item - Location

Inventory Item

Inventory Item 20

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

-
-
-
-
-

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

□□□□□

□□□□□□□□□□

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

□□□□

□□□□□□□□

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

□□□□

□□□□□□

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

□□□□□□□□

API □□

□□□□□□□ API □□□□□□□□□□□□

□□□□□□

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

□□□□□

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

□□□□□□□□

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

□□□□□

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


□□□□□□

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

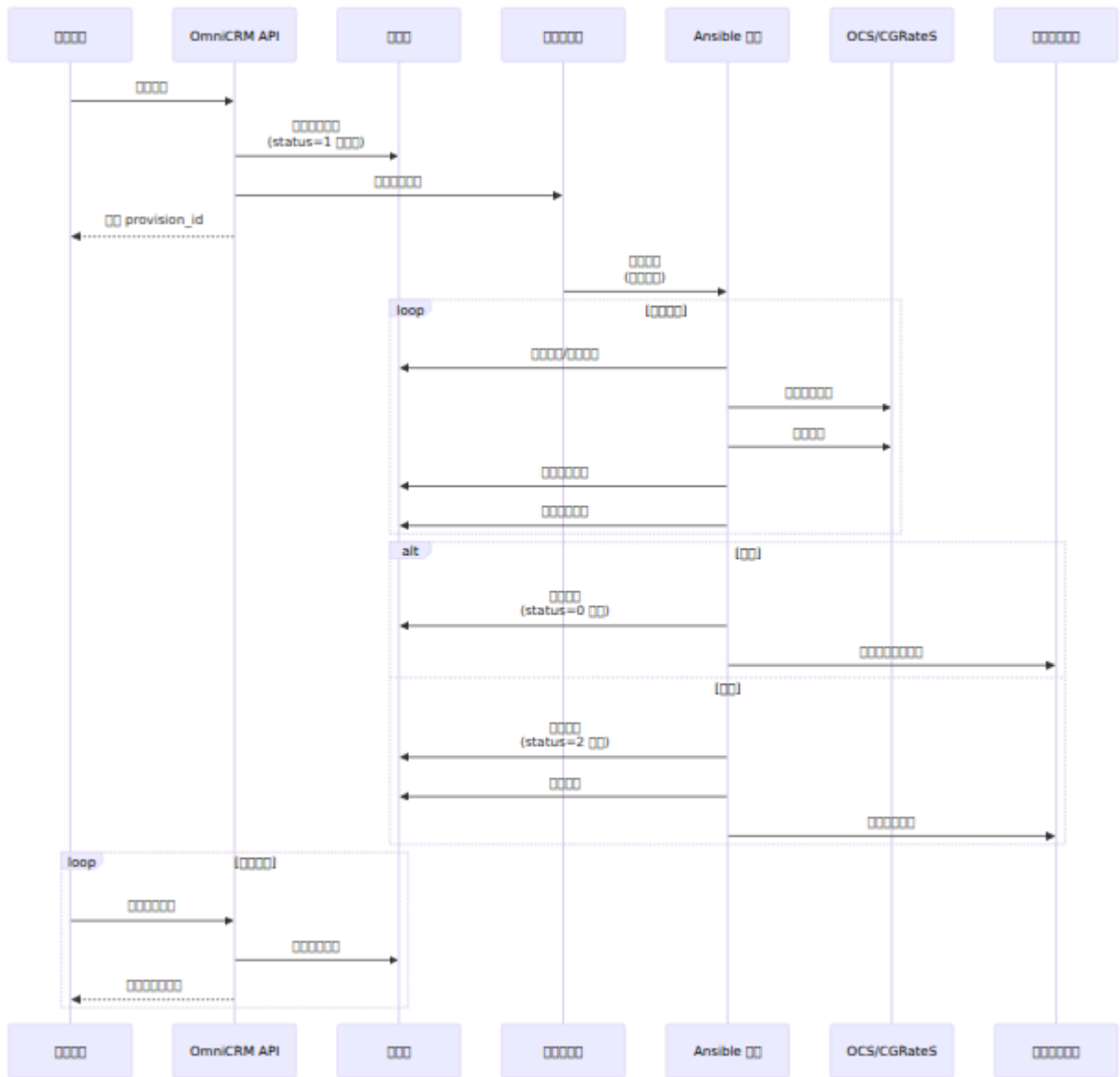
2. API □

OmniCRM API (Flask/Python)

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

□□ **API** □□:

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



5.

OmniCRM

-

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 - 一意制約

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

住所 - 住所情報

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

住所情報

00 - 0000000

00	00	00
service_id	00 (PK)	000000
customer_id	00 (FK)	000
product_id	00 (FK)	0000
site_id	00 (FK)	0000
service_name	000	0000
service_uuid	000	00000000
service_status	00	0000
service_billed	00	0000
wholesale_cost	00	000000
retail_cost	00	0000
bundled_parent	00 (FK)	00000000

00 - 0000000

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

一意

PK - 主キー

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

欄名 - データ型

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

備考

請求 - 請求明細

欄名	制約	説明
inventory_id	PK	在庫ID
customer_id	FK	顧客ID
service_id	FK	サービスID
inventory_template_id	FK	在庫テンプレートID
item		商品名 (SIM 含む)
item_state		'0', '000', '0000' など
itemtext1-20		商品テキスト
management_url		管理 URL
config_content		設定内容

在庫 - 商品マスタ

欄名	制約	説明
inventory_template_id	PK	在庫テンプレートID
item		商品名
itemtext1_label		商品テキスト1
itemtext2_label		商品テキスト2

商品

商品 - 商品マスタ

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=エラー

プロビジョニング - イベント

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=エラー
provisioning_result_json	JSON	結果JSON

プロビジョニング

プロビジョニング - イベント

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:

- [Account Management](#)
- [Billing Management](#)
- [Reporting and Analytics](#)

Stripe

Stripe

- [Account Management](#)
- [PCI Compliance](#)
- [Reporting and Analytics](#)
- [Billing Management](#)
- [Payment Methods](#)

Mailjet

- [Account Management](#)
- [Reporting and Analytics](#)
- [Campaign Management](#)
- [Template Management](#)

Google Maps

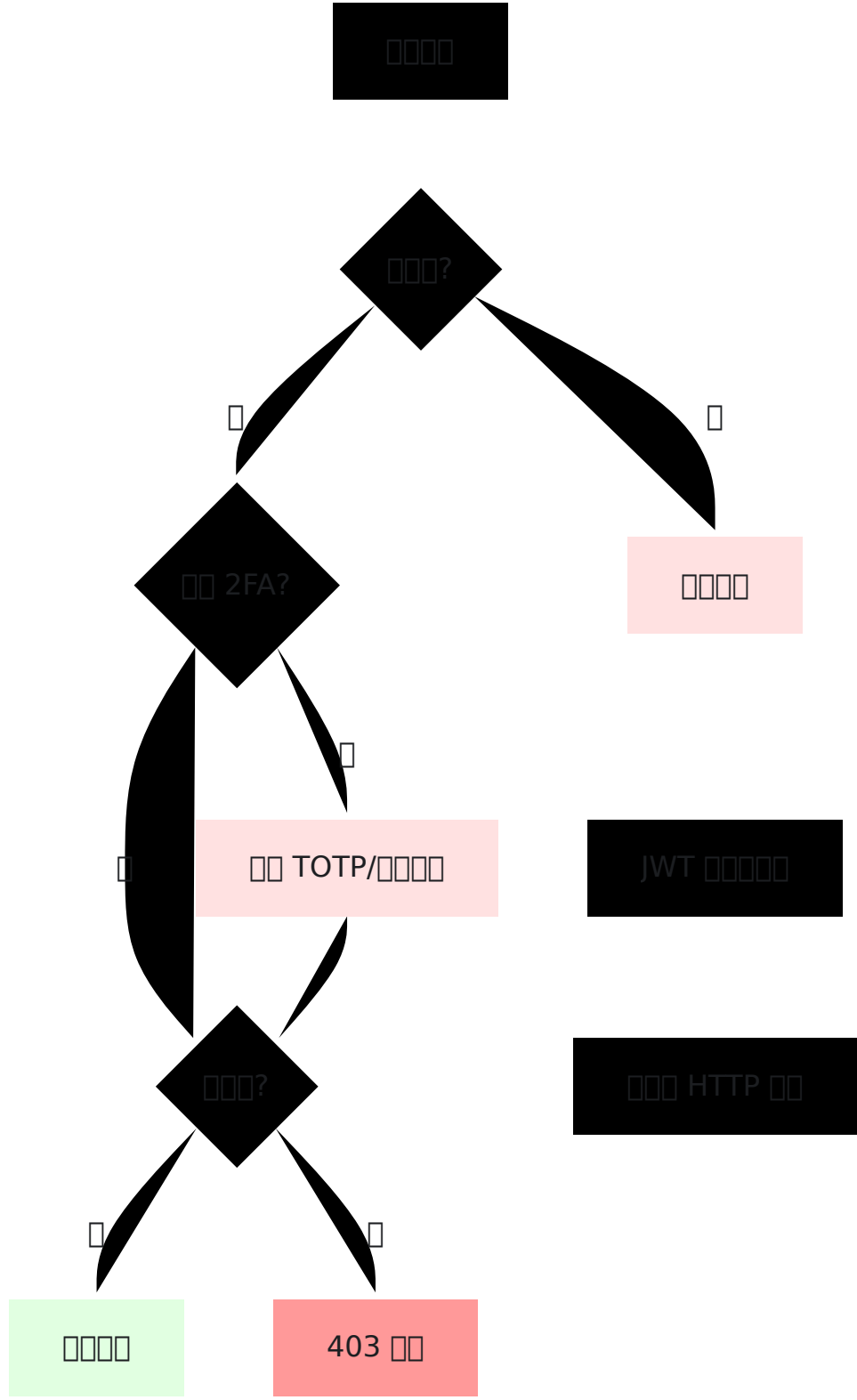
- [Account Management](#)
- [Reporting and Analytics](#)
- [Plus Code](#)
- [Maps API](#)

OCS/CGRateS

- [Account Management](#)
- [Reporting and Analytics](#)
- [Billing Management](#)
- [CDR](#)

- □□□□□□□□

□□□□

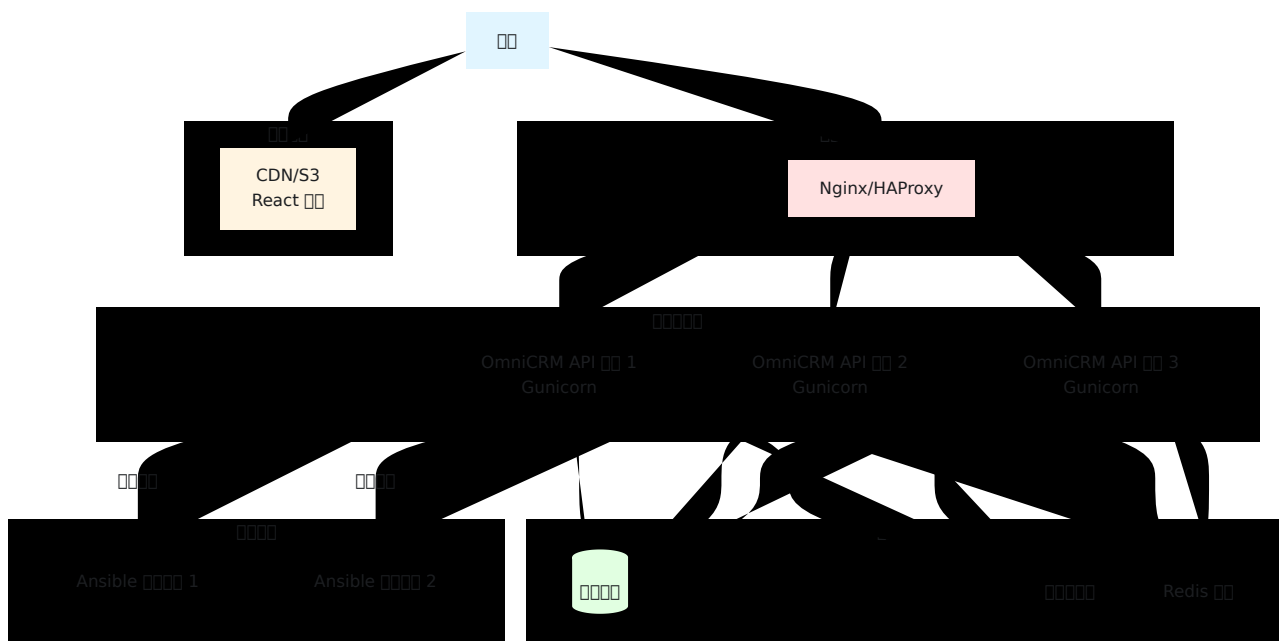


□□□□:

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

架构

部署:



技术栈

包括:

- 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:

- 数据库
- Ansible 部署
- 数据库
- 数据库

数据库

OmniCRM 数据库 Prometheus 数据库

数据库:

- API 数据库
- 数据库/数据库
- 数据库
- 数据库OCS Stripe Mailjet
- 数据库

数据库:

- 数据库Flask
- 数据库Ansible 部署
- 数据库
- 数据库

数据库: 数据库 /crm/metrics Prometheus 数据库

数据库

- Ansible 部署 - 数据库
- 数据库 - 数据库
- 数据库 - 数据库
- API 部署 - API 部署
- RBAC - 数据库

OmniCRM

OmniCRM 2FA RBAC UI

Self-Care Portal <self_care_portal> RBAC <rbac>

OmniCRM

OmniCRM

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

OmniCRM

OmniCRM

□□□□

□□□□□

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

□□□□□□□□□

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

□□□□□□

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

□□/□□□□□□

□□ □□□□ (□) □□□□□□□□□□

如何 2FA 如何

如何设置 2FA/如何设置 2FA 如何设置

如何设置

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

如何设置

如何设置

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

如何

如何 “如何” 如何

□□□□□□□□

□□□□□ REACT_APP_ALLOW_SOCIAL_LOGINS=yes □□□□□□□□□□□□

| [□ □□ Google □□] [● □□ Apple □□] [□ □□ Facebook □□]

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

□□□□□□□

□□ “□□□□□□” □□□□□□□□□□□□

□□□□□ (2FA) □□

□□□□□ 2FA □□□□□□□□□□2FA □□ TOTP□□□□□□□□□□□□□□□□□□□□□□□□□□

□□ 2FA □□

□□□□□□□□□□□□

□□□□□□□

□□□□□□□□□□ 2FA □□□□2FA □□□□□□□□□□□□□□

□ 1 □□□□□□□

| □□□□

| [□□] [□□]

□□□□□□□□□□□□□□□□ 2FA □□□□□□□□□□

□ 2 □□□□□□□

| [□□] [□□]

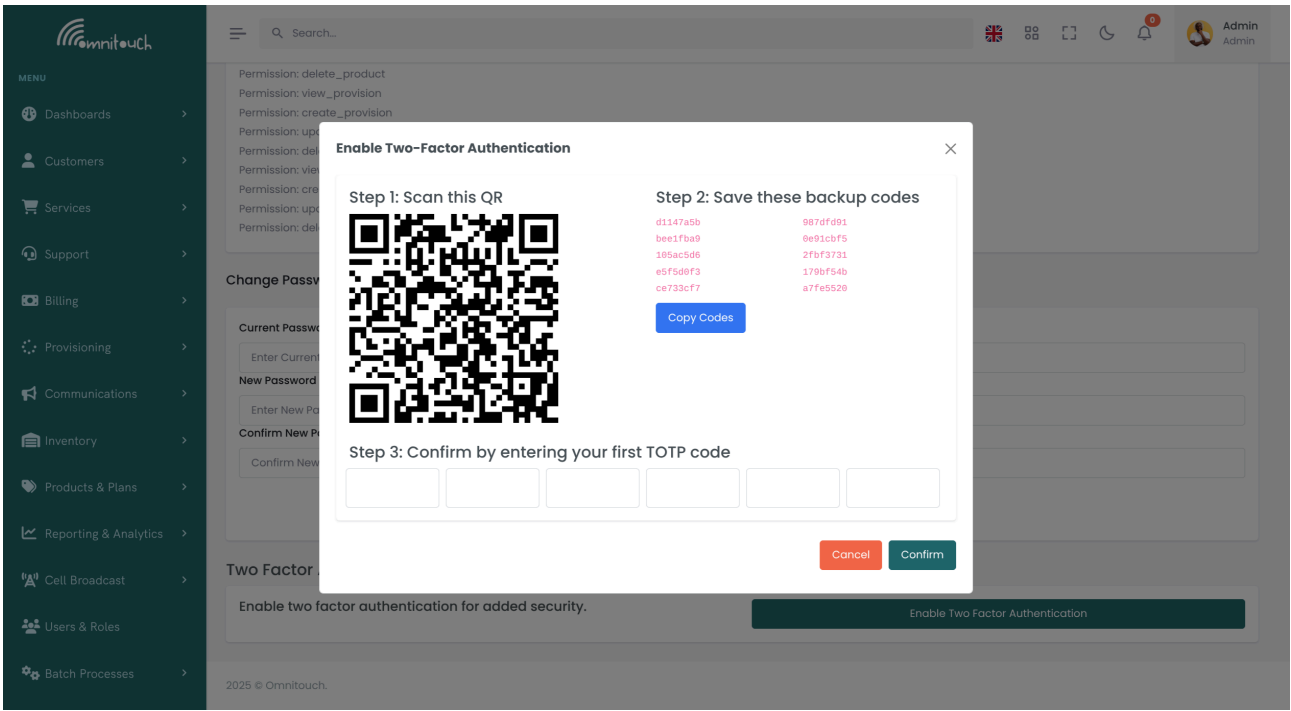
□□□□□

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

3. 2FA の設定

設定方法

設定完了後 2FA が有効



□ □ □ □ □ □ □ □ □ □ □ □

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

□ **1** □ □ □ □ □ □ □ □ □ □

□ □ □ □ □ □ □ □ "□ □ □ □ □ □"

1. 00 0000000
2. 00 “0000000”

0000000

- 00000000000000000000
- 0000000 Mailjet 0000000???
- 00000000000000000000 1 000000
- 0000000“0000000000000000”

000000000000

000000“0000000000000000”

0 2 0000000000

0000000000000000

0 [00]0

OmniCRM

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

1

3

[]

- 1.
- 2.
3. “ ”

- 8
-

- “ ”
-
-

/

XXXXXXXXXXXXXXXXXX

[XXXXXXXXXX]

XXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXX ❖❖XXXXXXXXXXXXXXXXXX

XXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

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

XXXXXX

- ➡ XX - XXXXXXXXXXXXXXXXXXXX
- XX XX - XXXXXXXX
- XX XXXX - XXXXXXXX
- XX XX **2FA** - XXXXXXX 2FAXXXXXXXX 2FA XXXXX
- ☒ XXXXXXXX - XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

XX **1** XXXXXXXXXXXXXXX (XX)

XXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

XXXXJohn Smith (<john@example.com>)

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[] []

2

“ ”

3

John Smith

[]

• • • •

[]

-
-
-
-

- 1.
2. “ ”
- 3.
- 4.

2FA

1 2FA ()

2FA

□□□□□□

□ 2 □□□□

□□ “□□ 2FA”

□ 3 □□□

□□□□

| John Smith □□□□□□□□□□□□□□□□□□□□ 2FA□

□□□□

- □□□□□□□□□□□□□□□□ 2FA □□□
- □□□□ 2FA □□□□ (□) □□□□□□□□
- □□□□□□□□□□□□□□□□ 2FA

□□□□□□□□□

□□□ 2FA □□□□□□□

1. □□□□□□□□□□□□□□□□

- □□□□□□□□
- □□□□
- □□□□□□

- 設定
2. 設定
 3. 設定 2FA

設定

設定

設定

設定 (✉) 設定 `login_count = 0`

設定 (✉)

設定

設定 Bob Wilson (<bob@example.com>)

設定 • 設定 • 設定 • 設定 • 設定

[設定] [設定]

設定 “設定”

設定

設定 **Mailjet** 設定

設定 `api_crmCommunicationUserWelcome`

設定

設定 (⇒) 設定

設定

設定

設定

□□□□

□□ admin customer_service_agent_1 customer

[□□] [□□□□]

□□□□□□

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

□□□□□□□□

□□ □□□□ (□) □□□□□

□□□□□□□□

□□□ John Smith (<john@example.com>)

⚠ □□□□□□□□□□

□□□□□□□ • □□□□□□□ • 2FA □□ • □□□□

□□□□□□□□□□□□□□

[□□] [□□□□]

□□ “□□□□” □□□□

□□□□□

□□□□□

□□□□□□□

□□□□□

- □□□□□□□□□□

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

2FA 2FA

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

2FA 2FA

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

2FA 2FA

2FA 2FA

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

2FA 2FA

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

2FA 2FA

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

認證碼

- 認證碼
- 認證碼
- 認證碼
- 認證碼
- 認證碼

認證碼

“認證碼”

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

2FA 認證碼

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

“”

- Cookies
- - Cookies
 - Cookies
 - OmniCRM
 -

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

- 1
- -
 -
 -

2FA

-
- -
 -
 -

##

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

密碼“密碼 2FA”

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

密碼密碼密碼

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

密碼密碼

密碼密碼

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

密碼密碼

- 密碼密碼密碼密碼密碼

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

2FA `auth`

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

`auth`

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

`auth`

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



0fT52ZvoZBE

XXXXXXXXXX

XX

XX/XXXX

XX
XX

XX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX

XX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXX ❖❖
 XXXXXXXXXXXXXXXXXXXXXXXXXXXX
 XXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXX

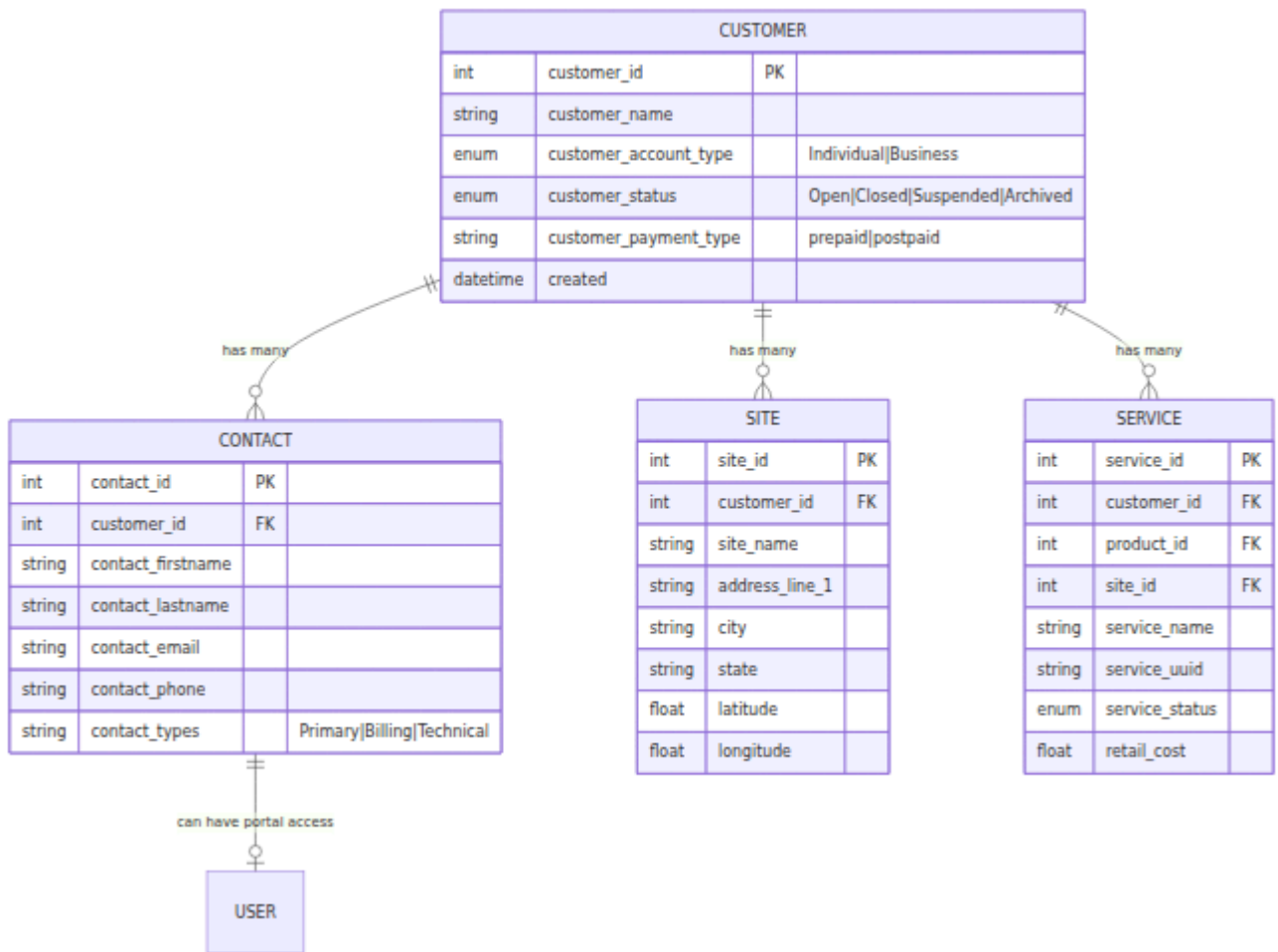
XXXXXXXXXXXXXXXXXXXX

XXXXXX

OmniCRMXX

XXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

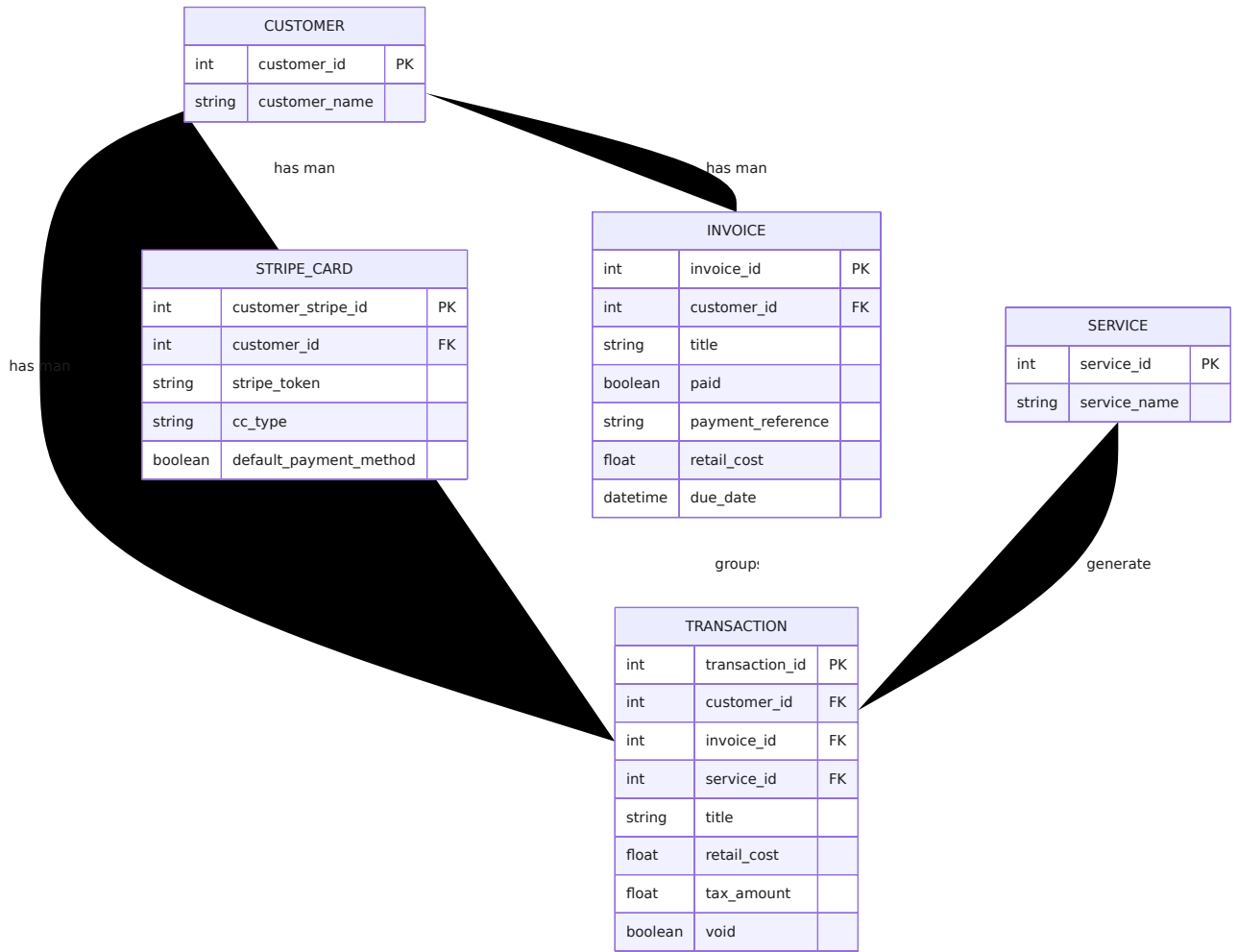


□□□□

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

□□□□□□□□

□□□□□□□□□□□□□□

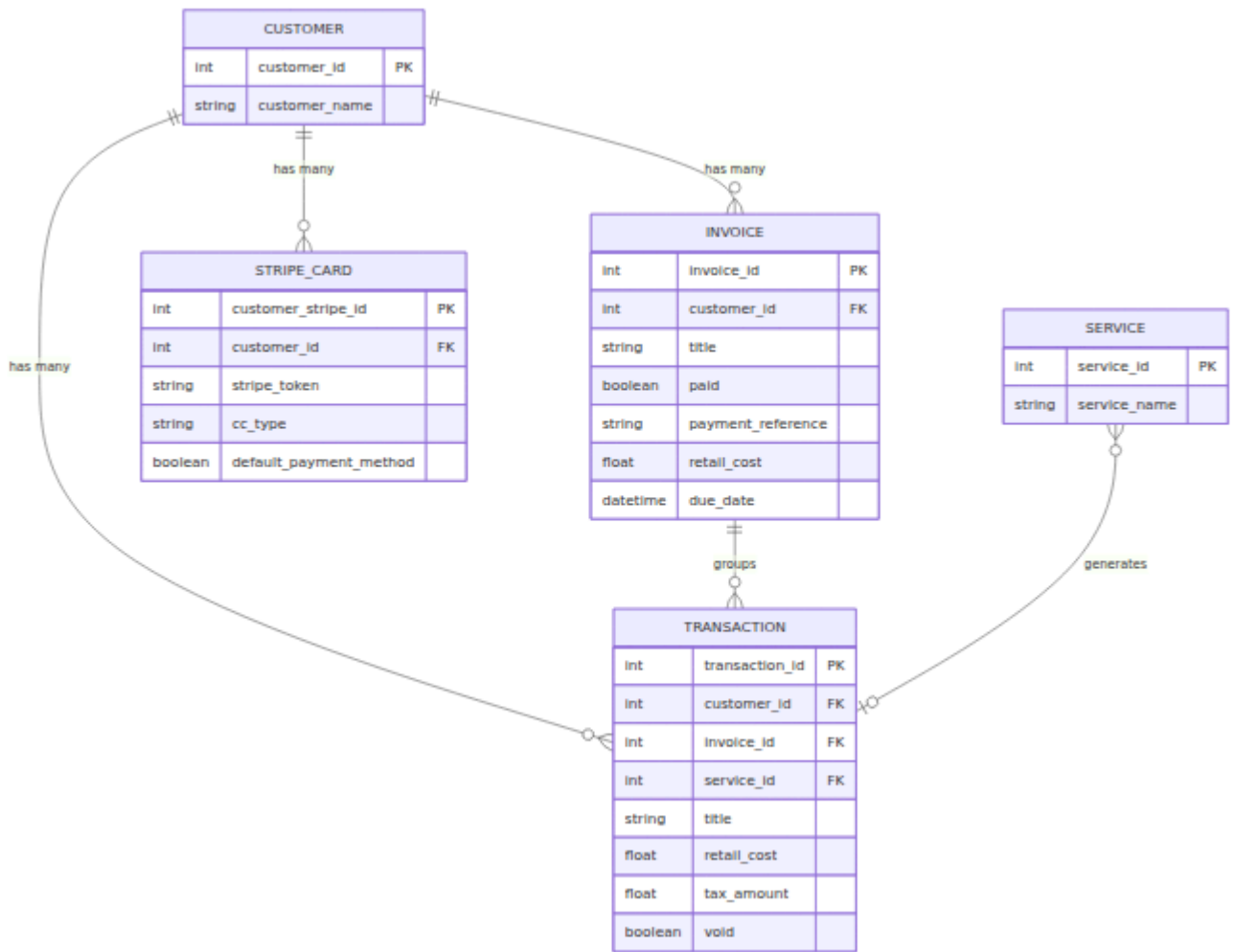


□□□□

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

□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□

- 國家2 - 國家代碼/國際代碼
- 城市 - 城市
- 州/省 - 州/省
- 郵政編碼 - 郵政編碼
- 城市 - 郵政編碼
- 國家代碼 - 國家代碼
- **Plus**代碼 - 11位數字國家代碼 8C3MFJV8+2F
- **Google Maps**代碼ID - 國家代碼

2. 國家代碼

- 國家代碼
- 國家代碼
- 國家代碼
- 國家代碼

Plus代碼

Plus代碼是國際電話號碼/國際代碼Plus代碼11位數字

代碼

國家代碼 ✓ 城市“123 Main Street” ✓ 城市1“123 Main Street” ✓ 州
 “London” ✓ 州“Greater London” ✓ 國家“United Kingdom” ✓ 郵政編碼“SW1A
 1AA” ✓ 郵政編碼51.5074 ✓ 郵政編碼-0.1278 ✓ Plus代碼“9C3XGPHC+3Q” ✓ Google代碼
 ID“ChIjdd4hrwug2EcRmSrV3Vo6III”

代碼

代碼

- 代碼
- 代碼 REACT_APP_DEFAULT_COUNTRY
- **Plus**代碼 12位數字 11 + 1位數字

代碼

代碼

代碼

□

□□□□□□□□

□□□□□□□□□□□□□□Plus□□□□OmniCRM□□□□

1. □□□□□□

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

2. □□□□

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

□□□

```

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

```

□□□47□□□□□□□□ □□□□47□□□□□□□□□□

3. □□□□□□

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

4. 📍📍📍□□□□□□

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



OmniCRM □□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

WebApp Web

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□

□□□

□□□

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

□□□□

□□□□□□□□□□□□

□□□□□□

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

□□□□□□

1. □□□ □□ → □□□□□□□□
2. □□□□□□□□□□

□□□□□□□□

□□□□□□□□□□◆◆◆□□□□□□□□

□□□□□□

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

□□□□□□

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

□□□□

□□□□□□□□□□□□□□□□

□□□□

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

□□□□□

□□□□□□□□□□□□ : □□□□□□□□

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

□□□□

□□□□□□□□

□□□□□□□□□□

□□□□□□□□□□□□

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

- 0000 - 0000000000

00000000/0000

00000000000000

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

00000000

00 "+ 0000" 000000000000

0000000000

00000000/0000

□□□□

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

□□□

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

□□□□□□

1. □□□□□□□□□□
2. □□□□□□□□□□

3. 0000“000000”0000
4. 000000000000
5. 0000000000

0000000000

00

00000000000000000000

- 00ID
- 00
- 00
- 00ID

0000

0000000000000000

00000000

- 00000 - 000000000000
- 00000 - 000000000000

000000

- 000000 - 00000000000000
- 000000 - 000000000000000000

00

0000000000000000

- **ID** - 0000ID000000/0000
- 00 - 0000000000
- 00 - 0000000000

- 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“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 \$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%
- "-"

111111

1111111111

1111111111

111111

- 1111111111
- 1111111111
- 1111
- "111111"11

- 00000000-0

000000

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

00000

000000000000

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

000 payments_invoices 0000000000000000

00000000

00000**1**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

Unit 4 Reading and Writing

1. Warm-up
2. Read the text and complete the tasks
3. Pre-reading
4. Reading
5. Post-reading
6. Speaking
7. Writing
8. Revision

Reading

Text 1

- Before you read, think about the main ideas of the text
- Read the text and complete the tasks

Text 2

- Before you read, think about the main ideas of the text
- Read the text and complete the tasks

Text 3

- Before you read, think about the main ideas of the text
- Read the text and complete the tasks

Text 4

- Before you read, think about the main ideas of the text
- Read the text and complete the tasks

Text 5

- Before you read, think about the main ideas of the text
- Read the text and complete the tasks

□□□□

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

OmniCRM API

OmniCRM API HTML templates using Jinja2

Invoice Templates

OmniCRM API invoice templates/

Files

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

Config

OmniCRM API crm_config.yaml

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

Invoice Data

Jinja2 template

Variables

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

Context

- {{ client.name }} - Client name

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

000000

0000000000000000

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

00000

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

000000000

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

□□□□□□□□

□□**1**□□□□□□

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

□□**2**□□□□**HTML/CSS**

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

□□□□□□□□

1. □□□□□□□□

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

<!-- □□□□□□□□ -->
<h1>□□□□□□□□</h1>
```

2. □□□□

```

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

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

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

```

3. Footer

```

<footer>
  <p>Footer text</p>
  <p>12345678901234567890</p>
  <p>Phone: +1-555-123-4567 | Email: billing@yourcompany.com</p>
  <p>ABN/12345678900</p>
</footer>

```

4. Links

```

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

```

5. Terms

```

<div class="terms">
  <h4>Terms</h4>
  <p>Payment is due within 30 days of invoice.</p>
  <p>Interest is charged at 2% per month.</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. Restart API

```

cd OmniCRM-API
sudo systemctl restart omnicrm-api

```

5. Deployment

1. Create deployment package
2. Deploy to production

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

```

```

<!-- 结账金额 -->
<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</strong></td>
  <td>${{ "%.2f"|format(total_amount + total_tax) }}</td>
</tr>

```

total_tax 结账金额 tax_amount 税金
 tax_percentage 税率 tax_percentage 0%

结账金额

税金

```

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

```

PDF 打印

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

CSS

- CSS 2.1
- CSS3 flexbox
- @font-face

打印

- JavaScript
- CSS Grid
-
- CSS

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

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

```

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

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

```

□□□□□

□□□□□□□◊◊□□□□□□□□□□□□

```

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

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

```

□□□□□□

□□□□□

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

- 文字列
- 数値

3. PDF

- 最大サイズ<5MB
- 文字
- 画像
- 埋め込み

4. 文字列

- 文字列
- 数値
- 日付

5. 埋め込み

- PDF
- Mailjet 15MB
- Gmail Outlook Apple Mail

API

API

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

PDF

PDF

- 文字列
- 数値

PDF

- CSS
- CSS 2.1 WeasyPrint CSS

2. 刪除 - 刪除所有已過期的
 - 刪除invoice_id對應的PDF
 - 刪除所有已過期的PDF
 - 增加 `last_accessed` 欄位
3. 刪除 - 刪除所有已過期的PDF
 - 刪除所有已過期的PDF
 - 刪除所有已過期的PDF
 - 刪除所有已過期的PDF

刪除

- 刪除 - 刪除所有已過期的PDF
- 刪除 - 刪除所有已過期的PDF
- 刪除 - 刪除所有已過期的PDF
- 刪除 - 刪除所有已過期的PDF

刪除

刪除PDF

- 刪除所有已過期的90天
- 刪除所有已過期的30天
- 刪除所有已過期的PDF

API

API UI

- 刪除所有PDF
- 刪除所有PDF
- 刪除所有PDF

刪除

```
-- 刪除所有PDF
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

- 1000000 - 1000000000000000000000
- 1000000 - 1000000000000000000000
- 1000000 - 1000000000000000000000

1000000

- 1,000 "k" \$1.5k
- 1,000,000 "M" \$2.3M
- 1,000,000,000 "B" \$1.1B

1000000

- "100" "100" 1000000000000
- 1000000000000000000000
- 1000000000000000000000
- 1000000000000000000000

100000

1000000000000000000000

10000

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



































□□□□

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

















□□□□

□□ "+ □□□□" □□□□





















□□□□

-  -  -  -  -  -  -  -  -  -  -  -  - 
-  - / “ []”
-  -  14
-  -  
-  -  
-  -  /

- ✓ () -  
- × () -  
-  -  
-  - 
-   
- /

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

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□

- □□□□ - □□□□ID□□□□□□□□□□□□□□□□□□
- □□□□ - □□□□□□□□□□□□□□□□□□□□
 - □□□□
 - □□□□□□
 - □□□□
 - □□□□□□□□□□ \$10.00 (10%) □
 - □□□□□□□□□□ “-”
- □□□□ - □□□□□□□□□□
 - □□□□
 - □□□□□□□□□□□□□□
 - □□□□□□□□□□
 - □□□□□□□□ + □□
- □□□□ - □□□□□□□□□□
 - □□PDF - □□□□PDF□□□□□□□□□□
 - □□□□□□□□ - □□□□□□□□□□□□□□□□□□□□□□
 - □□□□ - □□□□□□□□□□□□□□□□□□□□
 - □□ - □□Stripe□□□□□□□□□□□□□□□□Stripe□□□□

- 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
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

目的

- 目的を達成する
- 目的を達成するための計画を立てる

計画

1. 目的を達成するための計画を立てる
2. 目的を達成するための計画を実行する
3. 目的を達成するための計画を評価する

結論

- 目的を達成するために `void = true`
- 目的を達成するために `invoice_id` を `null`
- 目的を達成するために “ ”
- 目的を達成するために
- 目的を達成するために “:”
- 目的を達成するために
- 目的を達成するために “ ”

まとめ

- 目的を達成する
- 目的 = “目的を達成するための計画”

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

退款

如何退款

退款

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

退款

1. 使用 `Stripe`
2. 退款
3. 退款

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

7. 環境構築

環境構築

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

環境構築

- POS
- 環境構築
- 環境構築

環境構築

環境構築

環境構築

- ID
- 環境構築
- 環境構築

環境構築

環境構築

環境構築

- 環境構築 - 環境構築
- 環境構築 - 環境構築

環境構築

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. 00000000000000000000
 - o 0000 "+ 0000000"
 - o 000"00000 - 2025010"
 - o 00/00010100310000000000
 - o 0000000000
 - 00000000000000000000
 - 00 X 0000000000
 - 00000000000000000000
 - o 00 "0000"

5. 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
 - 〇〇〇〇 "+ 〇〇〇〇〇〇"
 - 〇〇〇〇"〇〇〇〇 - 2025〇1〇"
 - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
 - 〇〇 "〇〇" 〇〇〇〇〇〇
 - 〇〇 "〇〇〇〇"
6. 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇

〇〇〇〇**3**〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇

1. 〇〇〇〇〇〇〇
2. 〇〇〇〇〇〇 〇〇 〇〇
3. 〇〇 "+ 〇〇〇〇〇〇"
4. 〇〇〇〇〇〇〇〇〇
5. 〇〇〇〇〇〇〇20〇〇〇〇
6. 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
7. 〇〇〇〇〇〇〇〇〇
 - 〇〇〇〇〇〇〇〇〇〇〇〇"〇〇〇〇〇〇〇"〇
 - 〇〇 **X** 〇〇〇〇〇〇〇
 - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇"〇〇〇〇〇"〇
 - 〇〇 **X** 〇〇〇〇〇〇〇
 - 〇〇〇〇〇〇〇〇〇〇〇〇〇"〇〇〇〇18〇〇〇〇"
 - 〇〇〇〇〇〇〇〇〇〇〇〇
8. 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
9. 〇〇 "〇〇〇〇〇**18**〇"
10. 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
11. 〇〇/〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇

〇〇〇〇**4**〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇

1. 〇〇〇〇〇〇〇〇〇〇
2. 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
3. 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
 - 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇

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

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

XXXX**5**XXXXXXXXXXXX

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

XXXX**6**XXXXXXXXXXXXXXXXXXXX

1. **XXXXXXXXXXXX**
2. **XXXXXXXXXXXXXXXXXXXX**
3. **XXXXXX XXXXXXXXXXXXXXX**
4. **XXXXXXXXXX**
 - **IDXXXXXXXXXX**

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

0000**7**0000000000

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

0000**8**0000000000

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

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

9. Stripe API

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

PDF

PDF - PDF

- PDF
- PDF

PDF

- WeasyPrint
- HTML API

Stripe API

- Stripe API
- Stripe API

Stripe

- Stripe API
- Stripe API

Mailjet

- Mailjet API
- Mailjet API `crm_config.yaml`

Stripe

- Stripe API/POS
- Stripe API

PDF

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

Stripe

- Stripe API
- Stripe API

Stripe

- Stripe API
- Stripe API

Stripe

- Stripe API
- Stripe API

Stripe

- Stripe API
- Stripe API

API Security

- API Security
- API Security

API Security

- API Security
- API Security

API Security

- API JavaScript
- API Security

API Security

- API Security 0% null
- API Security

API Security

- API Security
- API Security PDF

API Security

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

Table of Contents

OmniCRM Stripe PayPal OmniCRM

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

Table of Contents

Table of Contents

- Stripe PayPal OmniCRM
- Stripe PayPal
-
-
-
- <self_care_portal>
-

Table of Contents

- Visa Mastercard American Express Discover
-
-

OmniCRM

OmniCRM

- stripe paypal
- Visa Mastercard
- 4
-
-
-

□□□□□□

□□□□□□

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

□□□□

□□□□□□□□□□

□□□□□□□□□□□□

□□□□□□□□□□□□□□60□□□□□□□□□□

□□□□□□

□□□□□□□□□□□□

□□□□

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

- 00 - 0000000000
- 00 - 00000000000000000000

00000000

000000000000 : 00000000

- 0000 - 000000000000
- 00 - 00000000

00000000

00 "000000" 000000000000

00**1**000000000000

000000000000 Stripe Elements PayPal SDK 00000000

000000

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

Overview

- Introduction - Project Goals

Requirements

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

2. Implementation

2.1. "Payment" Module

1. Frontend
 - Card number input
 - Expiry date input
 - CVC input
2. Backend
 - Connect to OmniCRM API
 - Connect to Stripe API (pm_1A2B3C4D)
 - Connect to OmniCRM
3. Security
 - OmniCRM integration
 - 4-digit PIN verification
 - OmniCRM integration

3. Testing

3.1. Unit Tests

Unit tests for Visa card number validation (1234)

3.2. Integration Tests

3.3. Security Tests

- 00000000000000000000
- 00000000000000000000
- 0000000000000000

0000000000

0000000000

- 00000000
- 0000
- 00000
- 0000000000000000

00000000

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

Visa00000005678000000000000000

0000000000000000

000000

0000000000000000000000000000

00000000

00000000000000000000

XXXXXXXXXX

XXXX XX → XXXXXXX XXXXXXXXXXXXX

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 - XXXX30/60/90XXXX
- XXXX - XXXX
- XXXX - XXXXXXXXXXXXX

XXX

- XX - XXXXXXXXXXXXXXXXX
- XX - XXXXXXXXXXXXXXXXXMailjet

XXXX

XXXXMailjetXXXXXXXXXX

- XXX**60** - XXXX

- 30日 - 1ヶ月
- 7日 - 1週間
- 1日 - 1日

1. 1ヶ月 / 7日 / 1日

2. 1ヶ月 / 7日 / 1日

Mailjet

- 1ヶ月
- 7日 / 4週間
- 1日
- 1ヶ月 / 7日 / 1日

3. integrations_mailjet

4. 1ヶ月 / 7日 / 1日

5. 1ヶ月 / 7日 / 1日

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

6. 1ヶ月 / 7日 / 1日

7. 1ヶ月 / 7日 / 1日

1. 1ヶ月 / 7日 / 1日
2. 1ヶ月 / 7日 / 1日 → 1ヶ月
3. 1ヶ月 / 7日 / 1日 / 1ヶ月 / 7日 / 1日
4. 1ヶ月 / 7日 / 1日

5. 環境問題

6. 社会問題

問

環境問題は、人類の生存と発展にとって重要な課題である。

- 環境汚染の防止
- 資源の持続可能な利用
- 気候変動の緩和

環境問題の解決には、政府、企業、市民の協力が不可欠である。

環境問題の解決には、持続可能な開発目標（SDGs）が指針となる。

問

1. 環境問題の種類

- 大気汚染
- 水質汚染
- 土壌汚染
- 森林破壊

2. 環境問題の原因

- 人口増加
- 産業革命
- 化石燃料の大量消費

3. 環境問題の影響

- 健康被害 - 呼吸器疾患の増加
- 生態系破壊 - 生物多様性の喪失
- 気候変動 - 異常気象の頻発

問

1-3環境問題の種類

- 大気汚染
- 水質汚染

- 〇〇〇〇〇〇〇〇〇〇〇〇〇〇

4-7

- 〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- 〇〇〇〇〇〇〇〇〇〇
- 〇〇〇〇〇〇〇〇〇〇〇〇

8-14

- 〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- 〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇

15

- 〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- 〇〇〇SIM〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- 〇〇〇〇〇〇〇
- 〇〇〇〇〇〇〇〇〇〇〇〇〇〇

〇〇〇〇〇〇〇

〇〇〇〇〇〇〇〇〇

- 〇〇〇〇30〇〇〇〇〇
- 〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- 〇〇〇〇〇〇〇〇〇〇
- 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇

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. 〇〇〇〇〇〇〇〇〇〇〇〇〇〇
2. 〇〇〇〇〇〇〇
3. 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

5. 銀行振込

銀行振込

銀行

OmniCRM銀行振込

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

OmniCRM銀行振込

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

OmniCRM銀行振込

- 銀行
- CVV/CVC
- 銀行
- PIN

PCI

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

<payment_system_guide>

1

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

2

Payment Methods

Payment methods are used to process payments from customers.

Payment methods include `<payment_methods>`, `<payments_transaction>`, `<payments_invoices>`, and `<payments_process>`.

Payment Methods

Payment methods are used to process payments from customers.

1. Add payment method → **[Payment Methods]**
2. Select payment method
3. Payment method details

Payment methods are used to process payments from customers.

Payment methods are used to process payments from customers.

- Add payment method → `<add_payment_method>`
- Update payment method → `<update_payment_method>`

Payment methods are used to process payments from customers.

Payment methods are used to process payments from customers. `<self_care_portal>` is used to process payments from customers.

- Payment methods are used to process payments from customers.
- Payment methods are used to process payments from customers.
- Payment methods are used to process payments from customers.

Payment Methods

Payment methods are used to process payments from customers.

□□ **1**□□□□□□

◆◆ □□ □□□□□□□□□□

□□□□□

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

📁 basics_payment

📄 2📄

📄 📄

📄

- 📄
- 📄/📄
- 📄
- 📄
- 📄

📁 payments_transaction

📄 3📄

📄 📄

📄

- 📄
- 📄/📄
- 📄
- 📄 PDF
- 📄
- 📄Stripe 📄

- payments_invoices

payments_invoices

payments_invoices

payments_invoices

payments

payments → invoices

1. payments

payments

- invoice_id
- invoice_id
- invoice_id null

payments

2. 查詢

查詢

- 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“#1234 查詢”
- 查詢 -\$45.00 查詢
- 查詢 `invoice_id` 查詢

查詢

3. 設定

- 設定項目
- 設定項目

設定 → 設定

Stripe 設定

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

設定

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

設定項目

設定項目

設定 **1** 設定項目

設定 設定項目

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

- 詳細な説明
2. 詳細
- 詳細説明
 - 詳細説明\$50.00
 - 詳細説明\$45.00
 - 詳細説明\$95.00
3. 詳細
- 詳細 → 詳細 "詳細"
 - 詳細説明
 - 詳細 "詳細"
 - 詳細 #INV-2025-001234 詳細 \$95.00
4. 詳細
- 詳細説明 #INV-2025-001234
 - 詳細説明 \$0.00
5. 詳細説明
- 詳細説明
 - 詳細説明 PDF 詳細説明
6. 詳細説明
- 詳細説明
 - 詳細説明
 - 詳細 "詳細" 詳細
 - 詳細説明
 - 詳細 "詳細"
 - Stripe 詳細
7. 詳細
- 詳細説明 "詳細"
 - 詳細説明 \$95.00

詳細 詳細説明

詳細 2 詳細説明

詳細 詳細説明



1. 詳細説明

- 00000103100
- 00000000000000000000
- 00 000000000000

2. 00000000

- 000 00 00
- 000 00000000
- 00000000000000
- 000000000000

3. 00000

- 000 00 → 00000000
- 00   000000000000
 - 00 "000000"
 - 0000
 - 000002025-01-01
 - 000002025-01-31
 - 000002025-02-15
 - 00 "0000"

4. 00000

- 000000000000
- 00000 00000 \$0.00

5. 000000000000

- 0000000000000000
- 000000000000

6. 00000

- 0000000000000000000000
- 0000000000000000/POS 00
- 00000 000000000000

000 00000100000000000000000000

00000 **3**000000000000

000 0000000000000000000000

1. 00000000

- 2 ວັນ
- \$10 ວັນ

2. ການຈັດລຳດັບ

- ຈັດລຳດັບ → ຈັດລຳດັບ → ຈັດລຳດັບ
- ຈັດລຳດັບ "+ ຈັດລຳດັບ"
- ຈັດລຳດັບ ຈັດລຳດັບ
- ຈັດລຳດັບ ຈັດລຳດັບ
- ຈັດລຳດັບ "ຈັດລຳດັບ"
- ຈັດລຳດັບ "ຈັດລຳດັບ 1 ຈັດລຳດັບ 8-9 ຈັດລຳດັບ 2 ຈັດລຳດັບ"
- ຈັດລຳດັບ 10.00
- ຈັດລຳດັບ "ຈັດລຳດັບ"

3. ການກຳນົດ

- ການກຳນົດ ການກຳນົດ **-\$10.00**
- ການກຳນົດ ການກຳນົດ
- ການກຳນົດ ການກຳນົດ **-\$10.00**

4. ການກຳນົດ

- ການກຳນົດ ການກຳນົດ
 - ການກຳນົດ ການກຳນົດ
 - ການກຳນົດ ການກຳນົດ
- ການກຳນົດ ການກຳນົດ
 - ຈັດລຳດັບ → ຈັດລຳດັບ "ຈັດລຳດັບ"
 - ການກຳນົດ ການກຳນົດ
 - ການກຳນົດ ການກຳນົດ

5. ການກຳນົດ

- ການກຳນົດ ການກຳນົດ \$35.00
- ການກຳນົດ ການກຳນົດ

ການກຳນົດການກຳນົດການກຳນົດການກຳນົດການກຳນົດການກຳນົດ

ການກຳນົດ **4** ການກຳນົດການກຳນົດ - ການກຳນົດ

ການກຳນົດການກຳນົດການກຳນົດ - ການກຳນົດການກຳນົດ

1. ການກຳນົດ

- ການກຳນົດການກຳນົດ

- Stripe 查詢 "ID"

- 查詢 ID 的資料

2. 查詢 ID

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

3. 查詢 ID

- 查詢 "ID" 的資料
- 查詢 ID 的資料

4. 查詢 ID

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

5. 查詢 ID

- 查詢 ID "ID"
- `payment_reference` 查詢 Stripe 查詢 ID

查詢 ID 的資料

查詢 ID 5 查詢 ID 的資料

查詢 ID 的資料 - 查詢 ID 的資料

1. 查詢 ID

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

2. 查詢 ID

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

- o 文档 "文档"
- o 文档 #1234 文档 "文档"

4. 文档

- o 文档 #1235 文档
 - 文档“文档 2025-02-10 - 文档 #001”
- o 文档 #1236 文档
 - 文档“文档 2025-02-10 - 文档 #001”

5. 文档

- o 文档 文档
- o 文档
- o 文档

6. 文档

- o 文档 文档 → 文档 文档 \$300.00
- o 文档 "文档" 文档

文档 文档

文档

文档

文档

- 文档
- 文档
- 文档
- 文档



文档

- 文档 100
- 文档
- 文档
- 文档
- 文档 PDF

□□□□

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

□□□□

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

□□□□□□□□□□□□□□

□□□□

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

□□□□

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

□□□□

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

□□□□□

□□□□□

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

□□□□□□□

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

□□□□□

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

□□□□□□□□□□□

□□□□□□□□□□□□□□

□□□

- □□□□□□□□□□□□□□
- □□□□□□“□□□□□□□□□□”

□□□

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

□□□□

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

□□□□□□□□□□□□□□

□□□

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

□□□


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

□□□□□

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

□□□□□□□□□□□□□□

□□□

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

□□□

1. 在 Stripe 网站上注册
2. 验证身份
3. 添加支付方式 - 信用卡/借记卡
4. 设置 Stripe 账户

Stripe 账户

- Stripe 账户类型：个人、企业、非营利组织
- Stripe 账户验证：提供身份证明、地址证明、银行对账单
- Stripe 账户设置：设置 Stripe 账户名称、地址、联系方式

Stripe 支付网关

Stripe

- Stripe 支付网关“Stripe”
- Stripe 支付网关
- Stripe 支付网关

Stripe

1. 在 Stripe 网站上注册
2. 验证身份
 - 身份证明
 - 地址证明
 - 银行对账单/支票
 - 联系方式
 - 其他证明

Stripe 支付网关

- Stripe 支付网关
- Stripe 支付网关
- Stripe 支付网关 POS 支付网关
- Stripe 支付网关

□□ ? ? ? □□□

□□□

- □□□□□□□□□□□□ → **B** □□□□□□□□□□□□

□□□

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

□□□□□

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

□□□□□

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

□□□□□

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

CGRateS

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

API

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

CGRateS provides a set of REST API endpoints for interacting with the system. The endpoints are organized into Playbooks - a collection of related endpoints.

Endpoints

CDR - Call Detail Record

Account - Account Information

ID - Account ID

Account - Account Information

Account - Account Information (+5)

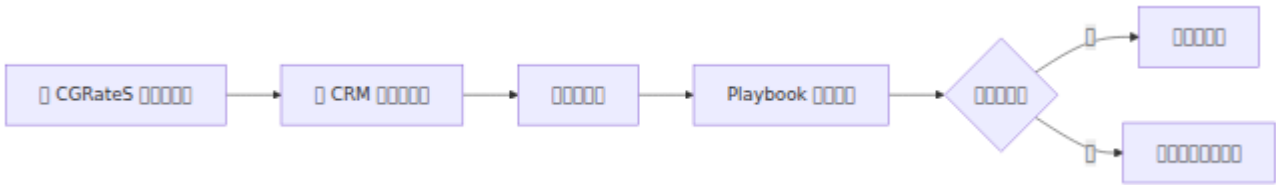
Account - Account Information 

API Endpoints

ExecuteAction **Playbook** is a REST API endpoint that allows you to execute a playbook in CGRateS.

API Endpoints

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



□□□□□□□□□□

□□□□□□□□□□□□□□

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

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

□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

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

□□□□□□□□□□□□□□□□

```

import cgrateshttpapi

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

# 1. Create CGRateS object
Action_50GB_Data_Pack = {
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_50gb-data-pack",
        "Tenant": tenant,
        "Actions": [
            {
                "Identifier": "*topup",
                "BalanceType": "*data",
                "Units": 50 * 1024 * 1024 * 1024,
                "ExpiryTime": "+720h",
                "Weight": 10
            }
        ]
    }]
}

result = OCS_Obj.SendData(Action_50GB_Data_Pack)
assert result['error'] is None or result['error'] == "EXISTS"

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

```

XXXXXXXXXXXXXXXXXX

```

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

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

```

-
-
-

#####

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

#####CGRateS ##### - #####
#####

*topup [] *topup_reset

[] ID

*topup -

*topup [] #####

#####

- ##### ID #####
- #####
- #####
- #####

□□□

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

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

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

□□□

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

*topup_reset - □□□□

*topup_reset □□ □□□□□□□□□□□□□□ ID□

□□□

- □□□□□□ ID □□□□□□

- 0000000000000000
- 0000000000
- 0000000000000000

000

```
# 000000
000
ID: "Data_Package__5368709120"
0: 1073741824 # 00 1GB
00: 2024-12-240000 1 00

# 00 *topup_reset 000
000
000: "*topup_reset"
000
ID: "Data_Package__5368709120" # 00 ID - 0000
0: 5368709120 # 5GB
ExpiryTime: "+5d"

# *topup_reset 000000
000
ID: "Data_Package__5368709120"
0: 5368709120 # 5GB0000 1GB 0000
OriginalValue: 5368709120
Value_hr: "5 GB"
Remaining_hr: "5 GB of 5 GB"
PercentUsed: 0
00: 2024-12-29000000 5 00
```

000

- 000000000000000000000000
- 000000000000000000000000
- 000000000000000000000000
- 000000000000000000000000

顧客 ID 取得

顧客 ID 取得

顧客 ID 取得

OmniCRM 顧客 ID 取得 API Web UI

顧客 ID

```
{顧客ID}__{顧客ID}
```

顧客 ID

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

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

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

顧客 ID 取得 - 顧客 ID 顧客 ID API 顧客 ID

API 顧客 ID

OmniCRM API CGRateS 顧客 ID 顧客 ID _hr 顧客 ID

```

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

        // 日本語コメント
        "ID_hr": "AU Data Domestic",
        "OriginalValue": 107374182400,
        "OriginalValue_hr": "100 GB",
        "Value_hr": "50 GB",
        "Remaining_hr": "50 GB of 100 GB",
        "PercentUsed": 50,
        "ExpiryTime_hr": "2025年1月25日22時"
      }
    ]
  }
}

```

API 実装

1. ID 抽出

```

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. 日本語コメント

```

# 日本語コメント
id_hr = descriptive_name.replace("_", " ") # "AU Data
Domestic"

```

3. 〇〇〇〇〇〇〇〇〇〇〇〇

```

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

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

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

```

4. 〇〇〇〇〇〇〇〇〇

```

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

```

5. 〇〇〇〇〇〇〇〇〇

```

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

```

Web UI 〇〇〇

〇〇〇〇〇〇 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇

```

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

// 00000000
<BalanceCard>
  <Title>{balance.ID_hr}</Title>           {/* "AU Data Domestic"
*/}
  <Progress value={balance.PercentUsed}> {/* 50% */}
    {balance.Remaining_hr}             {/* "50 GB of 100 GB"
*/}
  </Progress>
  <Expiry>{balance.ExpiryTime_hr}</Expiry> {/* "20250125002200"
*/}
</BalanceCard>

```

00000000

1. 000000 - 0000000000 UI 000000 "50 GB of 100 GB"000000 "50 GB 00"
2. 000000 - 00000000000000000000
3. 000000 - 0000 ID 00000000000000000000000000
4. 0000 - 0000 ***topup**000000000000 70 GB 000000 100 GB
 - o 0000 ID 000000 "AU_Data_Domestic__107374182400" 0000 100 GB
 - o 00000000170 GB
 - o UI 0000 "170 GB (70 GB rolled over + 100 GB new)"

0000 - 0000 **ID**

0000 0000000000000000 UI 00000000000000000000

```

# 0 - 0000 + 00000000
Action_Data_100GB = {
    "Actions": [
        {
            "BalanceId": f"AU_Data_Domestic_{100 * 1024 * 1024 *
1024}",
            "Units": 100 * 1024 * 1024 * 1024
        }
    ]
}

# 0 - 0000000000
Action_Data_100GB = {
    "Actions": [
        {
            "BalanceId": f"AU_Data_Domestic_100GB_{100 * 1024 *
1024 * 1024}", # 000
            "Units": 100 * 1024 * 1024 * 1024
        }
    ]
}

# 0 - 00000000UI 00000000
Action_Data_100GB = {
    "Actions": [
        {
            "BalanceId": "AU_Data_Domestic", # 00 __size
            "Units": 100 * 1024 * 1024 * 1024
        }
    ]
}

```

0000 - 00000

00000000000000000000000000000000

```
# 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 套餐
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "{{ balance_id }}",
            "Value": 5368709120,
            "ExpiryTime": "+5d",
            "Weight": 10
          }
        }]
      }

```

充值套餐 *topup 套餐

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

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

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

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

00 2000 ID 0 *topup0000

0000000 ID 0 *topup 0000000000000000

00000

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

- name: 账单 *topup 账单 ID
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "Data_5days__5368709120",
            "Value": 5368709120,
            "ExpiryTime": "+5d",
            "Weight": 10
          }
        }]
      }
}
```

账单 ID 账单 ID

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

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

3 ID *topup_reset

ID *topup_reset

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

- name: 00 *topup_reset 0000000000
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.SetActions",
        "params": [{
          "ActionsId": "Action_Reset_Monthly_Plan",
          "Actions": [{
            "Identifier": "*topup_reset",
            "BalanceType": "*data",
            "Units": 32212254720, # 30GB
            "ExpiryTime": "*monthly",
            "DestinationIds": "*any",
            "BalanceId": "Monthly_Plan__32212254720",
            "Weight": 10
          }]
        }]
      }
```

000 00000000000000 30GB0000000000

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

1 1 1 1 1

CGRateS

1 1 1 1 1

Blocker: true

1. →
2. →
3. → INSUFFICIENT_CREDIT_BALANCE_BLOCKER

00000

- 0000000000000000 000000000000000000000000
- 0000000000000000000000CGRateS 0000 000000
- 0000000000000000 *voice *data *sms *monetary

0000000

1. 0000

000000000000000000000000

```
Action_Suspend_Account = {  
    "id": "0",  
    "method": "ApierV1.SetActions",  
    "params": [{  
        "ActionsId": "Action_suspend-account",  
        "Overwrite": True,  
        "Tenant": tenant,  
        "Actions": [  
            # 0000000000000000  
            {  
                "Identifier": "*topup",  
                "BalanceId": "Suspension_Blocker",  
                "BalanceType": "*monetary",  
                "DestinationIDs": "*any",  
                "Units": 0, # 00  
                "BalanceWeight": 9999, # 00000 - 00000  
                "Blocker": True, # 000000  
                "Weight": 10  
            }  
        ]  
    }]  
}  
  
result = OCS_Obj.SendData(Action_Suspend_Account)
```

000 0000/00/SMS 0000000000000000

2. 0000

□□□□□□□□□□□□□□□□

```
Action_Monthly_Plan_With_Cap = {
  "id": "0",
  "method": "ApierV1.SetActions",
  "params": [{
    "ActionsId": "Action_monthly-with-cap",
    "Overwrite": True,
    "Tenant": tenant,
    "Actions": [
      # 10GB □□□□
      {
        "Identifier": "*topup_reset",
        "BalanceId": f"Included_Data__{10 * 1024 * 1024 *
1024}",
        "BalanceType": "*data",
        "DestinationIDs": "Dest_PLMN_OnNet",
        "Units": 10 * 1024 * 1024 * 1024,
        "ExpiryTime": "*month",
        "BalanceWeight": 1200, # □□□□
        "Weight": 95
      },
      # $50 □□□□□□□□□□
      {
        "Identifier": "*topup_reset",
        "BalanceId": "Overage_Cap",
        "BalanceType": "*monetary",
        "DestinationIDs": "*any",
        "Units": 5000, # $50.00 □□□□
        "ExpiryTime": "*month",
        "BalanceWeight": 1000, # □□□□□□
        "Blocker": True, # □ $50 □□□□□□□□
        "Weight": 90
      }
    ]
  }
}]
}
```

□□□

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

3. Overage_Cap 0 \$0 →

3.

```

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

```

100

4.

```

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

```

普通

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

□□□□□□□□□□

□□□ Blocker Disabled □□□

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

```
# □□□□□□ 10GB□□□□□□□□□□  
{  
  "BalanceId": f"Data_Cap_{10 * 1024 * 1024 * 1024}",  
  "Units": 10 * 1024 * 1024 * 1024,  
  "Blocker": True # □ 10GB □□□□□  
}  
  
# □□□□□□□□□□□□□□□□□□  
{  
  "BalanceId": f"Bonus_Data_{5 * 1024 * 1024 * 1024}",  
  "Units": 5 * 1024 * 1024 * 1024,  
  "Disabled": True # □□□□□□□□□□  
}
```

□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□

```

Action_Safe_Hybrid_Plan = {
  "id": "0",
  "method": "ApierV1.SetActions",
  "params": [{
    "ActionsId": "Action_safe-hybrid-plan",
    "Overwrite": True,
    "Tenant": tenant,
    "Actions": [
      {
        "Identifier": "*reset_account",
        "Weight": 700
      },
      # 500 000000
      {
        "Identifier": "*topup_reset",
        "BalanceId": f"Domestic_Voice_{500 * 60 *
10000000000}",
        "BalanceType": "*voice",
        "DestinationIDs": "Dest_Domestic_All",
        "Units": 500 * 60 * 10000000000,
        "ExpiryTime": "*month",
        "BalanceWeight": 1200,
        "Weight": 95
      },
      # $20 0000
      {
        "Identifier": "*topup_reset",
        "BalanceId": "Overage_Allowance",
        "BalanceType": "*monetary",
        "DestinationIDs": "*any",
        "Units": 2000, # $20.00
        "ExpiryTime": "*month",
        "BalanceWeight": 1000,
        "Weight": 90
      },
      # $50 00000000
      {
        "Identifier": "*topup_reset",
        "BalanceId": "Hard_Spending_Cap",
        "BalanceType": "*monetary",
        "DestinationIDs": "*any",
        "Units": 5000, # $50.00 0000
        "ExpiryTime": "*month",

```

```

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

```

00000

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

000000 1200 00000000 \$50

0000000000

1. 0000000000

```

"BalanceWeight": 9999 # 0000000000

```

2. 000000000000

```

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

```

3. 000000000000

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

4. 000000000000

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

5. 00000000

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

00000000

000000000000000000000000

000000

- 000000000000000000000000
- DestinationIDs 0000000000
- 0000000000 True

000000

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

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

0000000000

000000 00000000000000000000

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

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

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

Item 1 is larger than Item 2

Item 3

Item 3 is the largest

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

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

Item 2 is larger and valid longer

```
# 数据包的元数据
```

```
包 1:
```

```
ID: "Data_Package_A"
```

```
大小: 5GB
```

```
创建时间: 2024-12-25
```

```
索引: 11 # 索引 = 索引
```

```
包 2:
```

```
ID: "Data_Package_B"
```

```
大小: 10GB
```

```
创建时间: 2025-01-15
```

```
索引: 10 # 索引 = 索引
```

```
# 包 1 的索引 11 > 包 2 的索引 10
```

```
数据包的元数据
```

```
包 1
```

```
包 1 数据包的元数据
```

```
包 1 数据包的元数据“5GB 5”数据包的元数据
```

```
包 1
```

```

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

- name: 随机生成 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
          }
        }]
      }

```

测试

- 12月24日 00:00:00 → 5GB 2024年12月29日 00:00:00
- 12月25日 00:00:00 → 5GB 2024年12月30日 00:00:00
- 权重为10

2

“5GB 套餐”

```

- name: 充值接口
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "Rollover_Monthly_50GB",
            "Value": 53687091200, # 50GB
            "ExpiryTime": "+720h", # 30天
            "Weight": 10
          }
        }]
      }

```

充值接口 *topup 充值接口

充值接口

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

充值 3 充值接口

充值 “100GB 充值” 充值接口 100GB 充值

充值

```
- name: []
uri:
  url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
  method: POST
  body_format: json
  body:
    {
      "method": "ApierV1.SetActions",
      "params": [{
        "ActionsId": "Action_Monthly_100GB_Reset",
        "Overwrite": true,
        "Actions": [{
          "Identifier": "*topup_reset",
          "BalanceType": "*data",
          "Units": 107374182400, # 100GB
          "ExpiryTime": "*monthly",
          "BalanceId": "Monthly_Plan__107374182400",
          "Weight": 10
        }]
      }]
    }
```

```
- name: [] ActionPlan
uri:
  url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
  method: POST
  body_format: json
  body:
    {
      "method": "ApierV1.SetActionPlan",
      "params": [{
        "Id": "ActionPlan_Monthly_100GB",
        "ActionPlan": [{
          "ActionsId": "Action_Monthly_100GB_Reset",
          "Time": "*monthly",
          "Weight": 10
        }],
        "Overwrite": true,
        "ReloadScheduler": true
      }]
    }
```

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

```
uri:
  url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
  method: POST
  body_format: json
  body:
    {
      "method": "ApierV2.SetAccount",
      "params": [{
        "Account": "{{ service_uuid }}",
        "ActionPlanIds": ["ActionPlan_Monthly_100GB"],
        "ReloadScheduler": true
      }]
    }
}
```

□□□□□

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

□□ **4**□□□□□□□□□□□□□□

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

□□□□

```
# 充值送话费
- name: 充值送话费
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "Bonus_Data",
            "Value": 5368709120, # 5GB
            "ExpiryTime": "+240h", # 10天
            "Weight": 20
          }
        }]
      }
}
```

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

□□□□

- □□ 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 - 00/000000000000000000000000000000

ExpiryTime - 00000000

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

Balanced - 0000000000 ID = 000000 ID = 0000

Weight - 0000000000 = 00000000000000

Blocker - 000 true000000000000

Disabled - 000 true000000000000

00 Python 00000000000000

0000000000000000 Python 0000 cgrateshttpapi 0000000000000000
OCS_Obj.SendData() 000000

구현

```
import cgrateshttpapi
import time

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

구현

구현

구현

- 구현 - GSM/SMS 구현 Dest_International_UK
- PLMN** 구현 - 구현 Dest_PLMN_OnNet Dest_PLMN_US_Verizon

구현

- GSM/SMS 구현 → 구현 TO
- 구현 → 구현 PLMN 구현 FROM

구현

- 구현
- PLMN 구현
- PLMN 구현
- 구현

구현 **CGRateS** 구현

구현

구현

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

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

# 3000SMS의 크기
3000_sms = 3000
```

1 Python

```

Action_AU_Premium_Plan_1 = {
  "id": "0",
  "method": "ApierV1.SetActions",
  "params": [{
    "ActionsId": "Action_au-premium-plan-1",
    "Overwrite": True,
    "Tenant": str(tenant),
    "Actions": [
      # 重置账户
      {
        "Identifier": "*reset_account",
        "Weight": 700
      },
      # 100GB 国内流量
      # 国内流量 PLMN 重置
      # 国内流量 PLMN
      {
        "Identifier": "*topup_reset",
        "BalanceId": "AU_Data_Domestic__" + str(100 * 1024
* 1024 * 1024),
        "BalanceType": "*data",
        "DestinationIDs": "Dest_PLMN_OnNet", # 国内 PLMN
[mcc505.mnc057]
        "Units": 100 * 1024 * 1024 * 1024,
        "ExpiryTime": "*month",
        "BalanceWeight": 1200,
        "Weight": 90
      },
      # 3000 国内语音
      {
        "Identifier": "*topup_reset",
        "BalanceId": "AU_Voice_Domestic__" + str(3000 * 60
* 1000000000),
        "BalanceType": "*voice",
        "DestinationIDs":
"Dest_AU_Mobile;Dest_AU_Fixed;Dest_AU_TollFree;",
        "Units": 3000 * 60 * 1000000000,
        "ExpiryTime": "*month",
        "BalanceWeight": 1200,
        "Weight": 89
      },
      # 3000 国内语音
      {

```

```

        "Identifier": "*topup_reset",
        "BalanceId": "AU_SMS_Domestic__" + str(3000),
        "BalanceType": "*sms",
        "DestinationIDs": "Dest_AU_Mobile;",
        "Units": 3000,
        "ExpiryTime": "*month",
        "BalanceWeight": 1200,
        "Weight": 88
    },
    # 6GB 套餐
    {
        "Identifier": "*topup_reset",
        "BalanceId": "AU_Roaming_Data__" + str(6 * 1024 *
1024 * 1024),
        "BalanceType": "*data",
        "DestinationIDs": "Dest_Roaming_All",
        "Units": 6 * 1024 * 1024 * 1024,
        "ExpiryTime": "*month",
        "BalanceWeight": 1100,
        "Weight": 87
    },
    # CDR
    {
        "Identifier": "*cdrlog",
        "BalanceType": "*generic",
        "ExtraParameters": "
{\"Category\": \"^activation\", \"Destination\": \"AU Premium Plan
1\"}"
    }
    ]
}

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

```

测试

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

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

2 Python

20GB

```

Action_AU_Data_Addon_20GB = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_au-data-addon-20gb",
        "Overwrite": True,
        "Tenant": str(tenant),
        "Actions": [
            # 重置账户
            {
                "Identifier": "*reset_account",
                "Weight": 700
            },
            # 20GB 国内流量
            # 重置 20GB 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", # 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("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 1000000
            {
                "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")

```

100 100/SMS 10000000000000000000000000 PLMN 10000000000000000000000000 10000 10000000000

Python 10000000

10000000

- **ActionsId** 10000 - 1000000000000000000 CRM 10 product_slug 100000
- **Overwrite** - 1000 True 10000000000 ID 100000
- **Tenant** - CGRateS 10000

- **Actions** - 000000000000

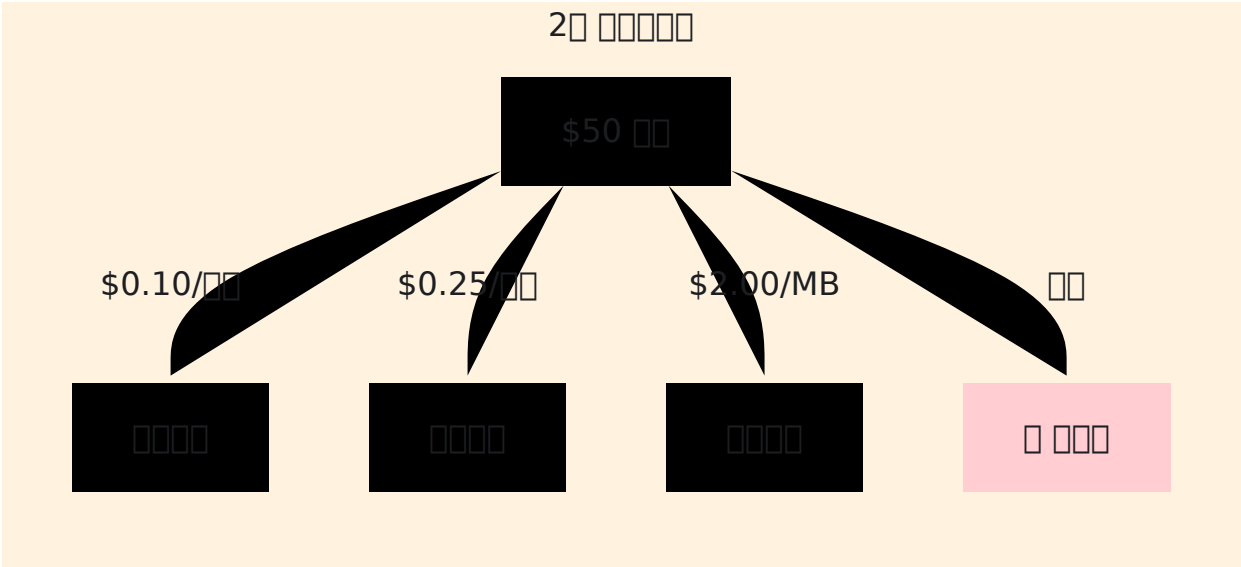
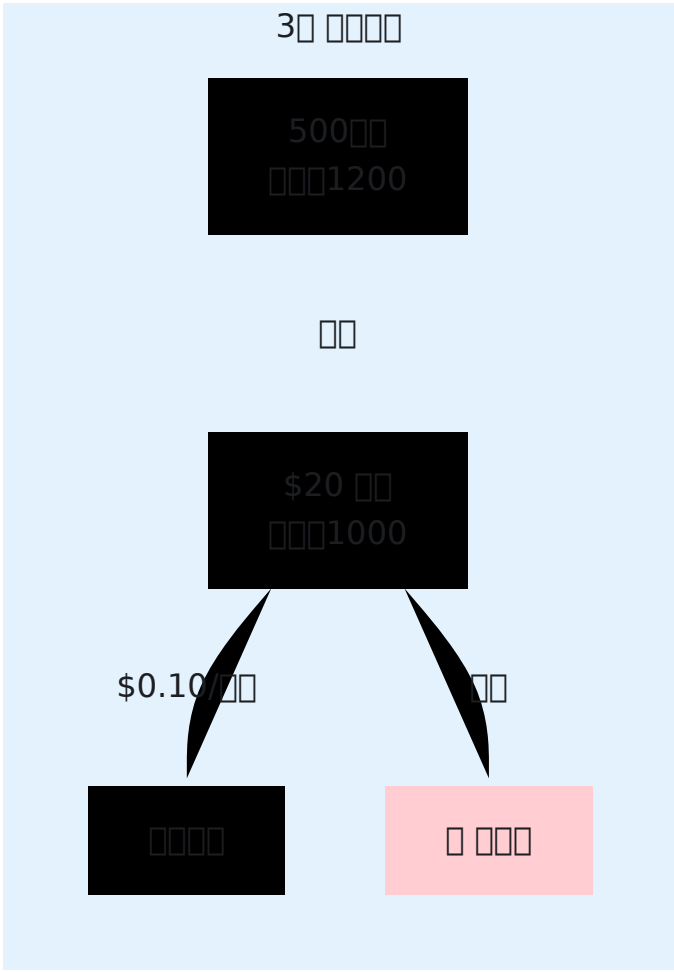
00000000

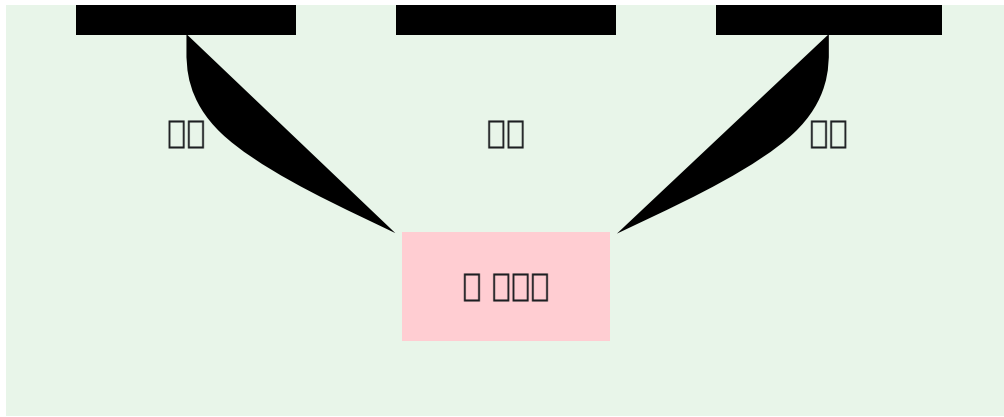
- **Identifier** - 000000 *topup 0 *topup_reset 0 *cdrlog 00
- **BalanceId** - 000000000000000000000000
- **BalanceType** - 000000 *data 0 *voice 0 *sms 0 *monetary 0
- **DestinationIDs** - 00000000000000
 - 00 00/SMS 00000000000000
 - 00 "Dest_AU_Mobile" 0 "Dest_International_UK" 0
 - 00 000000 PLMN 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

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

RAM 1GB

```

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    -    
                         



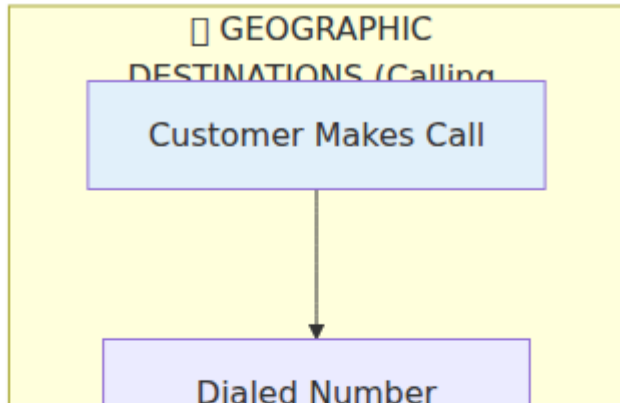
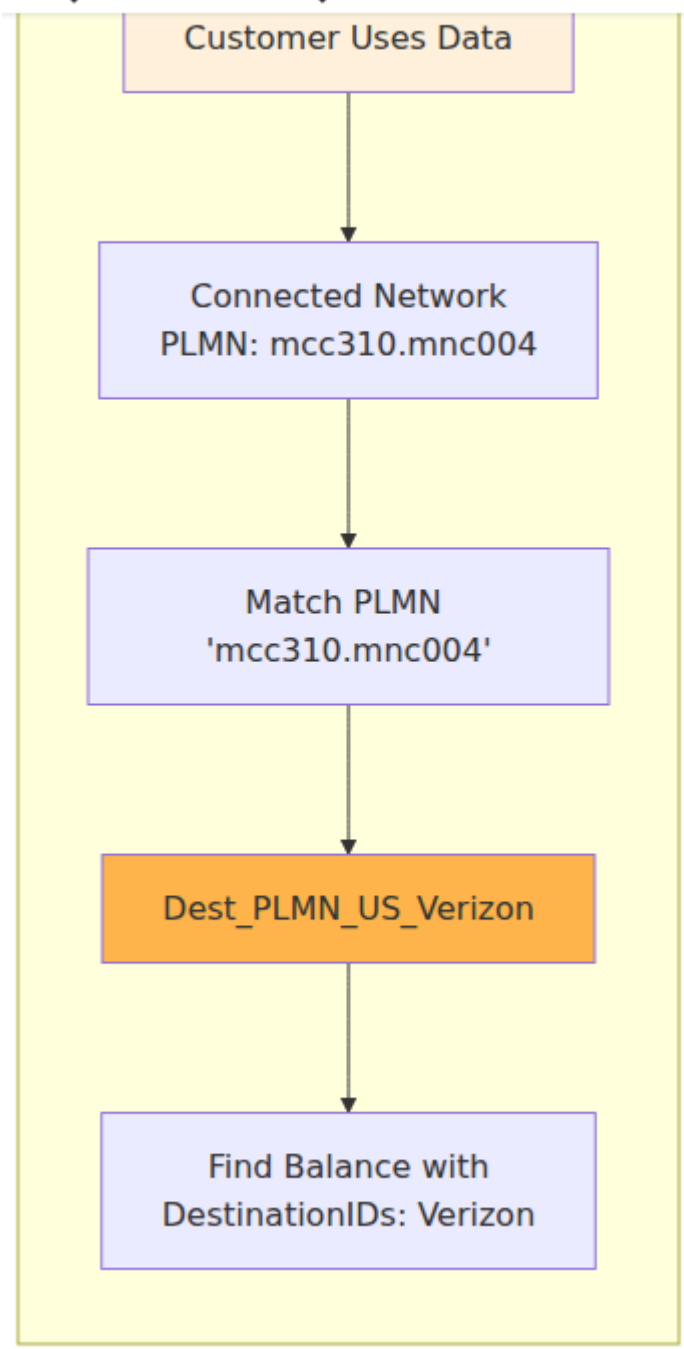
CGRateS           

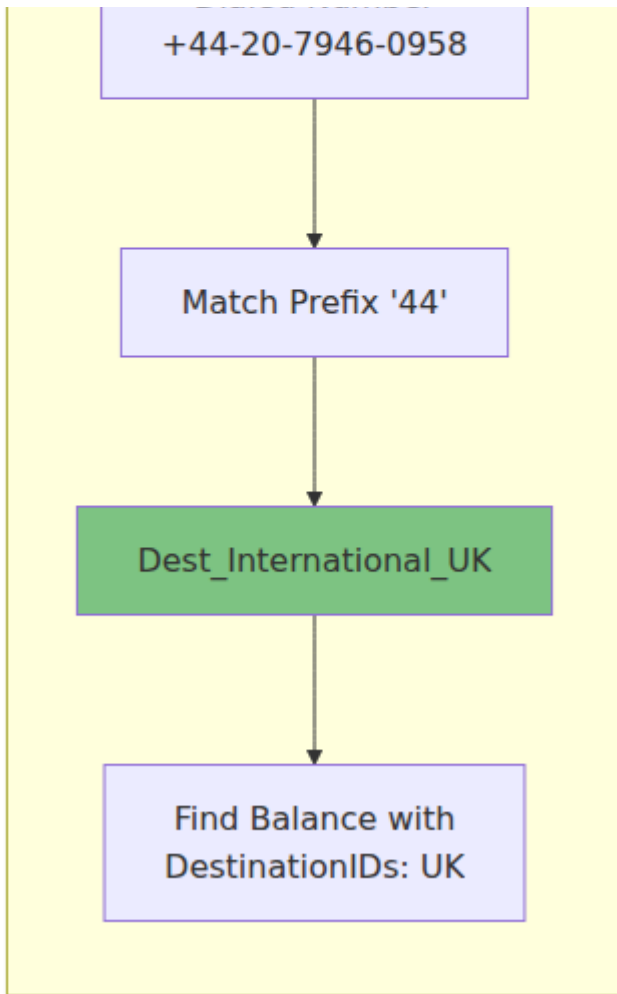
-   -                    
- **PLMN**   -                    

-                          
-                          

OmniCharge OmniRAN Downloads





□□□□

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

□□□□

```

import cgrateshttpapi
import time

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

1 □□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□ SMS □□□□□□

□□□□□

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

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

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

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

□□□□□

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

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

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

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

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

```

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

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

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

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

```


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

`XXXXXXXXXX`

`XXXXXXXXXXXX SIM XXXXXXXXXXXX`

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

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

PLMN

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

# Verizon Wireless
OCS_Obj.SendData({
  'method': 'ApierV2.SetTPDestination',
  'params': [{
    "TPid": tpid,
    "ID": "Dest_PLMN_US_Verizon",
    "Prefixes": [
      "mcc310.mnc004", "mcc310.mnc010", "mcc310.mnc012",
      "mcc310.mnc013",
      "mcc311.mnc480", "mcc311.mnc481", "mcc311.mnc482",
      "mcc311.mnc483"
    ]
  }]
})

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

PLMN

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

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

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

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

PLMN

PLMN

```

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

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

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

# Zone 3: Asia Pacific
OCS_Obj.SendData({

```

```

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

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

```

□□□

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

Table 1

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

Table 2

Table 2.1

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

Table 2.2

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

Table 3

Table 3.1

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

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

□□□□□□□□□□□□□□

```

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

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

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

```

Dest_PLMN_Zone_Europe
mcc262
mcc208
mcc222
mcc214
mcc234

mcc262
mcc208
mcc222
mcc214
mcc234

mcc262
mcc208
mcc222
mcc214
mcc234

```

# 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"]
  }]
})

```

000 00000000000000000000000000000000 0000000000000000 Dest_PLMN_Zone_Europe 00000000
 0000 PLMN 0000

00 **3000/SMS** 000000

000000000000 SMS 00000000

```

# 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"

"1"

"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: SIM Card Provisioning

OmniCRM uses **Ansible** for SIM Card Provisioning

Concepts SIM Card Provisioning <concepts_sim_provisioning> Ansible
SIM Card Provisioning - SIM Card eSIM

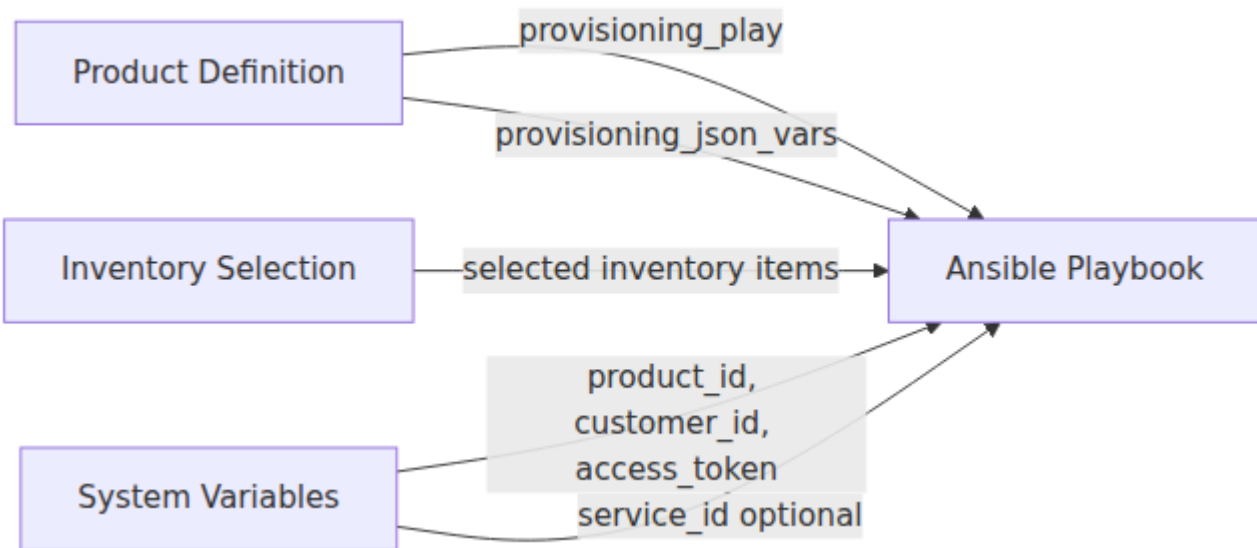
Playbooks

Playbooks in OmniCRM are Ansible playbooks that manage SIM Card Provisioning - SIM Card eSIM

Playbooks

OmniCRM uses Ansible

1. provisioning_play
2. provisioning_json_vars
3. playbook
4. 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 Example

```
# 1. Provision SIM card 1
# provisioning_play: play_simple_service
- CRM
- OCS
- SIM
-

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

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

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

OmniCRM `playbook` Example

Plays vs Tasks

Plays vs Tasks in OmniCRM playbooks

Playbook

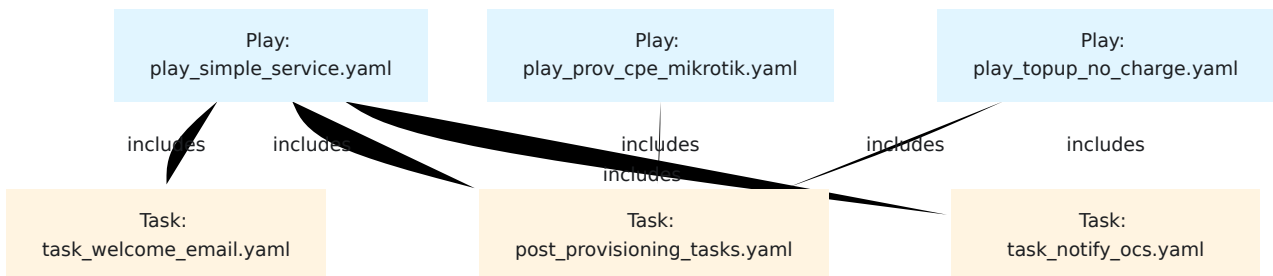
Plays `OmniCRM-API/Provisioners/plays/` playbooks

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

Task

plays Tasks `task_`

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



□□□□

name

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

hosts: localhost

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

gather_facts: no

Ansible □□□□□□□□□□□□

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

become: False

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

□□□□

□□ playbook □□□□□□□□□□□□

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

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

`crm_config.yaml` □□□□□□

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

□□□□□□

□□□□□□□□□□

□□□□□□

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

□□□□□□

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

□□ **API** □□□

```
- name: Get Product information from CRM API
  uri:
    url: "http://localhost:5000/crm/product/product_id/{{
product_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    return_content: yes
  register: api_response_product

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

📖 **Playbook** 📖

📖📖📖📖

📖📖📖📖📖📖📖📖

```

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

  tasks:
    - name: Main block
      block:

        # 1. 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. 余额增加
- name: Add 0 Monetary Balance
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*monetary",
          "Categories": "*any",
          "Balance": {
            "ID": "Initial Balance",
            "Value": 0,
            "ExpiryTime": "+4320h",
            "Weight": 1,
            "Blocker": true
          }
        }]
      }
    status_code: 200
  register: balance_response

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

- name: Add Service via API
  uri:
    url: "http://localhost:5000/crm/service/"
    method: PUT
    body_format: json
    headers:
      Content-Type: "application/json"
      Authorization: "Bearer {{ access_token }}"
    body:

```

```

        {
            "customer_id": "{{ customer_id }}",
            "product_id": "{{ product_id }}",
            "service_name": "{{ package_name }} - {{
service_uuid }}",
            "service_type": "generic",
            "service_uuid": "{{ service_uuid }}",
            "service_billed": true,
            "service_taxable": true,
            "service_provisioned_date": "{{
current_date_time.stdout }}",
            "service_status": "Active",
            "wholesale_cost": "{{
api_response_product.json.wholesale_cost | float }}",
            "retail_cost": "{{ monthly_cost | float }}"
        }
        status_code: 200
        register: service_creation_response

# 9.
- name: Add Setup Cost Transaction via API
  uri:
    url: "http://localhost:5000/crm/transaction/"
    method: PUT
    headers:
      Content-Type: "application/json"
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "customer_id": {{ customer_id | int }},
        "service_id": {{
service_creation_response.json.service_id | int }},
        "title": "{{ package_name }} - Setup Costs",
        "description": "Setup costs for {{ package_comment
}}",
        "invoice_id": null,
        "retail_cost": "{{ setup_cost | float }}"
      }
    return_content: yes
    register: transaction_response

# 10.
- include_tasks: post_provisioning_tasks.yml

```

```
rescue:

# 00/0000
- name: Print all vars for debugging
  debug:
    var: hostvars[inventory_hostname]

- name: Remove account in OCS
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "A pierV2.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

XXXXXXXXXXXXXXXXXXXXXXXX

```

- name: Service Topup Playbook
  hosts: localhost
  gather_facts: no
  become: False

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

    # 1. サービス情報取得
    - name: Get Service information from CRM API
      uri:
        url: "http://localhost:5000/crm/service/service_id/{{
service_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
        register: api_response_service

    # 2. サービス情報取得
    - name: Get Product information from CRM API
      uri:
        url: "http://localhost:5000/crm/product/product_id/{{
product_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
        register: api_response_product

    # 3. サービス情報設定
    - 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SMS
- 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SMSの送信先をONT

```

- name: CPE Provisioning Playbook
  hosts: localhost
  gather_facts: no
  become: False

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

# 1. 設定 CPE の ID
  - name: Set CPE inventory ID from hostvars
    set_fact:
      cpe_inventory_id: "{{ hostvars[inventory_hostname]['WiFi Router CPE'] | int }}"
      when: "'WiFi Router CPE' in hostvars[inventory_hostname]"

# 2. API から CPE のデータを取得
  - name: Get Inventory data for CPE
    uri:
      url: "{{ crm_config.crm.base_url
      }}/crm/inventory/inventory_id/{{ cpe_inventory_id }}"
      method: GET
      headers:
        Authorization: "Bearer {{ access_token }}"
      return_content: yes
      register: api_response_cpe

# 3. API から Site の情報を取得
  - name: Get Site info from API
    uri:
      url: "{{ crm_config.crm.base_url
      }}/crm/site/customer_id/{{ customer_id }}"
      method: GET
      headers:
        Authorization: "Bearer {{ access_token }}"
      return_content: yes
      register: api_response_site

# 4. CPE の位置情報を更新
  - name: Patch CPE inventory item with location
    uri:

```

```
url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ cpe_inventory_id }}"
method: PATCH
body_format: json
headers:
  Authorization: "Bearer {{ access_token }}"
body:
  {
    "address_line_1": "{{
api_response_site.json.0.address_line_1 }}",
    "city": "{{ api_response_site.json.0.city }}",
    "state": "{{ api_response_site.json.0.state }}",
    "latitude": "{{ api_response_site.json.0.latitude }}",
    "longitude": "{{ api_response_site.json.0.longitude
}}}"
  }
status_code: 200
```

5. 更新 CPE

```
- name: Set CPE hostname
set_fact:
  cpe_hostname: "CPE_{{ cpe_inventory_id }}"
  cpe_username: "admin_{{ cpe_inventory_id }}"

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

6. 创建 WiFi 网络

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

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

- name: Create WiFi PSK
```

```

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

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

# 7. 配置文件
- name: Set config filename
  set_fact:
    config_name: "{{ cpe_hostname }}_{{ lookup('pipe', 'date
+%Y%m%d%H%M%S') }}.cfg"
    config_dest: "/tmp/{{ cpe_hostname }}_{{ lookup('pipe',
'date +%Y%m%d%H%M%S') }}.cfg"

- name: Create config from template
  template:
    src: "templates/cpe_router_config.j2"
    dest: "{{ config_dest }}"

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

# 9. 打补丁
- name: Patch CPE inventory with config
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ cpe_inventory_id }}"
    method: PATCH
    body_format: json
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      {
        "itemtext3": "{{ wifi_ssid }}",
        "itemtext4": "{{ wifi_psk }}",
        "management_url": "{{ cpe_hostname }}",
        "management_username": "{{ cpe_username }}",
        "management_password": "{{ cpe_password }}",
        "config_content": "{{ config_content.content |

```

```

b64decode }}",
    "inventory_notes": "Provisioned: {{ lookup('pipe',
'date +%Y-%m-%d') }}"
    }
    status_code: 200

# 10.
- name: Email configuration to support
  uri:
    url: "https://api.mailjet.com/v3.1/send"
    method: POST
    body_format: json
    headers:
      Content-Type: "application/json"
    body:
      {
        "Messages": [{
          "From": {
            "Email": "provisioning@example.com",
            "Name": "Provisioning System"
          },
          "To": [{
            "Email": "support@example.com",
            "Name": "Support Team"
          }],
          "Subject": "CPE Config - {{ cpe_hostname }}",
          "Attachments": [{
            "ContentType": "text/plain",
            "Filename": "{{ config_name }}",
            "Base64Content": "{{ config_content.content }}"
          }
        ]
      }
    }
    user: "{{ mailjet_api_key }}"
    password: "{{ mailjet_api_secret }}"
    force_basic_auth: true
    status_code: 200

```

□□□□□□

□□ CGRateS ActionPlans □□□□□□□□□□

```

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

# 2. 创建 Action
- 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
- 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. `ApierV1.ActionPlan` `actions`

- 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

```

目录

playbooks 目录 plays 目录

目录

task_welcome_email.yaml - 目录

```

# 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
```

□□

□□□□□□□□

```

# 0000000000000000
# 00000 15 000000000 1 00
# 0000000000 50% 000
- name: Calculate days until end of month
  command: "date -d 'last day of this month' +%d"
  register: days_in_month

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

- name: Calculate pro-rata amount
  set_fact:
    days_remaining: "{{ (days_in_month.stdout | int) -
(current_day.stdout | int) }}"
    pro_rata_cost: "{{ (monthly_cost | float) * (days_remaining |
float) / (days_in_month.stdout | float) }}"

```

00 Playbooks

0000

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

0000 playbook

```

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

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

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

    - name: Verify health check
      assert:
        that:
          - health_check.status == 200

```

□□□□

□□□□□□

```

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

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

```

□□□□□□□□□□

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
- 配置環境
- 執行安裝

配置

配置環境

1. 安裝 Ansible

安裝 Ansible 的步驟

2. 配置 Playbook

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

□□□□□

□□ OmniCRM playbooks □□□□□□□□□□

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

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

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

      - name: Add transaction
        uri: ...

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

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

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

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

□□□□□□

□□ play_simple_service.yaml □□□□□

```

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

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

  # 2.  []
  - name: Try to get Service information from CRM API to
    Deprovision
    uri:
      url: "http://localhost:5000/crm/service/service_id/{{
service_creation_response.json.service_id }}"
      method: GET
      headers:
        Authorization: "Bearer {{ access_token }}"
      return_content: yes
      register: api_response_service
      ignore_errors: True
      when: service_creation_response is defined and
service_creation_response.json is defined and
service_creation_response.json.service_id is defined

  - name: Print api_response_service
    debug:
      var: api_response_service
      when: api_response_service is defined

  # 3.  [] service_uuid
  - name: Set service_uuid facts for Deprovision
    set_fact:
      service_uuid: "{{ api_response_service.json.service_uuid }}"
    when:
      - service_uuid is not defined
      - api_response_service is defined
      - api_response_service.json is defined

  # 4.  [] OCS []
  - name: Remove account in OCS
    uri:

```

```

url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
method: POST
body_format: json
headers:
  Content-Type: "application/json"
  Authorization: "Bearer {{ access_token }}"
body:
  {
    "method": "ApierV2.RemoveAccount",
    "params": [{
      "Tenant": "{{ crm_config.ocs.ocsTenant }}",
      "Account": "{{ service_uuid }}",
      "ReloadScheduler": true
    }]
  }
  status_code: 200
register: response
ignore_errors: True
when: service_uuid is defined

- name: Print response from OCS account removal
  debug:
    var: response
  when: response is defined

# 5. CRM サービス削除
- name: Delete Service from CRM if it was created
  uri:
    url: "http://localhost:5000/crm/service/service_id/{{
service_creation_response.json.service_id }}"
    method: DELETE
    headers:
      Authorization: "Bearer {{ access_token }}"
    status_code: 200
  register: delete_service_response
  ignore_errors: True
  when: service_creation_response is defined and
service_creation_response.json is defined and
service_creation_response.json.service_id is defined

- name: Print delete service response
  debug:
    var: delete_service_response
  when: delete_service_response is defined

```

```
# 6. 状態確認
- 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"
```

状態確認

ignore_errors: True

状態確認 ignore_errors: True 状態確認

状態

when 状態確認

```
when: service_creation_response is defined
```

状態

状態確認

1. 状態確認
2. 状態確認
3. 状態 OCS/状態
4. 状態確認

状態

状態確認

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

状態確認

OCS/CGRateS 状態

```
- name: Remove account in OCS
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV2.RemoveAccount",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "ReloadScheduler": true
        }]
      }
    status_code: 200
    ignore_errors: True
    when: service_uuid is defined
```

CRM

```
- name: Delete Service from CRM
  uri:
    url: "http://localhost:5000/crm/service/service_id/{{
service_id }}"
    method: DELETE
    headers:
      Authorization: "Bearer {{ access_token }}"
    status_code: 200
    ignore_errors: True
    when: service_id is defined
```

```
- name: Release SIM card back to inventory pool
  uri:
    url: "http://localhost:5000/crm/inventory/inventory_id/{{
sim_inventory_id }}"
    method: PATCH
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "customer_id": null,
        "service_id": null,
        "item_state": "Available"
      }
    status_code: 200
  ignore_errors: True
  when: sim_inventory_id is defined
```

ActionPlans

```
- name: Remove ActionPlan from account
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.RemoveActionPlan",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Id": "ActionPlan_Monthly_{{ service_uuid }}_{{
product_id }}"
        }]
      }
    status_code: 200
  ignore_errors: True
```

```
- name: Release payment authorization
  uri:
    url: "http://localhost:5000/crm/payments/release/{{
authorization_id }}"
    method: POST
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "metadata": {
          "release_reason": "provisioning_failed"
        }
      }
    return_content: yes
  ignore_errors: True
  when: authorization_id is defined
```

□□□□□□□□

□□□□□□□□

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

□□ **API** □□□

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

□□□□□□

□□ 1 OCS □□□□□□□□□□

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

□□ 2 □□□□□□□□□□

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

□□ 3 □□□□□□□□□□

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

□□□□□□

1. □□□□ **ignore_errors**

□□□□ - □□□□□□□□□□□□

2. □□ **when** □□□□□□□□□□

□□□□□□□□□□□□

```
when: service_creation_response is defined
```

3. 環境構築

環境構築の準備

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

4. 環境構築

環境構築の準備

- 環境構築
- OCS のインストール
- 環境構築

5. 環境構築

環境構築の準備

6. 環境構築

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

環境構築

環境構築の準備

- 環境構築
- 環境構築
- 環境構築
- 環境構築
- 環境構築

環境構築の準備

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

Ansible

Ansible `block/rescue` blocks allow you to define a series of tasks to execute in a specific order, and then define a `rescue` block to handle any errors that occur during the execution of the tasks.

Example

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

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

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

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

For more information, see the `concepts_provisioning` page.

OmniCRM API

OmniCRM API - UI

OmniCRM

API RESTful API JWT API IP

API Swagger API

API URL

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

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 2024-01-01
- API 2024-01-01
- API 2024-01-01
- API cron 2024-01-01

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>

- Ansible Playbook Play <concepts_ansible> Ansible service_id ICCID

MSISDN
UI

MSISDN

UI

UI

category UI

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

Ansible playbook 安裝包

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

UI 安裝包/安裝包

安裝包

service_type 安裝包

安裝包

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

安裝包

- 安裝包
- 安裝包
- 安裝包
- 安裝包

mobile 安裝包 fixed 安裝包

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□

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

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

📄📄📄📄📄

📄📄📄📄📄📄📄📄📄📄

- 📄📄📄 - 📄📄📄📄📄📄📄📄
- 📄📄📄 - 📄📄📄📄📄📄📄📄
- 📄 - 📄📄📄📄📄📄

📄📄📄📄📄

- 📄📄📄 - 📄📄📄📄📄📄 FontAwesome 📄📄
- 📄📄📄📄 - 📄📄📄📄📄📄📄📄
- 📄/📄📄📄 - 📄📄📄📄📄
- 📄📄📄 - 📄📄📄📄📄 \$ 📄📄
- 📄📄📄 - 📄📄📄📄📄📄📄📄📄

Font Awesome

Font Awesome 5.11.0

Font Awesome 5.11.0

Font Awesome

Font Awesome

□□□□

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


□□□□□□

□□□□□□ Ansible □□□□□□□□□□

□□□□□

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

□□□□□

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


XXXX

XX

XXXX

- **product_id** - XXXXXXXXXXXXXXXX
- **product_name** - UI XXXXXXXXXXXXXXXX
- **product_slug** - URL API XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
- **category** - XXXXXX UI XXXXXXXX
 - standalone - XXXXXXXXXXXXXXXXXXXXXXX
 - addon - XXXXXXXXXXXXXXXX
 - bundle - XXXXXXXXXXXXXXXX
 - promo - XXXXXXXX
- **service_type** - XXX
XXXXX
- **comment** - XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
- **icon** - UI XXXXX FontAwesome XXXXXXXX `fa-solid fa-sim-card`

XXXX

- **retail_cost** - XXXXXX  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 0
- **wholesale_cost** - XXXXXXXXXXXXXXXXXXXXXXXX
- **retail_setup_cost** - XXXXXXXXXXXXXXXXXXXXXXXX
- **wholesale_setup_cost** - XXXXXXXXXXXXXXXXXXXXXXXX
- **tax_percentage** - XXXXXXXXXXXXXXXXXXXXXXXX 10 10% 12.5 12.5% XXXXXXXXXXXXXXXX
0XXXXXXXXXXXXXXXXXXXXXXXXXXXXX

折扣

折扣的計算方式如下：

- 折扣 10% 的 \$50.00 → 折扣 \$5.00
- 折扣 0% 的 \$50.00 → 折扣 \$0.00
- 折扣 10% 的 \$0.00 → 折扣 \$0.00

折扣條件

- **enabled** - 折扣是否啟用，默認為 false
- **residential** - 是否適用於住宅客戶
- **business** - 是否適用於商業客戶
- **customer_can_purchase** - 客戶是否可以購買，默認為 true
- **available_from** - 折扣開始日期/時間
- **available_until** - 折扣結束日期/時間

合約

- **contract_days** - 合約期限，默認為 30 天，範圍為 30 到 365 天，0 表示無限期
- **auto_renew** - 是否自動續約
 - `prompt` - 提示客戶
 - `true` - 自動續約
 - `false` - 不自動續約

- **allow_auto_renew** - 是否自动续费 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** - 关联的 Ansible playbook 名称 .yaml 文件路径 `OmniCRM-API/Provisioners/plays/`
- **provisioning_json_vars** - 关联的 JSON 文件 Ansible playbook 需要的变量 `customer_id` `product_id` `service_id` `access_token`
- **inventory_items_list** - 关联的 inventory 项 ['SIM Card', 'Mobile Number'] 每个项包含 ID 和名称
- **relies_on_list** - 依赖的 ID 列表

其他属性

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

□□□□□□

□□□□□□ **SIM**□

```
{
  "product_id": 1,
  "product_slug": "Mobile-SIM",
  "product_name": "Mobile SIM Only",
  "category": "standalone",
  "service_type": "mobile",
  "provisioning_play": "play_psim_only",
  "provisioning_json_vars": "{\"iccid\": \"\", \"msisdn\": \"\"}",
  "inventory_items_list": "['SIM Card', 'Mobile Number']",
  "retail_cost": 0,
  "retail_setup_cost": 0,
  "wholesale_cost": 3,
  "wholesale_setup_cost": 1,
  "contract_days": 0,
  "residential": true,
  "business": true,
  "enabled": true,
  "customer_can_purchase": true,
  "icon": "fa-solid fa-sim-card",
  "features_list": "['Australian Phone Number (04xxx)', 'Fastest speeds', 'Best coverage', 'Roaming on the Mainland']",
  "terms": "Must be activated within 6 months. All credit lost if service is not used for 12 months.",
  "comment": "Physical SIM card for use with Mobile Phones"
}
```

XXXXXXXXXXXXXXXXXXXXSIM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

□□□□□□□□□□

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

Service Data

```

{
  "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"
}

```

The `service_id` field is used to identify the service in the API.

Getting Started with Ansible

Ansible provides a simple way to manage your infrastructure using YAML-based playbooks. You can use Ansible to manage your servers, networks, and cloud resources.

Installation

Ansible is available on most Linux distributions. You can install it using the following command:

1. `pip` install **provisioning_json_vars** - `pip install provisioning_json_vars`
2. `pip` install `ansible` - `pip install ansible`
3. `ansible-playbook` - `ansible-playbook my_playbook.yml`

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 - 客户 ID
3. **Playbook** 产品 ID - 产品 playbook 调用 CRM API

示例 - SIM 卡 ID: 789 - 客户 ID: 101

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

Playbook 调用 ID 调用 CRM API 调用 ICCID IMSI MSISDN

Ansible 示例

示例 `extravars` 调用 Ansible playbook 调用 playbook

```
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 访问令牌
- `provisioning_json_vars` 配置
- API 调用

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

- `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

欄位 `include_disabled=true` 查詢

UI 查詢

- 查詢
- 查詢
- 查詢
- 查詢

查詢

查詢

1. 查詢 - 欄位 `service_type` 查詢
 - 查詢 `service_type = "mobile"` 查詢 `service_type = "mobile"` 查詢
 - 查詢
2. 查詢 - 欄位 `relies_on_list`
 - 查詢
 - 查詢
3. 查詢 - 欄位 `residential` 及 `business` 查詢

查詢

查詢 `service_type = "mobile"` 查詢

- `business = true` `category != "addon"`
- `business = true` `category = "addon"` `service_type = "mobile"`
-

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□

- **service_id** - □□□□□□□□□□□□□□□□
- **customer_id** - □□□□□□□□□□□□□□□□
- **product_id** - □□□□□□□□□□□□□□□□
- **service_name** - □□□□□□□□□□□□
- **service_type** - □□□□□□□□□□VoIP□IPTV□□□□□□□□□□
- **service_uuid** - □□ OCS/CGRateS □□□□□□□□□□□□□□□□
- **icon** - □□□□□□□□□□□□ FontAwesome □□□□□□□□□□

□□□□□□□□

- **service_status** - □□□□□□□□□□□□□□□□□□□□
- **service_provisioned_date** - □□□□□□□□□□□□□□□□□□
- **service_active_date** - □□□□□□□□□□□□□□
- **service_deactivate_date** - □□□□□□□□□□□□□□□□
- **contract_end_date** - □□□□□□□□□□□□□□

□□□□□□

- **retail_cost** - □□□□□□□□□□□□□□□□□□
- **wholesale_cost** - □□□□□□□□□□□□
- **service_billed** - □□□□□□□□□□□□□□□□true□
- **service_taxable** - □□□□□□□□□□□□□□□□true□
- **invoiced** - □□□□□□□□□□□□□□□□□□
- **promo_code** - □□□□□□□□□□□□□□□□

Attributes

- **service_visible_to_customer** - `service_visible_to_customer` true
- **service_usage_visible_to_customer** - `service_usage_visible_to_customer` true

Playbooks

- **provisioning_play** - Ansible playbook
- **provisioning_json_vars** - playbook vars
- **deprovisioning_play** - Ansible playbook
- **deprovisioning_json_vars** - playbook vars

Fields

- **bundled_parent** - `service_id`
- **site_id** - `site_id`

Metadata

- **service_notes** - `service_notes`
- **created** - `created`
- **last_modified** - `last_modified`

API/UI

API/UI

PATCH `/crm/service/{service_id}`

- `service_name`
- `service_type`
- `service_status`
- `service_notes`
- `retail_cost`
- `wholesale_cost`
- `service_billed`
- `service_taxable`
- `service_visible_to_customer`
- `service_usage_visible_to_customer`
- `service_active_date`
- `service_deactivate_date`
- `contract_end_date`
- `icon`
- `promo_code`
- `site_id`

Attributes

Attributes

- service_id, customer_id, product_id
- service_uuid
- service_provisioned_date
- provisioning_play, provisioning_json_vars
- deprovisioning_play, deprovisioning_json_vars
- bundled_parent
- invoiced
- created, last_modified

Usage

Usage: Web UI, API, Ansible playbooks

Usage

1. API - Ansible playbooks
2. - "Relies On"
3. -
 - /
 - `relies_on_list`
 - `available_from/available_until`
 - `enabled` | `customer_can_purchase`
4. -
5. - `inventory_items_list` SIM
6. - "API"

API Ansible

###

Step 1: Prerequisites

API Prerequisites

- `provisioning_play` - Ansible playbook (run `play_psim_only`)
- `provisioning_json_vars` - Provisioning JSON vars
- `customer_id` - Customer ID
- `product_id` - Product ID
- `service_id` - Service ID
- `access_token` - API Access Token

Step 2: Execution

Execution Steps

1. Run `provisioning_json_vars` to provision the service
2. Verify the service is provisioned
3. Retrieve the access token
 - `access_token` - API Access Token (CRM JWT)
 - `initiating_user` - Initiating User ID
 - `customer_id`, `product_id`, `service_id`
4. Run `{inventory_type: inventory_id}` to provision the inventory

Example JSON

```
{
  "customer_id": 123,
  "product_id": 456,
  "service_id": 789,           # Service ID
  "SIM Card": 1001,          # SIM Card ID
  "Mobile Number": 1002,    # Mobile Number
  "monthly_cost": 30,       # From provisioning_json_vars
  "data_gb": 50,           # From provisioning_json_vars
  "access_token": "eyJ...", # API Access Token
  "initiating_user": 5      # Initiating User ID
}
```

Step 3: Verification

Provisioner Provision

- `provision_id` - ID
- `provisioning_play` - playbook
- `provisioning_json_vars` - JSON
- `task_count` - playbook YAML
- `provisioning_status` - 1 = 0 = 2 = 1
- `product_id` `customer_id` `service_id` -

4 playbook

API

1. `OmniCRM-API/Provisioners/plays/{playbook_name}.yaml` playbook YAML
2. `ansible_runner.run()`
 - `playbook` - YAML
 - `extravars` - Ansible
 - `inventory` - `'localhost,'`
 - `event_handler` -
3. playbook

5 ProvisioningEventHandler

Ansible Provision_Event

- `event_name` - playbook
- `event_number` -
- `provisioning_status` -
 - **0** = -
 - **1** = -
 - **2** = -
 - **3** = - `ignore_errors: true` playbook
- `provisioning_result_json` -

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` `inventory_id`

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`

`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 令牌
- `initiating_user` - 發起用戶
- 服務 ID `{{ inventory_type }}: inventory_id`
- 提供 `provisioning_json_vars` 變數
- 其他參數

Playbooks 執行

- 獲取 `GET /crm/product/product_id/{{ product_id }}`
- 獲取 `GET /crm/customer/customer_id/{{ customer_id }}`
- 獲取 `GET /crm/inventory/inventory_id/{{ SIM_Card }}`
- 發送 `POST /crm/transaction/`
- 更新 `PUT /crm/service/`
- 修補 `PATCH /crm/service/{{ service_id }}`
- 修補 `PATCH /crm/inventory/inventory_id/{{ inventory_id }}`

執行 **playbook**

執行步驟

1. Playbook 需要 `customer_id` `product_id` `service_id` `access_token`
2. 獲取 `GET /crm/service/{{ service_id }}` 取得 `service_uuid`
3. 獲取 `GET /crm/product/product_id/{{ product_id }}` 取得 `product_id`
4. 向 OCS 發送 `POST to CGRateS` 提供 `retail_cost`
5. 向 OCS 發送 `POST to CGRateS` 提供其他參數
6. 向 CRM 發送 `POST /crm/transaction/` 提供相關數據
7. 完成

執行 `Provision` 事件 `Provision_Event` 的相關操作

OCS 執行

OCS 執行 `CGRateS` 相關操作與 CRM 的集成，OCS 執行相關操作

- **NUMBERS** - **NUMBERS**DID **NUMBERS**
- **NUMBERS** - **NUMBERS** MB **NUMBERS**
- **NUMBERS** - **NUMBERS** GB **NUMBERS**
- **NUMBERS** - **NUMBERS** \$30 **NUMBERS** 10GB **NUMBERS**
- **NUMBERS** - **NUMBERS**

CRM **NUMBERS** OCS **NUMBERS**

NUMBERS

NUMBERS OCS/CGRateS **NUMBERS**

NUMBERS

NUMBERS UI **API****NUMBERS**

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

```

{
  "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

- /
-
-

□□□□□□

Web UI □□□□□□□□

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

□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□ / □□□□□□

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

□□□

□□□□□□□□□□□□

□□□□/□□□
 Action □□□

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

[[]]

[[\$10 PAYG]]

[[\$0]]

[[/]]

[[ThresholdS]]

[[PAYG / \$0]]

[[CRM]] [[OCS]]

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

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

[[OCS]] [[ActionPlans]]

[[ThresholdS]]

[[CRM]] [[ActionPlans]]

[[ActionPlans]] [[CRM]]

[[ActionPlans]]

[[CRM]] [[ActionPlans]]

1. API - GET /crm/service/{service_id} API

-
- `service_uuid` OCS
- CGRateS API UUID ActionPlans
- `Get_ActionPlans(service_uuid)` CGRateS

2. ActionPlan - ActionPlan

```
{
  "ActionPlanId": "ServiceID_abc-123-
def_ProductID_456_...",
  "PlanName": "Monthly_Renewal_Plan",
  "NextExecTime": "2025-02-01T00:00:00Z",
  "custom_NextExecTime_hr": "in 11 days",
  "ActionPlanId_split_dict": {
    "ServiceID": "abc-123-def",
    "ProductID": 456,
    "CustomerID": 789,
    ...
  }
}
```

- `ActionPlanId` - / /
- `PlanName` - playbook
- `NextExecTime` - ActionPlan ISO
- `custom_NextExecTime_hr` - "11" "2025 2 1"
- `ActionPlanId_split_dict` - ActionPlanId

3. UI - ActionPlans ActionPlansTable

- ActionPlanId ProductID
- `retail_cost`
- `custom_NextExecTime_hr`
-

- `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

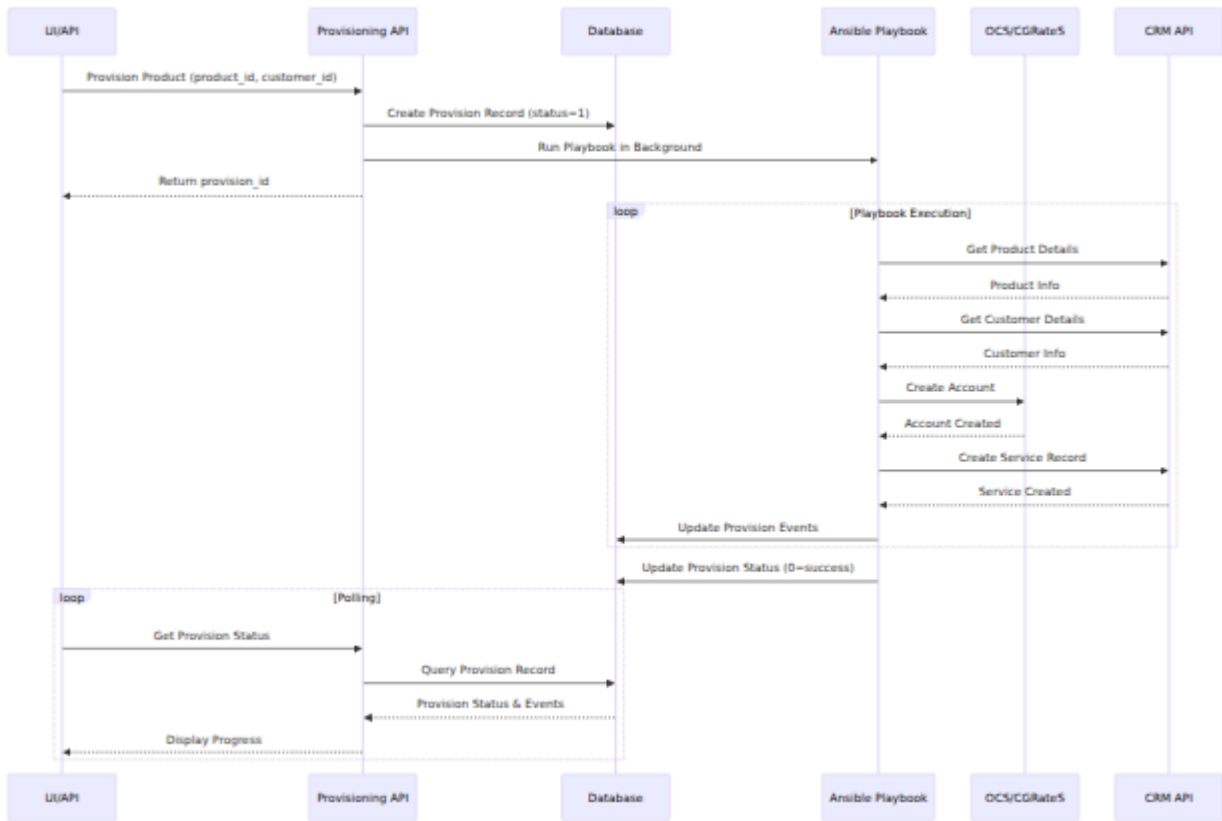
-
- ActionPlans
- ActionPlans

-
-
-
-



OmniCRM **Ansible** SIM provisioning
SIM

SIM <concepts_sim_provisioning> SIM eSIM



- `provisioning_json_vars` - Ansible JSON
- `inventory_items_list` - `['SIM Card', 'Mobile Number']`
- `product_id` `product_name` -

```
{
  "product_id": 1,
  "product_slug": "Mobile-SIM",
  "product_name": "Mobile SIM Only",
  "provisioning_play": "play_psim_only",
  "provisioning_json_vars": "{\"iccid\": \"\", \"msisdn\": \"\"}",
  "inventory_items_list": ["SIM Card", "Mobile Number"],
  "retail_cost": 0,
  "retail_setup_cost": 0,
  "wholesale_cost": 3,
  "wholesale_setup_cost": 1
}
```

1. `provisioning_play`

`OmniCRM-API/Provisioners/plays/play_psim_only.yaml`

2. `extra_vars`

- i. `provisioning_json_vars` `{"iccid": "", "msisdn": ""}`

- ii. /API

- iii. `product_id` `customer_id`

- iv. `access_token` `refresh_token`

3. `inventory_items_list`

UI/API

- **SIM** ID - SIM Card ID
- **Mobile Number** - Mobile Number

Inventory ID `extra_vars` provisioning

```
extra_vars = {
  "product_id": 1,
  "customer_id": 456,
  "SIM Card": 789,           # SIM Card ID 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 linked to this service
    uri:
      url: "{{ crm_config.crm.base_url
      }}/crm/inventory/customer_id/{{ customer_id }}"
      method: GET
      register: inventory_api_response
      ignore_errors: True

  - name: Return inventory to pool
    uri:
      url: "{{ crm_config.crm.base_url
      }}/crm/inventory/inventory_id/{{ item.inventory_id }}"
      method: PATCH
      body_format: json

```

```
    body:
      service_id: null
      customer_id: null
      item_state: "Used"
with_items: "{{ inventory_api_response.json.data }}"
ignore_errors: True

- name: Delete Account from Charging
  uri:
    url: "http://{{ crm_config.ocs.OCS }}/jsonrpc"
    method: POST
    body:
      {
        "method": "ApierV1.RemoveAccount",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}"
        }]
      }
    ignore_errors: True

- name: Delete Attribute Profile
  uri:
    url: "http://{{ crm_config.ocs.OCS }}/jsonrpc"
    method: POST
    body:
      {
        "method": "APIerSv1.RemoveAttributeProfile",
        "params": [{
          "ID": "ATTR_ACCOUNT_{{ service_uuid }}"
        }]
      }
    ignore_errors: True

- name: Remove Resource Profile
  uri: ...
  ignore_errors: True

- name: Remove Filters
  uri: ...
  ignore_errors: True

- name: Deprovision Subscriber from HSS
  uri:
```

```

      url: "{{ item.key }}/subscriber/{{
item.value.subscriber_id }}"
      method: DELETE
      loop: "{{ hss_subscriber_data | dict2items }}"
      ignore_errors: True
      when:
        - deprovision_subscriber | bool == true

- name: Patch Subscriber to Dormant State
  uri:
    url: "{{ item.key }}/subscriber/{{
item.value.subscriber_id }}"
    method: PATCH
    body:
      {
        "enabled": true,
        "msisdn": "9999{{ imsi[-10:] }}", # []
        "ue_ambr_dl": 9999999, # []
        "ue_ambr_ul": 9999999
      }
    loop: "{{ hss_subscriber_data | dict2items }}"
    when:
      - deprovision_subscriber | default(false) | bool ==
false

# []
- name: Set status to "Success" if Manual deprovision /
Fail if failed provision
  assert:
    that:
      - action == "deprovision"

```

[][][][][][][][]

1. []

[][][][][][][][]

- [] [] block [] rescue []
- [] action: "deprovision" [] rescue

2. []

rescue

- OCS
-
-
- HSS
-

“”

-
-
-
-

3. ignore_errors

ignore_errors: True

-
-
-

“ OCS ”

- OCS
-
-
- HSS

4.

rescue

```
- name: Set status to "Success" if Manual deprovision / Fail if failed provision
  assert:
    that:
      - action == "deprovision"
```

□□□□□

- □□ `action == "deprovision"` □□□□□□□□□□□□ 0□
- □□ `action` □□□□ `!= "deprovision"` □□□□□□□□□□□□ 2□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

5. □□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
- name: Deprovision Subscriber from HSS
  uri: ...
  when:
    - deprovision_subscriber | bool == true

- name: Patch Subscriber to Dormant State
  uri: ...
  when:
    - deprovision_subscriber | default(false) | bool == false
```

□□□□□□□□□□

- □□□□□□ SIM □□□□□□□□ `deprovision_subscriber: true` □
- □□□□□□ SIM □□□□□□□□ HSS □□□□ `deprovision_subscriber: false` □

□□□□□□□□

□□□□□□

```
{
  "product_id": 1,
  "customer_id": 456,
  "provisioning_play": "play_psim_only"
}
```

rescue

```
{
  "service_id": 123,
  "service_uuid": "Service_abc123",
  "action": "deprovision",
  "provisioning_play": "play_psim_only"
}
```

rescue

action: "deprovision"

provisioning_json_vars

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

```
{
  "product_id": 1,
  "customer_id": 456,
  "monthly_cost": 45,
  "data_limit_gb": 150
}
```

extra_vars

-
-
-

```
{
  "product_id": 10,
  "product_name": "Extra 10GB Data",
  "provisioning_play": "play_local_data_addon",
  "provisioning_json_vars": "{\"data_gb\": 10}",
  "inventory_items_list": "[]"
}
```

```
extra_vars = {
  "product_id": 10,
  "customer_id": 456,
  "service_id": 123, #
  "data_gb": 10,
  "access_token": "eyJ..."
}
```

API 호출

API 호출 방법

- UI 호출
- REST API 호출
- API 호출 `PUT /crm/provision/`

UI 호출

UI 호출 방법

1. UI 호출

UI 호출 방법

- `provisioning_play` - Ansible 플레이북
- `inventory_items_list` - 인벤토리 항목 목록 ['SIM Card', 'Mobile Number']
- `provisioning_json_vars` - JSON 변수

2. 인벤토리 항목 목록

```
inventory_items_list 인벤토리 항목 목록을 반환합니다. 인벤토리 항목 목록은 인벤토리 항목의 이름과 값을 포함합니다.
```

3. 인벤토리 항목

JavaScript 호출 `PUT /crm/provision/` 호출

```
PUT /crm/provision/
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "product_id": 42,
  "customer_id": 123,
  "SIM Card": 5001,
  "Mobile Number": 5002
}
```

4. API 設定

設定

- Bearer API IP
- CREATE_PROVISION
- initiating_user
-
- OmniCRM-API/Provisioners/plays/play_psim_only.yml

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 Provision_Event 包
- 部署环境部署 0=包2=包3=部署环境部署 JSON
- UI 部署环境部署部署环境部署环境部署
- 部署环境部署部署环境部署

UI 部署

部署环境部署 1部署环境部署

- 部署环境部署 - 部署环境部署部署/部署/部署
- 部署 - 部署环境部署部署
- 部署环境部署 - 部署环境部署部署/部署环境部署

部署

部署环境部署部署部署12 部署环境部署 8 部署

✓ CRM API ✓ ✓ SIM ICCID:
8991101200003204510 ✓ 555-0123 ☞ OCS/CGRateS ...

11. Provisioning

Provisioning

- provisioning_status: 0 - Success
- provisioning_status: 2 - Provisioning failed

UI Provisioning

- Provisioning failed
- Provisioning X
- Provisioning failed provisioning.failure_list

Provisioning

Provisioning

Provisioning

- initiating_user JWT ID
- API ID
- IP ID

Provisioning

Provisioning

- 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 5-10 refresh_token 15-30

```
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

```



```
{
  "method": "ApierV1.SetActionPlan",
  "params": [{
    "Id": "ActionPlan_Service123_Monthly_Charge",
    "ActionsId": "Action_Service123_Add_Monthly_Data",
    "Timing": {
      "Years": [],
      "Months": [],
      "MonthDays": [1],
      "Time": "00:00:00Z"
    },
    "Weight": 10,
    "ActionTriggers": [
      {
        "ThresholdType": "*min_event_counter",
        "ThresholdValue": 1,
        "ActionsID": "Action_Service123_HTTP_Callback"
      }
    ]
  }]
}
```

□□□□□ HTTP POST □□

□□□□□□□□□□□□□□□□ play_topup_no_charge.yaml □□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□

□□ play_topup_monetary.yaml □

```
- name: Mobile Topup Monetary - 2024
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Get Product information from CRM API
      uri:
        url: "http://localhost:5000/crm/product/product_id/{{
product_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
        register: api_response_product

    - name: Get Service information from CRM API
      uri:
        url: "http://localhost:5000/crm/service/service_id/{{
service_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
        register: api_response_service

    - name: Set service facts
      set_fact:
        service_uuid: "{{ api_response_service.json.service_uuid
}}"
        customer_id: "{{ api_response_service.json.customer_id }}"
        package_name: "{{ api_response_product.json.product_name
}}"
        monthly_cost: "{{ api_response_product.json.retail_cost
}}"

    - name: Get Customer Payment Method
      uri:
        url: "http://localhost:5000/api/payments/methods/default?
customer_id={{ customer_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
```

```

    return_content: yes
    register: api_response_payment_method

- name: Charge customer
  uri:
    url: "http://localhost:5000/api/payments/charge"
    method: POST
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "customer_id": "{{ customer_id | int }}",
        "amount": "{{ monthly_cost }}",
        "currency": "USD",
        "metadata": {
          "description": "{{ package_name }} topup",
          "service_id": "{{ service_id | int }}"
        }
      }
    register: api_response_payment

- name: Add monetary balance to OCS
  uri:
    url: "http://{{ crm_config.ocs.OCS }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*monetary",
          "Balance": {
            "Value": "{{ monthly_cost | float * 100 }}",
            "ExpiryTime": "+4320h"
          }
        }
      ]
    status_code: 200

- name: Add Transaction to CRM
  uri:

```

```

url: "http://localhost:5000/crm/transaction/"
method: PUT
headers:
  Authorization: "Bearer {{ access_token }}"
body_format: json
body:
  {
    "customer_id": {{ customer_id | int }},
    "service_id": {{ service_id | int }},
    "title": "{{ package_name }}",
    "retail_cost": "{{ monthly_cost | float }}"
  }
- name: Send Notification SMS
  uri:
    url: "http://sms-gateway/SMS/plaintext/{{ api_key }}"
    method: POST
    body_format: json
    body:
      {
        "source_msisdn": "YourCompany",
        "destination_msisdn": "{{ customer_phone }}",
        "message_body": "Thanks for topping up {{ monthly_cost
        }}"
      }
    status_code: 201
    ignore_errors: True

```

□□□□

1. □ API □□□□□□□□□□□□
2. □□□□□□□□□□
3. □□ Stripe API □□□□□
4. □ OCS □□□□□□
5. □ CRM □□□□□
6. □□□□ SMS□□□ □ignore_errors: True□□□□□□□□□□□□□□

000

0000000000000000OmniCRM 00 00000000000000000000000000

00000000000000

1. 00000000000000000000
2. IPTV 0000000000 1 0 service_id
3. 00 IP 0000000000 1 0 service_id

0000000000

- 0000000000000000 service_id
- 0000000000 extra_vars
- 0000000000 Provision 00

00000000

00000000000000000000000000000000

00000000

0000000000 2

0000000000000000

- API 000000000000
- 0000000000 assert: that: response.status == 200
- 0000000000
- 0000000000
- 0000000000
- OCS/OCS 00

0000000000 3

00000000000000000000

- 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 `stderr` 配置
- 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
- □□□□□□□□□□□□□□□□
- □□□□□□□□ action□□□□□□□□□□/□□
- **IP** □□□□□□□□□□□□□□□□□□

□□

- □□□□□□□□□□ API □□□□□□□□
- □□□□□□ UI □□ 2-5 □□□□□□□□
- □□□□□□□□ Ansible □□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□

□□□□

- `concepts_ansible` - □□ Ansible □□□□
- `concepts_sim_provisioning` - SIM □□□□□□□□ eSIM□
- `concepts_api` - CRM API □□□□□□□□
- `concepts_products_and_services` - □□□□□□□□
- `administration_inventory` - □□□□□□□□
- `□□□□□□` - □□□□□□□□□□

Step 2: Data

Define the data structure for the inventory items.

Example:

Inventory[John ▼]

Item 1 • John Smith (ID: 123) • John Doe (ID: 456) • Johnson Enterprises (ID: 789)

Example:

- Item 1
- Item ID
- Item Name
- Item Price

Example:

Item

Example: `inventory_items_list`

Step 3: Data

Example: `inventory_items_list`

□□□□□□□□

□□□□□□

SIM□ *

□□SIM□□ • SIM-00123 - ICCID□8944...0001□□□ • SIM-00124 - ICCID□
8944...0002□□□ • SIM-00125 - ICCID□8944...0003□□□

□□□□ *

□□□□□ • +44 7700 900123□□□□ • +44 7700 900124□□□□ • +44 7700
900125□□□□

□□□□□□□□

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

□□□□□□

- □□□□□□□□□□□□□□□□

- 00000000000000000000
- 000000“00”0000“0000”

00 4 0000000000000000

0000000000000000000000

0000000000000000

000

[0] [0]

00000000

- 0000000000000000
- 000000000000000000000000

00000 0000000000000“0”000000000000000000000000000000“0”0

00 5 00000000

000000000000000000000000

000John Smith ID 123 000000 - 20GB

000 • SIM 000SIM-00123 ICCID 8944...0001 • 000000+44 7700 900123

000 • 00000£0.00 • 00000£15.00

00000000

[00] [000000]

00**“000000”**0000000000

00 6 00000000

0000000000000000

✓ 000000 ✓ SIM ICCID 8944...0001 ✓ 000000+44 7700 900123 ↻
OCS 000000... 0 000000 0 00000000

00060000030000

00000.200000API00000000

000000

0000

↻ 00000

0 00000000

0000000000

00 7000

000

000000John Smith00

00ID0789 00000000 - +44 7700 900123 000000

[00000] [00]

00**“0000”**00000000000000

000

00000000

00000000OCS

000000000000000000000000000000

00ID0456

[00000] [00] [00]

- 000000000000000000000000
- 0000000000
- 000000000000000000000000???

□□□□

□□□□□□□□□□

□□□□□□□□“addon”□□□□□□□□□□□□□□

1. □□□□□□

- □□□□□□□□□□
- □service_type□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□

2. □□□□□□

- □□□□□□□□service_id□□□□
- □□□□□□□□OCS□□
- □□□□□□□□□□□□

3. □□□□□

- □□□□□□□□□□
- □□□□□□□□□□□□□□□□

□□□

□□□□□□□ • □□ - +44 7700 900123□ID□789□ • □□ - +44 7700 900456□ID□790□

□□□□□□□□□□□□□□ [□□ - +44 7700 900123 ▼]

[□□] [□□]

□□□□□□□

□□□□□□□□□□□

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

□□□□□□□

□□□□□□[□□□□□ - 123 Main St ▼]

□□□□□□[Jane Doe - <jane@acme.com> ▼]

□□□□□□[Bob Smith - <bob@acme.com> ▼]

□□□□□□[PO-2025-001234_____]

[□□] [□□]

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□

- CSV□□□□□□□□
-
-
-

□□□□□□□

□□□□ **1**□□□□□□□

-
- → □□
-
- **“□□ - 20GB□□”**
- **“□□□□□□”**
- “+1234567890”
- John Smith**
- SIM□
-
-
-
-
-

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.

"

-
- -
 -

- 認證碼驗證碼

“認證碼”

- 認證碼驗證碼
- 認證碼
 - 認證碼驗證碼
 - 認證碼驗證碼“*”“*”
 - 認證碼驗證碼

“認證碼”

- 認證碼驗證碼
- 認證碼
 - 認證碼
 - 認證碼ID
 - 認證碼

認證碼

- 認證碼驗證碼
- 認證碼
 - 認證碼5
 - 認證碼
 - 認證碼OCS
 - 認證碼

認證碼驗證碼

- 認證碼OCS
- 認證碼
 - 認證碼
 - 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. 基礎設施即服務 (IaaS)
2. 平台即服務 (PaaS)
3. 軟體即服務 (SaaS)

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的安全
- 身份驗證 - 用戶和設備的身份驗證
- 訪問控制 - 對資源的訪問控制
- 隱私 - 用戶隱私的保護

雲端服務的安全性解決方案

結論

5G 網路和雲端服務的未來

□□□□□

□□□□□

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

□□□□□□

□□ **1**□□□□□□□□

□□□□□□□□□□□□□□□□ “□□□□□”

□□ **2**□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

Modem-12345 - TP-Link AX1800 • Modem-12346 - TP-Link AX1800 • Modem-12347 - Netgear RAX40

3

[] []

- -
- -

4

“”

✓ ✓ ↻ ...

- 1.
2. Ansible (provisioning_play)
- 3.
- 4.
5. OCS
- 6.

Complete Product Lifecycle Guide - Adding Addons
<guide_product_lifecycle>

項目名

項目ID: "項目ID" 項目ID

項目ID

5GB 項目ID

項目ID20250110 項目ID20250117 項目ID 項目ID £5.00

WiFi 6 項目ID

項目ID2024011201 項目ID Modem-12345 項目ID 項目ID £10.00

項目名

項目ID

- 項目ID
- 項目ID "項目ID" 項目ID
- 項目ID "項目ID" 項目ID
- 項目ID

項目ID

- 項目ID - 項目ID
- 項目ID - 項目ID
- 項目ID - 項目ID
- 項目ID - 項目ID

項目ID

項目ID CPE 項目ID

項目名

項目ID 1 項目ID

- 項目ID "項目ID"
- 項目ID

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

- `inventory_items_list`

Inventory Items

- `inventory_items_list`
- `inventory_items`
 - `inventory_items`
 - `inventory_items`
 - `inventory_items` `inventory_items_list`

Inventory

- `inventory`
- `inventory`
 - `inventory` `auto_renew: true`
 - `inventory`
 - `inventory`

Inventory

- `guide_product_lifecycle` - `guide_product_lifecycle`
- `csa_add_service` - `csa_add_service`
- `concepts_products_and_services` - `concepts_products_and_services`
- `administration_inventory` - `administration_inventory`

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□ □□□□ <csa_add_service> □□□□□□□□ <csa_service_usage> □□□□□□□□
<csa_modify> □□□□□□□□ <concepts_products_and_services> □

□□□□□□

□□□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□

- □□ **ID** - □□□□□□
- □□□□ - □□□□□□□□□□“□□ - +44 7700 900123”□

- 網路 - 網路服務
- 網路 - IPTV/VoIP
- 網路 - 網路服務
- 網路 - 網路服務
- 網路 - 網路/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

網路服務

網路

- 網路 - SIM 網路
- 網路 - DSL 網路
- **IPTV** - 網路
- **VoIP** - 網路

□□□□

□□□□□□□□

□□□□□□□□□□□□□□

□□□□□□□□□□□□

□□□ □□□□□□□□□□□□

□□

□□□□□□□□□□□□□□

□□□


- □□□□
- □□ ID
- □□□□
- □□□□□□□□□□□□
- □□ UUID

□□□ □□“0770”□□□□□□□□□□□□□□□□“0770”□□□□

□□□□

□□□□□□□□□□□□□□

□□□□□

- □□ ID□□□□□□□□□□
- □□□□□□□□□□□□
- □□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□
- □□□□□□□□□□□□
- □□□□□□□□

□□□□

- □□□□□□□□A-Z□□□□-□□□□□□-□□□□
- □□□□□□□□Z-A□□□□-□□□□□□-□□□□
- □□□□□□□ ▲ □ ▼ □□□□

□□□□□□□□

□□□□□□□□□□□□□□□□□□

□□□□□□□□

□□□□□□□□□□□□□□□□□□

□□□□□□□□

1. □□ - □□□□□□□□□□□□□□
2. □□ - □□□□□□SIM □□□□□□□□□□
3. □□ - □□□□□□□□
4. □□□□ - □□/□□/□□SMS □□□□
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

消息接收失败

消息接收失败，请检查网络或消息接收状态。

消息接收失败

消息接收失败

消息接收失败

消息接收失败，请检查网络或消息接收状态。

消息接收失败

消息接收失败，请检查网络或消息接收状态。

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

消息接收失败

API Integration

Retrieve scheduled actions for a CGRateS account

- Endpoint - `CGRateS` API `ActionPlan`
- Method - `GET`
- Auth - API Key
- Params - Account ID

Example

Request: `CGRateS` API `ApierV1.GetScheduledActions` CRM Account ID

Response

JSON Response

- Account ID (string)
- Actions (array)

```
{ "AccountID": "Acme Corp FixedWireless_75628fa5", "Actions": [ { "AccountID": "Mega", "Price": "$89.99" }
```

```
], "Timestamp": "2025-11-11 10:45" }
```

```
[ ] [ ]
```

- Use "AccountID" field

4. `CGRateS` `CGRateS`

5. `CGRateS` `CGRateS`

`CGRateS`

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

`CGRateS`

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

`CGRateS`

`CGRateS`

1. `CGRateS` `CGRateS` `CGRateS`

2. `CGRateS`

`CGRateS`

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

`CGRateS`

[code] [code]

3. `CGRateS` “`CGRateS`” `CGRateS`

4. ActionPlan `CGRateS` `CGRateS`

5. 0000000000000000

000000

- ActionPlan 0 CGRateS 000
- 0000000000
- 00000000000000
- 0000000000000000

000

- 0000000000
- 0000000000
- 0000000000
- 0000000000

00

000

0000000

0000

- 00000000000000
- 0000000000
- 000000000000000000000000 "N/A"

0000

- 0000000/UUID
- 0000000000
- 0000000000000000 UUID
- 000000000000000000000000 "N/A"

0000

- 00 CGRateS ActionPlan 000000

- 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 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

◀ 查詢 1 [2] 3 4 5 查詢 ▶

查詢[10 ▼]

- 10

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

CGRateS 簡介

CGRateS 是一個用於記錄和處理呼叫記錄的系統。

每個 ID 都有一個 ATTR_ACCOUNT_{service_uuid}

每個 ID 的 UUID 是 Mobile_SIM_c2880638，每個 ID 的 ATTR_ACCOUNT_Mobile_SIM_c2880638

安裝步驟

1. 安裝 CGRateS
2. 設置 “CGRateS 服務”
3. 配置 “數據庫”
4. 配置 “系統參數”
5. 設置 “用戶名”

□□□□

- □□ - □□□□□□□□□□ *req.Account □ *req.IMSI □
- □□ - □□□□□□

- `*constant` - 常數
- `*variable` - 由 RSRParser 解析的變數
- `*composed` - 組成的規則
- `*usage_difference` - 使用量差值
- `*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
- 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
- 項目 CGRateS 項目
- 項目
- 項目 JSON 項目

□□□

- □□□□□□□□□□
- □□□□□□□□

- 000000000000
- 0000000000
- 000000000000

0000

000000000000000000000000"00"00000000000000

00

0000000000000000 UUID000000000000 `*string:~*req.Account:{service_uuid}` 0000
00

CGRateS API 00

00 CGRateS 00000000000000000000 OmniCRM API 000000

0000 `POST /crm/ocs/proxy`

000000

- `method` - CGRateS API 00000000 `APIerSv1.GetAttributeProfile` 0
- `params` - 00000000
- `customer_id` - 00 ID00000000000000
- `service_id` - 00 ID00000000000000

000000

- `tenant` - CGRateS 00000000000000

000000

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


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

○○○○○○○○

○○ OCS○○○○○○○○○○○○○○○○○○○○ Self-Care Portal <self_care_portal>○○

○○

○○○○○○○○

- ○○○○○○ - ○○○○○○○○○○○○○○○
- ○○○○○○ - ○○○○○○○○○○○○○○○○○○○
- ○○○○ - ○○○○○○○○○
- ○○○○ - ○○○○○○○○○○○
- ○○○○ - ○ 3 ○○○○○○○○○○○

○○○○○○○○

○○○○○○

1. ○○○ ○○ → ○○○○
2. ○○○○○○○○○○○ ○○○○ (:)
3. ○○ "○○○○"

○○○○○○

1. ○○○○○○○○○○○
2. ○○ "○○○○" ○○

○○○○○○

1. ○○○○○○
2. ○○○ ○○ ○○
3. ○○○○○○○○○○○ "○○○○○○○"

○○○○○○○○○○○○○○○○○○○○○○○○○

□□□□□□

□□□□□□□□□□□□□□□□□□□□

□□□

□□□□□□□□□□□□□□□□

□□□□

- 伺服器 - 伺服器
- 伺服器 - 伺服器
- 伺服器 - 伺服器/網
- 伺服器 - 伺服器

伺服器

伺服器

伺服器

- 伺服器 - 伺服器
- 伺服器 - 伺服器
- 伺服器 - 伺服器“12.5GB / 20GB”
- 伺服器
 - 伺服器>50% 伺服器
 - 伺服器20-50% 伺服器
 - 伺服器<20% 伺服器
- 伺服器 - 伺服器

伺服器

伺服器

伺服器

伺服器 (GB) 伺服器 (MB)

伺服器

伺服器12.5GB / 20GB (62%)

伺服器

- 伺服器 - 5GB 10GB 20GB 伺服器
- 伺服器 - 伺服器500GB 1TB

45 / 250 (18%)

Options

- 160 options
- [unclear]
- [unclear]

Options

[unclear]

Price (£, \$, €, ¥)

Options

£15.50 / £20.00 (77%)

Options

- [unclear]
- [unclear]
- [unclear]
- [unclear]

???

"[unclear]" [unclear]

Options

20GB

12.5 GB 7.5 GB 2025 1 25 15 10

5GB

5.0 GB 0 GB 2025 1 31 21 20

□ □□□ 3GB

□□□0 GB □□□3.0 GB □□□2025□1□18□□□□□ □□□30

□□□□17.5 GB

□□□□□

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

□□□□□

□□□□□□□□□□□□□□□□

- □□□ = □□□□
- □□□ = □□□□

□□□□□

- □□□□□□□ 30□□□□□□□□□□□□□□□□
- □□□□□□□ 20□□□□□□□
- □□□□□□□ 10□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□ CGRateS □□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□ **CGRateS Actions and Topup Behaviors**□

□□□□□

□□□□□ **3** □ □□□□□□□□□

□□□□□

- □□□□□□□

- 00000
- 00000
- 000000000

00000

- 000000000
- 000000000
- 000000000
- 000000000000000

000

- 000000000000000
- 000000000000000000
- 000000000

00000000000000000000

00000

000000000000000

- 00 (GB)
- 00 (00)
- 00 (00)
- 00 (00)

000

| 00012.5GB 00 000125 min 00 00045 msg 00 000£15.50 00

00000/0000

000000

- 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
- 接続速度
 - 接続速度 OCS
 - OCS API
 - UUID

接続

- 接続速度 OCS
- 接続速度
 - 接続速度
 - 接続速度
 - 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 碼

查詢服務

- 查詢 服務狀態
- 查詢
 - 查詢服務狀態
 - 查詢 OCS 碼

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 配置と UI 表示

- `*data` → 通信料
- `*voice` → 通話料
- `*sms` → SMS 料
- `*monetary` → 課金

課金項目 + 単位 + 課金率

例

- 通信料 → GB 1024³
- 通話料 → 分 60×10^9
- SMS 料
- 課金 → 円 100

設定

OCS 設定 [CGRateS Actions and Topup Behaviors](#)

例

CGRateS 設定

- 課金
- 単位
- 課金率
- 課金

例

1. 設定

- `auto_renew = true`
- OCS 設定
- 課金

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

□□□□

□□□□□□□□□□□□□□□□

- □□□□□□□□
- □□□□□□□□□□

□ **UI** □□□□

□□□□□□□□□□□□

| □□□□2025□2□1□□□22□□□ □□□□£15.00

□□□□

□□□□□□

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

□□□□□□□□□□

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

□□□□□□

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

RBAC

RBAC Overview

RBAC is a security model that uses roles to define permissions. Roles are assigned to users and permissions are assigned to roles.

Example: A user is assigned the role of `Self-Care Portal <self_care_portal>`, which grants them access to the Self-Care Portal.

Example: The `RBAC <rbac>` role is assigned to the `Customers <basics_customers>` role, which grants them access to the `Self-Care Portal <self_care_portal>`.

Steps

1. Create Roles

1. Create Role - Define permissions and scope
2. Assign Role - Assign role to user or group
3. Assign User - Assign user to role
4. Assign Group - Assign group to role
5. Assign Admin - Assign admin to role

Permissions

Permissions are defined for each role.

- `can_impersonate` - Allows user to impersonate other users
- `admin` - Grants full administrative access

Permissions can be inherited from parent roles.

□□□□□□

□□ **Web UI** □

1. □□□□ — □ CRM □□□□
2. □□□□ — □□□□□□□□
3. □□“□□□□□□” — □□□□□□□□□□□□□□□□
4. □□□□ — □□□□□□□□
5. □□□□ — □□□□□□□□□□

□□ **API** □

□□□□□□

□□□ `POST /auth/impersonate`

□□□□ `can_impersonate` □ `admin`

□□□

```
{  
  "user_id": 42  
}
```

JSON

```
{
  "success": true,
  "impersonating_user_id": 1,
  "target_user_id": 42,
  "impersonation_start": "2025-01-04T15:30:00Z",
  "access_token": "new_token_for_impersonated_user",
  "refresh_token": "new_refresh_token"
}
```

JSON 格式

JSON 格式

JSON 格式

- 成功 — 成功返回 JSON 格式
 - 成功
 - 成功
 - 成功
 - 成功
- 参数 — 参数
 - `impersonating_user_id` — 用户 ID
 - `target_user_id` — 目标用户 ID
 - `impersonation_start` — 开始时间
- 消息 — UI 消息
 - "成功 [消息] 完成"
 - "成功"
- 日志 — 日志
 - 成功
 - 成功
 - 在 `ImpersonationLog` 中记录

API

Web UI

1. 停止 — 停止 impersonation
2. 停止 — 停止 impersonation “停止”
3. 停止 — 停止 impersonation

API

POST /auth/stop_impersonation

成功時返すレスポンス

レスポンス

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

失敗時返すレスポンス

API

成功時返すレスポンス `ImpersonationLog` オブジェクト

- **impersonating_user_id** — impersonating ユーザー ID
- **target_user_id** — target ユーザー ID
- **impersonation_start** — impersonation 開始日時
- **impersonation_end** — impersonation 終了日時
- **impersonation_duration** — impersonation 継続時間

失敗時返すレスポンス

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

API

API endpoint

```
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. `id` — `id`
2. `user_id` — `user_id`
3. `start_date` — `start_date`
4. `end_date` — `end_date`
5. `impersonation_logs` — `impersonation_logs`
6. `can_impersonate` — `can_impersonate`
7. `impersonation_logs` — `impersonation_logs`

Example

- `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“00000000”000000

- 000000000000 `can_impersonate` 0 `admin` 00
- 000000000000000000000000

0000000000000000

- 0000000000000000000000000000
- 0000000000000000000000

0000000000

- 00000000 API 00000000 `POST /auth/stop_impersonation`
- 000000000000 cookies 000000000000

000000000000

- 0000000000000000000000000000
- 000000000000 - 0000000000 ID

📖📖📖📖 CGRateS 📖📖

📖📖📖📖📖 OmniCRM 📖📖📖📖📖📖 CGRateS 📖📖📖📖

📖📖📖 CGRateS 📖📖📖 Python 📖📖📖📖📖📖📖📖📖 CRM 📖 OCS📖📖📖📖📖📖

📖📖

📖📖📖📖📖📖📖📖📖📖📖📖

1. **CGRateS/OCS** - 📖📖📖📖📖📖📖📖📖📖📖📖
2. **OmniCRM** - 📖📖📖📖📖📖📖📖📖📖📖📖

📖📖📖📖📖📖📖📖📖📖📖📖 Ansible Play 📖 CGRateS 📖📖📖📖

📖📖📖📖CGRateS 📖📖

CGRateS 📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖 OmniCRM 📖📖📖 CGRateS 📖📖📖📖📖📖

- 📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖
- 📖📖📖📖📖📖 MO 📖 MT 📖/SMS 📖📖📖📖📖📖📖📖📖
- 📖📖📖📖📖📖 2G/3G 📖📖 CAMEL📖MO-MT SMS📖USSD📖📖📖📖📖📖📖📖📖 LTE/5G 📖 Diameter
- 📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖
- 📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖 LCR📖📖📖📖📖📖

📖📖 **CRM** 📖📖 - 📖📖📖📖 OmniCRM 📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖📖 CGRateS 📖📖📖📖📖

[CGRateS 📖📖](#) 📖📖 [CGRateS 📖📖📖📖📖📖](#) 📖📖📖📖📖📖📖📖📖📖

Python Requests

Part 1 CRM API Authentication

```
import requests
import json

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

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

# Authentication
response = session.post(crm_url + '/crm/auth/login', json={
    "email": "admin@example.com",
    "password": "your_password"
}, headers=headers)

assert response.status_code == 200
headers['Authorization'] = 'Bearer ' + response.json()['token']
print("CRM Authentication Successful")
```

Part 2 API Calls

Authentication successful. Now we can use the session to make API calls. For example, we can use the session to make a GET request to the CRM API to retrieve a list of contacts.

```

inventory_list_new = []

# SIM
inventory_list_new.append({
    "item": "SIM Card",
    "itemtext1_label": "ICCID",
    "itemtext2_label": "IMSI",
    "wholesale_cost": 0.2,
    "retail_cost": 1,
    "allow_dropdown_staff": True,
    "allow_dropdown_customer": True,
    "icon": "fa-solid fa-sim-card"
})

# 
inventory_list_new.append({
    "item": "Mobile Number",
    "itemtext1_label": "E.164 Mobile Number",
    "itemtext2_label": "Type",
    "wholesale_cost": 0.2,
    "retail_cost": 3,
    "icon": "fa-solid fa-phone"
})

# 
for inventory_template in inventory_list_new:
    # 
    inventory_list_existing = session.get(
        crm_url + '/crm/inventory/item_template/',
        headers=headers
    )

    # 
    if inventory_template['item'] in [x['item'] for x in
inventory_list_existing.json()]:
        # 
        inventory_template_id = [
            x['inventory_template_id']
            for x in inventory_list_existing.json()
            if x['item'] == inventory_template['item']
        ][0]

        response = session.patch(

```

```

        crm_url + '/crm/inventory/item_template/' +
str(inventory_template_id),
        json=inventory_template,
        headers=headers
    )
    assert response.status_code == 200
    print(f"👉👉👉👉: {inventory_template['item']}")
else:
    # 🐞🐞🐞🐞
    response = session.put(
        crm_url + '/crm/inventory/item_template',
        json=inventory_template,
        headers=headers
    )
    assert response.status_code == 200
    print(f"👉👉👉👉: {inventory_template['item']}")

```

👉👉👉👉

- **item**👉👉👉 - 🐞🐞🐞
- **itemtext1_label** - 🐞🐞🐞🐞🐞🐞🐞🐞
- **itemtext2_label** - 🐞🐞🐞🐞🐞🐞🐞🐞
- **wholesale_cost** - 🐞🐞🐞
- **retail_cost** - 🐞🐞🐞🐞🐞
- **allow_dropdown_staff** - 🐞🐞🐞🐞🐞🐞🐞🐞
- **allow_dropdown_customer** - 🐞🐞🐞🐞🐞🐞🐞🐞
- **icon** - Font Awesome 🐞🐞

🐞 3🐞🐞🐞 **CGRateS**

```

import cgrateshttpapi
import time

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

tenant = "your_tenant_name"
tpid = str(tenant) + "_" + str(int(time.time()))

```

4 CGRateS

1. - /SMS Dest_AU_Mobile Dest_International_UK
2. **PLMN** - Dest_PLMN_OnNet Dest_PLMN_US_Verizon

- /SMS →
- → PLMN

-
- PLMN
- PLMN mccXXX.mncYYY
-

CGRateS -

5 CGRateS

CRM CGRateS

CGRateS

- Python OCS_Obj.SendData
- *topup *topup_reset
- SMS
- Balanceld BalanceType DestinationIDs Units ExpiryTime Weight
-
-

□□□□□□

- ***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", # 澳洲 Action_au-premium-
plan-1
    "product_name": "AU 澳洲 1",
    "comment": "澳洲 100GB 通话3000 短信3000 6GB 流量",
    "provisioning_json_vars": "",
    "terms": "澳洲 30 天",
    "service_type": "mobile",
    "residential": True,
    "business": True,
    "enabled": True,
    "icon": "fa-solid fa-mobile",
    "features_list": "['澳洲 100GB 流量', '澳洲 3000 短信', '澳洲 3000 短信 SMS', '6GB 流量']",
    "wholesale_cost": 45.00,
    "wholesale_setup_cost": 0,
    "auto_renew": "True",
    "customer_can_purchase": True
})

```

澳洲 100GB 流量

```

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", # 000 Action_au-data-
addon-20gb
    "product_name": "AU 0000 20GB",
    "comment": "00000 20GB 00",
    "provisioning_json_vars": "",
    "terms": "000000000000",
    "service_type": "mobile",
    "residential": True,
    "business": True,
    "enabled": True,
    "icon": "fa-solid fa-mobile",
    "features_list": "['00000 20GB 00']",
    "wholesale_cost": 22,
    "wholesale_setup_cost": 0,
    "auto_renew": "False",
    "customer_can_purchase": True
})

```

00000000

- **category**0000 - 00000
 - standalone - 000000000000000
 - addon - 00000000
 - bundle - 0000000
- **provisioning_play**0000 - 0000 Ansible playbook0000
 - play_topup_charge_then_action 0 play_simple_service 0
- **relies_on_list** - 0000000 product_slugs 00000000
- **contract_days** - 000000000 = 000030 = 000
- **retail_cost**0000💎💎 - 00000000/00000
- **retail_setup_cost** - 000000000000000
- **product_slug**0000 - 000 **CGRates** 000000000000000 Action_{product_slug}0
- **product_name**0000 - 00000

- **comment** - 文字列
- **provisioning_json_vars** - ansible playbook 用の JSON 変数
- **terms** - 利用規約
- **service_type** - サービスの種類
- **residential** - 住宅用
- **business** - ビジネス用
- **enabled** - 有効/無効
- **inventory_items_list** - ansible Python 変数
- **icon** - Font Awesome アイコン
- **features_list** - ansible Python 変数
- **wholesale_cost** - 小売価格
- **wholesale_setup_cost** - 小売設置価格
- **auto_renew** - 自動更新 "True" か "False" 指定
- **customer_can_purchase** - 顧客が購入できるかどうか

ansible-playbook

7 API

```
#
product_list_existing = session.get(crm_url + '/crm/product',
headers=headers)
existing_products = product_list_existing.json()

#
for product in product_list_new:
    print(f": {product['product_slug']}")

    #
    if product['product_slug'] in [x['product_slug'] for x in
existing_products]:
        print(f": {product['product_slug']}")

        # product_id
        product_id = [
            x['product_id']
            for x in existing_products
            if x['product_slug'] == product['product_slug']
        ][0]

        product['product_id'] = product_id

    #
    response = session.patch(
        crm_url + '/crm/product/' + str(product_id),
        json=product,
        headers=headers
    )
    print(f": {response.status_code}")
    assert response.status_code == 200
    print(f": {product['product_slug']}")
else:
    #
    print(f": {product['product_slug']}")
    response = session.put(
        crm_url + '/crm/product',
        json=product,
        headers=headers
    )
    print(f": {response.status_code}")
```

```
assert response.status_code == 200
print(f"response: {product['product_slug']}")
```

CRMSession

CRM CGRateS API API

```
# CGRateS
ActionsId = "Action_au-premium-plan-1"

# CRM
product_slug = "au-premium-plan-1"

# playbook
cgr_action_name = "Action_" + product_slug
# : "Action_au-premium-plan-1"
```

Playbooks

ExecuteAction playbook SetBalance
CGRateS API OmniCRM-
API/Provisioners/plays/play_topup_charge_then_action.yaml
ExecuteAction Playbooks

product_slug

```

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

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

```

CGRateS " " "

Ansible Playbooks

"50GB " "

1. CGRateS

5 " " 50GB " "

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

1. 创建 "50GB 套餐" product_slug: 50gb-data-pack
2. Playbook play_topup_charge_then_action
3. Playbook 创建 Action_50gb-data-pack
4. Playbook 调用 APIerSv1.ExecuteAction 调用 ActionId Action_50gb-data-pack
5. CGRateS 创建
6. 创建 50GB 套餐

创建 Playbooks

创建

创建 CRM 创建 CGRateS 创建 CGRateS

创建 1

创建 SMS

CRM

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

创建 *topup_reset 创建 Python

002 0000000000000000

000 000000000000000000

CRM 000

```
{
  "product_slug": "10gb-addon", # 000 Action_10gb-addon
  "product_name": "10GB 0000",
  "category": "addon",
  "retail_cost": 15.00,
  "auto_renew": "False", # 00000
  "contract_days": 0,
  "provisioning_play": "play_topup_charge_then_action",
  "features_list": "['10GB 0000000', '30 0000']"
}
```

000 00 *topup 000000000000 CGRateS 00 - 000

003 0000000000000000

000 000000000000000000

CRM 000

```
{
  "product_slug": "fixed-50gb-monthly", # 000 Action_fixed-50gb-monthly
  "product_name": "00 50GB 0000",
  "category": "addon",
  "retail_cost": 35.00,
  "auto_renew": "True",
  "contract_days": 30,
  "provisioning_play": "play_topup_charge_then_action",
  "features_list": "['50GB 00', '0000', '000']"
}
```

000 00 *topup_reset 0000000000 CGRateS 00 - 00000

□□□□□□□□□□

□□□□ □□□□□□□□□□□□CGRateS □□□□□□□□□□□□

- □□□□□□□□□□ MO □ MT □□/SMS □□□□□□
- □□□□□□□□□□ 2G/3G □ CAMEL□MO-MT SMS□USSD□□□□□ 4G/5G □ Diameter
- □□□□□□□□□□□□□□□□□□□□□□□□
- ❖❖❖□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□ OmniCRM □□□□□□□□□□□□ PAYG □□□

□□□□□□

□□□□□□□□□□□□□□□□SMS□□□

1. CGRateS □□□□□□□□□□□□□□□□□□□□□□ GB□
2. □□□□□□□□□□□□□□□□□□□□□□
3. □□□□□□□□□□□□□□□□□□□□□□□□CGRateS □□□□□□□□□
4. □□□□□ □□□□□□□□\$0.10 □ MB□
5. □□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□

```
# 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TPRatingProfile時，APIOptions必須設定為 `*monetary`
- 設定TPRatingProfile時，RatingPlanId必須設定為 **1 MB**


```
{
  "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)

# data1
"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 = 700
{"Identifier": "*reset_account", "Weight": 700}
{"Identifier": "*topup_reset", "Weight": 90}
{"Identifier": "*cdrlog", "Weight": 80}

# 余额权重 = 1200
"BalanceWeight": 1200 # 100/1000
"BalanceWeight": 1100 # 1000
"BalanceWeight": 1000 # 1000

```

7. 计费策略

计费策略配置

```

# 100MB
Units = 100 * 1024 * 1024 # 100MB 计费

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

```

计费策略

计费策略配置

计费策略 Playbook 配置 "计费策略" 的 "ActionsId 配置"

计费策略 CGRateS 配置

计费策略

1. 计费策略 CGRateS 配置
2. 计费策略 `ActionsId = "Action_" + product_slug`
3. 计费策略

查詢“查詢範圍”

查詢 查詢範圍

查詢 DestinationIDs 查詢範圍

查詢

1. 查詢 CGRateS 查詢範圍
2. 查詢範圍 *any
3. 查詢 ID 查詢

查詢範圍

查詢 查詢範圍

查詢 查詢範圍

查詢

1. 查詢 *topup 或 *topup_reset 查詢
2. 查詢範圍 0
3. 查詢範圍

查詢

查詢 查詢範圍

查詢 查詢 *topup_reset 查詢 *topup

查詢

- 查詢 Identifier 查詢 *topup_reset 查詢 *topup
- 查詢 Balanceld 查詢範圍

查詢 CGRateS 查詢範圍



OmniCRM
AMBER

SMS

OmniTouch OWL

OmniTouch OWL

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

OWL Web Chrome/Firefox/Safari/Edge



PWS

-
-
- 500
-
- CBC
- TOTP
-

SMS 10

500

"

-
-
-
-
-

- -
- -

2

OWL

- -
- -
- -

“”

3

TOTP

- RSA SecurID
- Google Authenticator, Authy, Microsoft Authenticator, TOTP

2FA

2FA

1. Google Authenticator, Authy, Microsoft Authenticator
2. OWL QR
- 3.
- 4.
- 5.

2FA <2fa>

OWL → 2FA

-
-

- 语言 - 语言包
- 语言 - 语言包
- 语言 - 语言包

语言

CBC语言包

- 语言 - ISO语言包en es fr zh
- 语言 - 语言包1395语言包

语言包

语言

语言包

- 语言 - 语言包ID语言包
- 语言 - 语言包MCC-MNC
- **RAT**语言 - 语言包LTE 5G UMTS GSM

语言包

语言包

Web UI

1. 语言包 - 语言包CBC语言包
2. 语言“语言” - 语言包
3. 语言包
 - 语言包4370
 - 语言包
 - 语言包5-60
 - 语言包999语言包
 - 语言包
 - 语言包“语言”

4. 國際化

- 雙“引號”
- 國際化
- 國際化GSM7 1395 Unicode
- 國際化

5. 國際化

- 雙“引號”
- 國際化
- 國際化MCC-MNC
- 國際化RAT LTE 5G
- 國際化

6. □□□□ - □□□□□□□□“□□□□”

□□**API**□

□□□ `PUT /crm/cbc/`

□□□□ `CREATE_CBC_MESSAGE`

□□□□

```
{
  "messageIdentifier": "4370",
  "category": "emergency",
  "repetitionPeriod": 10,
  "numberOfBroadcasts": 999,
  "warningPeriodSec": 3600,
  "channelIndicator": "basic",
  "localized_messages": [
    {
      "language": "en",
      "messageBody": "EARTHQUAKE WARNING: Magnitude 6.5 earthquake
detected. Take cover immediately. Drop, Cover, Hold On."
    },
    {
      "language": "es",
      "messageBody": "ADVERTENCIA DE TERREMOTO: Terremoto de
magnitud 6.5 detectado. Cúbrase inmediatamente. Agáchese, Cúbrase,
Agárrese."
    }
  ],
  "tracking_areas": [
    {
      "tracking_area": "12345",
      "operator": "310-410",
      "rat_type": "LTE"
    },
    {
      "tracking_area": "12346",
      "operator": "310-410",
      "rat_type": "5G"
    }
  ]
}
```

□□□

```
{
  "cbc_message_id": 123,
  "cbc_unique_id": "550e8400-e29b-41d4-a716-446655440000",
  "messageIdentifier": "4370",
  "category": "emergency",
  "repetitionPeriod": 10,
  "numberOfBroadcasts": 999,
  "warningPeriodSec": 3600,
  "channelIndicator": "basic",
  "initiating_user": 5,
  "approving_user": null,
  "created": "2025-01-10T14:30:00Z",
  "localized_messages": [...],
  "tracking_areas": [...]
}
```

□□□□□□□□□□□□□□□□□□

□□□□□□□

□□□□□□

□□□ GET /crm/cbc/

□□□□□ VIEW_CBC_MESSAGE

□□□□CBC□□□□□□□□□□□□□□□□□□□

□□□□

□□□ PATCH /crm/cbc/{cbc_message_id}

□□□□□ UPDATE_CBC_MESSAGE

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□CBC□

□□□□□

如何配置CBC

如何配置CBC的URL

OWL CBC

CBC URL在crm_config.yaml中

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

如何

如何配置CBC

1. OmniCRM CBE
2. 如何配置CBC API
3. 在{cbc_url}/alerts/send中发送HTTP POST
4. CBC的CBSP SBc-AP N50
5. CBC
6. 如何配置CBC

如何

如何配置CBE的CBC

如何

如何配置2G/3G/4G/5G的CBC

OWL的2G 3G 4G 5G

CBSP - 2G/3G - 如何配置BSC

如何配置CBSP的CBC 2G GSM BSC

- 如何配置2G 3G
- 如何配置BSC

- 3GPP RNC/BSC OWL CBC
- CBSP

3GPP RNC/BSC SABP RNC/BSC CBSP RNC

SBc-AP - 4G/5G - MME/IWF

SBc-AP OWL CBC 4G/5G eNodeB/gNodeBs MME

- 4G LTE
- 5G 2025
- MME OWL CBC
- SBc-AP

N50 - 5G - AMF

5G N50 OWL CBC 5G gNodeBs AMF

- OWL CBC
- 2025 5G SA AMF
- 5G SA

MNO

OWL CBC

- CBC MNO
-
- MME RNC BSC

OWL CBC

OWL  

- **NetAct**
- **U2000 / U2020**
- **NetNumen / ZXPOS**
- **ENM**

OmniTouch

RBAC

OWL RBAC



RBAC

- - OWL
- -
- -
- -

RBAC

1. -
2. -
3. -

CRUD

- -
- -
- -  

- `role` - `role`

CBC

- `CREATE_CBC_MESSAGE` - `role`
- `VIEW_CBC_MESSAGE` - `role`
- `UPDATE_CBC_MESSAGE` - `role`
- `DELETE_CBC_MESSAGE` - `role`

`role`

`role` `<rbac>`

role

`role`

`role`

1. `role` URL
2. `role`
3. `role` 2FA `role` 2FA

`role`

`role` `<authentication_flows>`

`role`

`role`

1. `role` "role"
2. `role`
3. `role`

`role`

`role` → `role`

1. `role` → `role`

OWL

→

OWL

-
-
-
-
-
-

-
- NDMO
-
-

□□□□

□□□□ VIEW_CBC_MESSAGE □ CREATE_CBC_MESSAGE

□□□□

□□□□

□□□□

- □□□□ - □□□□□□□□□□
- □□□□□ - □□□□□□□□□□
- □□□□ - □□□□□□□□
- □□□□ - □ cellSites.txt □□□□

□□□□

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

□□□□

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

□□□□

□□1□□□□□□□

□□□□□□□□□□□□□□□□ /cellSites.txt □□□□□□□

- **lat/lng** - □□□□
- **site_name** - □□□
- **sectors** - □□□□□□□□□□

□□2□□□□□□□

□□□□□□

1. □□□□□□□□□□

- □□□□
- □□□□
- □□□□□□□□

2. □□□□□□□□□□

- □□□□
- □□□□□□
- □□□□□□□□□□

3. □□□□□□□□□□

- □□□□□□
- □□□□□□
- □□□□□□□□

□□**3**□□□□□□□□

□□□□□□

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

□□□□□□

□□**4**□□□□□□

□□□□□□□□

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

□□□□□□

□□□□□□□□□□□□

□□□□□

1. □□**CBC**□□□□□□□□□□□□
2. □□□□□□□□□□□□□□
3. □□□□□□□□□□□□□□
4. □□□□□□□□□□□□□□□□
5. □□**CBC**□□□□□□□□□□□□

□□□□□□□□□□

□□□□□

□□□□□

□□□□□□□□□□□□□□□□□□

- □□□□ - □□□□/□□□□□□
- □□□□ - □□□□□□□□
- □□□□ - □□□□□□□□□□**AMBER**□□
- □□□□□□□□ - □□□□/□□□□□□

□□□□□□□□

□□□□□ `site_data.json` □□

```
[
  {
    "area": "Coastal Region",
    "polygon": [
      {
        "coord": [
          {"lat": -33.8688, "lng": 151.2093},
          {"lat": -33.8650, "lng": 151.2070},
          {"lat": -33.8600, "lng": 151.2150},
          {"lat": -33.8688, "lng": 151.2093}
        ]
      }
    ]
  }
]
```

□□□□□□□□

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

□□□□

Google Maps API

□□□□□□□□□□Google Maps API□□□

```
REACT_APP_GOOGLE_API_KEY=your_google_maps_api_key_here
```

□□□□□

□□□□□□□□□□□□ public/cellSites.txt □

```
# [lat,lng,name,sector1,sector2,sector3  
-33.8688,151.2093,SYD_CENTRAL,0,120,240  
-33.8650,151.2070,SYD_HARBOUR,45,165,285
```

□□□□□□□□□□

- □□□□□□
- □□□□□□
- □□GPS□□□□□□
- □□□□□□□□RNC□□□□

□□□□□□

□□□□ `mapStyles.js` □□□□□□□□□□□□□□□□

- □□□□□□
- □□□□□□
- □□□□□□
- □□□□□□

□□

□□□□□□

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

□□□□□□

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

□□□□□□

- 繁體中文
- 簡體中文
- 繁體中文(台灣)
- 簡體中文(大陸)

繁體中文

繁體中文(台灣) Omnitouch 繁體中文(大陸) OWL 繁體中文

繁體中文

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

繁體中文

- 繁體中文(台灣) 360 繁體中文
- 繁體中文(大陸) 360 繁體中文
- 繁體中文(台灣) 360 繁體中文
- 繁體中文(大陸) Unicode 繁體中文

繁體中文

- 繁體中文(台灣) 360 繁體中文
- 繁體中文(大陸) 360 繁體中文
- 繁體中文(台灣) 360 繁體中文
- 繁體中文(大陸) 360 繁體中文

繁體中文

- 繁體中文(台灣) 4379 4380-4381 繁體中文
- 繁體中文(大陸) 4379 4380-4381 繁體中文
- 繁體中文(台灣) CBC 繁體中文

- 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 Omnitouch 2FA 2FA

2FA 2FA GIS Google Earth Omnitouch 2FA 2FA

2FA

2FA 2FA 2FA 2FA

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

- 0000000000000000
- 0000000000000000

0000000000000000 Omnitouch 0000000000000000

000000

- 0000000000000000
- 0000000000000000
- 0MNO00000000000
- 0000000000000000

000000

00 Omnitouch 0000000000000000

000000

000000000000

- 000 **K8s** - Kubernetes 0000000000
- 000 - VMware 0 Proxmox 0 HyperV
- 000 - 0000000000
- 000 - AWS 0 GCP
- 000 - 00000000

000000

0000000000

- 000000000000 OWL CBE 00000000000000000000
- 000000/00000000000000000000000000
- 000000000000000000000000 NDMO 00000000000000000000000000000000

0000000000000000000000 BSC/MME 000000000000

CBE 0 CBC 00

|||||MNO|||CBC

CBE|CBC

- CBE|CBC
-
- CBE|CBC|TCP|443|TLS
- |||MNO|NDMO|||CBE

|||

|||CBE|||Citrix|VPN|||

-
-
-

|||

CBE|NDMO / |||

3|||

- 2|**CBE**|/|||
- 1|||

|||

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

CBC MNO

3

- 2 CBC
- 1

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

OWL

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

16. 00000000 - 0000000000

0000

00000000000000000000

GSM 70000000000000000

- 00093000
- 000150 × 93 = 001395000

Unicode UCS-20000000000000000

- 00041000
- 000150 × 41 = 00615000

OWL000

- 00000000500000
- Web UI00000000000000000000

000000

0000000000000000

- 00000000000000
- 00000000
- 0000000000
- CBC API000000
- 00000000

0000000000000000

```
SELECT * FROM cbc
WHERE created >= '2025-01-01'
ORDER BY created DESC;
```

□□□□□□□□

□□□□□□□□□□□□□□□□

□□□□□

- □□□2015□□□□□□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□□□□□/SMS□□□□
- □□□□□□□□□□SMS□□□□□□□□

□□□□□

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

□□□□□□□□

□□□□□□□□□□□□□□□□

1. □□□□□□□□□□□4379□4380-4381□
2. □□□□□□□□□□□□□□□□
3. □□□□□□□□□□□□□□□□□□□□
4. □□□□□□□□□□□□□□□□□□□□

□□□□□

OWL□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□/SMS

□□□□□□□□SMS□□□□□□□□□□

- 2019年10月1日起，所有在境內經營的企業，均應按照《企業會計準則第14號——收入》的規定，對收入確認進行會計處理。
- 2019年10月1日起，所有在境內經營的企業，均應按照《企業會計準則第14號——收入》的規定，對收入確認進行會計處理。
- 2019年10月1日起，所有在境內經營的企業，均應按照《企業會計準則第14號——收入》的規定，對收入確認進行會計處理。

2019年10月1日起

2019年10月1日起

- 2019年10月1日起
- 2019年10月1日起
- 2019年10月1日起
- 2019年10月1日起

2019年10月1日起

2019年10月1日起

- 2019年10月1日起
- 2019年10月1日起
- 2019年10月1日起

2019年10月1日起

2019年10月1日起

- 2019年10月1日起
- 2019年10月1日起
- 2019年10月1日起
- 2019年10月1日起

2019年10月1日起

OWL CBC 2019年10月1日起

- 2019年10月1日起
- API

- 000000
- 0000
- 00000000
- 0000

0000000000000000

00000000

0000

0000000000000000

- 00000000000000
- 00000000000000000000000000000000
- 00000000
- 00000000000000

000000

- 000000437904380-43810000000000
- 000000000000000000000000000000
- 000000000000000000
- 00000000000000

00000000

◆◆ 000000000000000000000000/00000000 Omnitouch 000000000000000000000000

000000

OWL 00000000000000

- 000 NetAct
- 00 U2000 / U2020
- 00 NetNumen / ZXPOS

- ENM

ENM

ENM Omnitouch

ENM

ENM

ENM

ENM NDMO

ENM GIS Google Earth Omnitouch

ENM

ENM NDMO

- ENM
- ENM
- ENM
- ENM

ENM Omnitouch

ENM

ENM

ENM

ENM

- ENM
- ENM
- ENM

- 2FA 2FA

2FA / 2FA

OmniTouch 2FA

- 2FA
- / /
-

1. 2FA
2. / /
3. / /
4. / /

API

API

API

- / -
- -
- -
- -

API

API

CBC

□□□□

PUT /crm/cbc/

□□□□□□

GET /crm/cbc/

□□□□

PATCH /crm/cbc/{cbc_message_id}

□□□□

DELETE /crm/cbc/{cbc_message_id}

□□□□API□□□□□Swagger□□/crm/docs/□

□□□□

□□□□

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** □

□□□□□□□□□□□□□□□ □□□□

□□□ • 89445380000000001234 (□□) - □□ ID: 456

□□□□□□□ □□□□ □□□□□□□□□□

□□ **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

API 例

```
GET /utilities/search_everything?  
search=Smith&page=1&per_page=10&search_closed_records=false  
Authorization: Bearer <token>
```

パラメータ

- search - 検索条件
- page - ページ番号 (デフォルトは1)
- per_page - ページサイズ (デフォルトは10)
- search_closed_records - 閉じたレコードを返すかどうか (デフォルトはfalse)

レスポンス

```
{
  "data": [
    {
      "id": 123,
      "name": "John Smith",
      "customer_id": 123,
      "type": "customer"
    },
    {
      "id": 456,
      "name": "John Smith",
      "customer_id": 123,
      "type": "contact"
    },
    {
      "id": 789,
      "name": "Mobile - +44 7700 900123",
      "customer_id": 123,
      "type": "service"
    }
  ],
  "pagination": {
    "current_page": 1,
    "per_page": 10,
    "total_pages": 5,
    "total_items": 47
  }
}
```

□□□□□□□□□□

□□□□□□□□□□ SQL UNION□

```

-- Customers
SELECT customer_id AS id,
       customer_name AS name,
       customer_id,
       'customer' AS type
FROM customer
WHERE customer_name LIKE '%Smith%'
      AND customer_status = 'Open'

UNION ALL

-- Contacts
SELECT contact_id AS id,
       CONCAT(contact_firstname, ' ', contact_lastname) AS name,
       customer_id,
       'contact' AS type
FROM customer_contact
WHERE (contact_firstname LIKE '%Smith%' OR
      contact_lastname LIKE '%Smith%' OR
      contact_email LIKE '%Smith%' OR
      contact_phone LIKE '%Smith%')

UNION ALL

-- Sites
SELECT site_id AS id,
       site_name AS name,
       customer_id,
       'site' AS type
FROM customer_site
WHERE site_name LIKE '%Smith%'

UNION ALL

-- Inventory
SELECT inventory_id AS id,
       itemtext1 AS name,
       customer_id,
       'inventory' AS type
FROM inventory
WHERE itemtext1 LIKE '%Smith%' OR
      itemtext2 LIKE '%Smith%'

```

UNION ALL

```
-- Services
SELECT service_id AS id,
       service_name AS name,
       customer_id,
       'service' AS type
FROM customer_service
WHERE service_name LIKE '%Smith%' OR
       service_uuid LIKE '%Smith%'
```

□□□□□□□□□□

□□□□

□□□□

- □□□□□□□□ LIKE □□□ %term% □
- □□□□□□□□
- □□□□□□ > 100k □□□□□□□□□□□□□□
- □□□□□□□ 10 □□□□□□

□□□□

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

□□□□

□□□□□□□□□□

- □□□ □□□□□□□□□□□□□□
- □□□
 - □□□□□ "+44 7700 900123" □□□ "07700900123" □□□□
 - □□□□□□□ "john.smith@example.com" □□□ "john" □□□□

- 000000 00000000000000000000

0000

- 000 000000000000000000
- 00000
 - 0000000000
 - 00000000000000000000
 - 00000000000000

0000000000

- 000 0000/0000000000
- 00000 0000 ID 00000000000000000000000000000000

00000000

- 000 000“00000000”000
- 00000 0000000000

00000

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

Objectives

OmniCRM > Self-Care Portal
<self_care_portal>

- SIM - SIM
- WiFi - WiFi dongles
- -

SIM Card Provisioning <concepts_sim_provisioning>
SIM

Stripe

Stripe

Stripe

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

CRM

URL CRM

CRM

- SMS
- CRM

- 000000000000
- 000000

0000000000 IP 000 IMSI 000000000000

0000

000000 **4** 0000

1 00000000

00000000000000000000

000

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

00000

00000

- 000000000000 `REACT_APP_TOPUP_PRICE_PER_DAY` 00
- 000000 \$10 00
- 0000 `REACT_APP_CURRENCY_CODE` 00

2 00000000

00000000000000

- 0
- 0
- 000000

00000000

- 信用卡
- 銀行存款
- 支票
- 電子支付

3 信用卡

Stripe Elements, PayPal SDK

信用卡

- Visa, Mastercard, Amex
- 銀行
- Apple Pay, Google Pay, PayPal

安全

- PCI 認證
- OmniCRM
- 3D Secure
- 加密

流程

1. 信用卡
2. 銀行
3. 支票
4. 電子支付
5. 其他

其他

OCS 認證

4 銀行

銀行

2025 年 1 月 17 日

客户 ID: TXN-123456

标题

内容

正文

正文

正文

正文

正文

1. 正文

正文

- 状态: succeeded
- 计算: days × price_per_day
- 正文

2. 正文

- API 请求: POST /oam/topup_dongle
- 参数: service_uuid, IMSI
- 参数: OCS/CGRateS
- 参数: play_topup_hotspot

3. 正文

正文

- **HotspotTopup** 参数 - 正文
 - payment_intent_id

- service_uuid
- imsi
- 手机号码
- topup_amount
- 手机号码/日期/时间
- 数据源 - 数据源
 - 数据“数据源 - 7 天”
 - 数据topup_amount数据源
 - 数据 service_id 和 customer_id
- 数据源 - 数据源
 - 数据源
 - 数据源
 - 数据源Stripe payment_intent_id
- 数据源 - 数据源
 - 数据“数据 [数据源]”
 - 数据topup_amount数据源 - 数据源
 - 数据源数据源

4. 数据源

数据源数据源数据源 playbook play_topup_hotspot数据源

- 数据 days 数据
- 数据 OCS/CGRateS API
- 数据源数据源
- 数据源数据源

数据源 playbook 数据源

```

- name: Top up hotspot service
  hosts: localhost
  tasks:
    - name: Add balance to OCS
      uri:
        url: "{{ ocs_api_url }}/add_balance"
        method: POST
        body:
          imsi: "{{ imsi }}"
          days: "{{ days }}"
          service_uuid: "{{ service_uuid }}"

```

□□□□□

□□□□□□□□□□□□□□□□□□

SMS □□□

□ OCS □□□□□□ □ Action_Balance_Low □ Action_Balance_Out □
 Action_Balance_Expired □□

□□□□□□□

□ OCS/CGRateS □□□□□□□□□□□□□□□□

□□□□□□□

- Action_Balance_Low - □□□□□□□□□□□□□□ 2 □□
- Action_Balance_Out - □□□□□
- Action_Balance_Expired - □□□□□

□□□□□□□□□□□□□□□□□□□□□□

0000

0000

“00000000”

- 000 Stripe 0000000000000000
- 000
 - 00 REACT_APP_STRIPE_PUBLISHABLE_KEY 000000
 - 00 Stripe 000000000000
 - 000000000000 JavaScript 00

“000000000...”

- 000 00000000OCS 000000playbook 0000
- 000
 - 00000000GET /crm/provision/provision_id/<id>
 - 00 OCS/CGRateS API 000000
 - 00 playbook play_topup_hotspot.yaml 000000
 - 00 Ansible 00

“00000000”

- 000 000000000000???
- 000 00000000000000000000000000000000000000

“000000000”

- 000 UI 000000000000
- 000
 - 00 REACT_APP_TOPUP_PRICE_PER_DAY 00000000000000 \$10
 - 000000000000
 - 000000000000

0000

00000000

HotspotTopup

```
SELECT
  hotspot_topup_id,
  service_uuid,
  days,
  topup_amount,
  status,
  payment_intent_id,
  created
FROM hotspot_topup
WHERE status = 'Failed'
ORDER BY created DESC;
```

```
GET /crm/provision/provision_id/<provision_id>
```

Stripe

Stripe <https://dashboard.stripe.com>

- Stripe PCI 1
- OmniCRM
-
-

-
- IP

- IMSI 查詢
- 查詢

查詢

- 查詢 - 查詢
- 查詢 IP 及 IMSI
- 查詢
- 查詢 CRM 查詢

查詢

查詢

1. 查詢 - 查詢
2. 查詢 - 查詢
3. 查詢 **playbook** 查詢 - 查詢 playbook 查詢
4. 查詢 **OCS** 查詢 - 查詢 OCS API 查詢
5. 查詢 - 查詢 `REACT_APP_TOPUP_PRICE_PER_DAY`

查詢

1. 查詢 **URL** - 查詢
2. 查詢 - SMS 查詢
3. 查詢 - 查詢
4. 查詢 - 查詢

查詢

1. 查詢 **Stripe webhooks** - 查詢 webhook 查詢
2. 查詢 - 查詢 `payment_intent_id`
3. 查詢 - 查詢
4. 查詢 - 查詢
5. 查詢 - 查詢 <5 查詢

□□□□

- `payments_process` - □□□□□
- `concepts_provisioning` - □□□□□
- `Payment System Guide <payment_system_guide>` - □□□□□□□□□□
- `payments_transaction` - □□□□
- `payments_invoices` - □□□□

Table of Contents

OmniCRM Ansible

Table of Contents

OmniCRM

1. Introduction - Overview of the project
2. Prerequisites - Required software and dependencies
3. Installation - Step-by-step instructions for setting up the environment
4. Configuration - Details on how to configure the system

Table of Contents

OmniCRM

Table of Contents

Table of Contents

- Introduction
- Prerequisites
- Installation
- Configuration

Table of Contents

Table of Contents

Table of Contents

Table of Contents

- `wholesale_setup_cost` - 0.00
- `retail_setup_cost` - 0.00

API

```
{
  "retail_cost": 15.00,
  "wholesale_cost": 5.00,
  "retail_setup_cost": 0.00,
  "wholesale_setup_cost": 1.00
}
```

API 1

API 1: Prepaid Mobile 20GB

API 1 SIM Card

API 1: Prepaid Mobile 20GB SIM Card

API 1: Prepaid Mobile 20GB

API 1: UI 1000 → 10000

API 2: Prepaid Mobile 20GB

```
{
  "product_name": "Prepaid Mobile 20GB",
  "product_slug": "prepaid-mobile-20gb",
  "category": "standalone",
  "service_type": "mobile",
  "enabled": true,
  "icon": "fa-solid fa-sim-card",
  "comment": "Prepaid Mobile 20GB SIM Card"
}
```

API 2

- `product_name` - 产品名称
- `product_slug` - 产品 API 接口 URL 标识符
- `category` - "standalone" 独立产品/服务
- `service_type` - 服务类型
- `enabled` - 是否启用
- `icon` - UI 图标 FontAwesome 图标
- `comment` - 备注

第 3 步

```
{
  "retail_cost": 15.00,
  "wholesale_cost": 5.00,
  "retail_setup_cost": 0.00,
  "wholesale_setup_cost": 1.00,
  "contract_days": 30
}
```

解释

- 零售成本 £15.00
- 批发成本 £5.00
- 零售设置成本 £10.00 200% 折扣 67% 折扣
- 批发设置成本 -£1.00
- 合同天数 30 天

第 4 步

```
{
  "residential": true,
  "business": false,
  "customer_can_purchase": true,
  "available_from": "2025-01-01T00:00:00Z",
  "available_until": null
}
```

- 住宅产品

- 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"]
}
```

設定する

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

設定する

inventory_items_list Python `<administration_inventory>`

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

#
inventory_items_list: ["Modem Router", "Static IP Address"]

#
inventory_items_list: []

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

1.

inventory_items_list

2.

3.

- -
- item_state "New" "In Stock" "Assigned" "Damaged"
- service_id customer_id NULL
- - /

“SIM ”

- ID
- `itemtext1` - SIM ICCID
- `item_location`

4.

- “”
-
-

5. Ansible

“” ID Ansible

```
#  
# - SIM inventory_id: 5001  
# - inventory_id: 5002  
  
# Ansible  
{  
  "product_id": 42,  
  "customer_id": 123,  
  "SIM Card": 5001,           # ID  
  "Mobile Number": 5002,     # ID  
  "access_token": "eyJ..."  
}
```

`hostvars[inventory_hostname]['SIM Card']` ID

6.

Ansible ID

```

- name: SIM
  uri:
    url: "{{ crm_config.crm.base_url
  }}/crm/inventory/inventory_id/{{ hostvars[inventory_hostname]['SIM
  Card'] }}"
  method: GET
  headers:
    Authorization: "Bearer {{ access_token }}"
  register: api_response_sim

- name: ICCID IMSI
  set_fact:
    iccid: "{{ api_response_sim.json.itemtext1 }}"
    imsi: "{{ api_response_sim.json.itemtext2 }}"

```

SIM ICCID IMSI HSS

7. “”

```

- name: SIM
  uri:
    url: "{{ crm_config.crm.base_url
  }}/crm/inventory/inventory_id/{{ hostvars[inventory_hostname]['SIM
  Card'] }}"
  method: PATCH
  body:
    {
      "service_id": "{{
  service_creation_response.json.service_id }}",
      "customer_id": "{{ customer_id }}",
      "item_state": "Assigned"
    }

```

- “”
-
-


```
inventory_items_list: ["ONT Device", "GPON Port", "IPv4 Address",  
'Fiber Drop Cable']"
```

- ONT 設備名稱
- GPON 設備名稱 OLT 設備名稱
- IPv4 設備名稱 IP
- 設備名稱

設備名稱

```
inventory_items_list: ["Rental Modem"]"
```

- 設備名稱
- 設備名稱

設備名稱

1. 設備名稱

設備名稱

- 設備 SIM 設備名稱
- 設備 IP 設備名稱
- 設備名稱

設備名稱

2. 設備名稱

設備名稱

- 設備 SIM 設備名稱
- 設備名稱
- 設備名稱
- 設備名稱

3. 設備名稱

□□□□□□

- □ SIM □□□□□□□□□□
- □□□□□□□□□□
- □□ CPE □□□□□□□□□□
- □□ IP □□□□□□

4. □□□□

□□□□□□□□□□□□□□

- □□□□ SIM □□□□
- □□□□□□
- □□□□□□□□□□
- □□□ COGS□□□□□□

5. □□□□




□□□□□□□□□□□□□□

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

□□□□□□□□□□□□

□□□ “□□□□□□”□□□

□□□

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

□□□□□□

1. □□□□□□□□□□□□□□ → □□ → □□


```
{
  "features_list": "20GB 存储空间可扩容至30 GB",
  "terms": "30 天无理由退换货"
}
```

存储空间

30天 - 无理由

退换货

- 存储空间 - 存储空间可扩容至30 GB
 - “存储空间 20GB”“存储空间 50GB” - 存储空间
 - 存储空间
- 30天 - 无理由
 - “30天”“30天”
 - “30天”
 - 30天
- 退换货 - 无理由
 - 退换货“无理由”→退换货❓❓❓
 - 退换货
 - 退换货
- 30天 - 无理由
 - 退换货“无理由”
 - 退换货
 - 退换货
- SEO** 存储空间 - 存储空间
 - 存储空间“存储空间”→存储空间
 - 存储空间

存储空间 - 无理由

退换货

- 存储空间 - 存储空间
 - “30天 30天” - 31天
 - “30天” - 30天

- 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. 注册与登录

注册与登录流程

2. 忘记密码

忘记密码重置流程

- 忘记密码
- 重置密码
- 验证邮箱
- 设置新密码
- 完成重置

3. 套餐

套餐选择与配置

套餐 • 20GB 流量 • 无限通话 • 无限短信 • 30 天试用

每月 £15.00 / 首次 £0.00

[按钮] [按钮]

4. 条款与隐私

服务条款与隐私政策

服务条款

[terms](#) 服务条款

服务 **UI** 界面

1. 注册与登录

注册与登录流程

2. 忘记密码

忘记密码重置流程

[]

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\"
  }
```

```
}"  
}
```

□□□□□□□□

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

□□ **1**□□□□□□□□ **5GB** □□□□□

□□□□□□□□□□□□□□□□□□□□

```
{
  "product_name": "5GB 套餐",
  "product_slug": "5gb-data-boost",
  "category": "addon",
  "service_type": "mobile",
  "enabled": true,
  "icon": "fa-solid fa-plus",
  "comment": "套餐 5GB 套餐",

  "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 套餐 7 天",
  "terms": "套餐 7 天套餐",

  "provisioning_play": "play_topup_charge_then_action",
  "provisioning_json_vars": "{
    \"data_gb\": 5,
    \"days\": 7
  }"
}
```

套餐 2 套餐

套餐套餐套餐套餐套餐套餐套餐

```

{
  "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
  }"
}

```

모뎀 렌탈

- `category: "addon"` - 모뎀 렌탈
- `contract_days: 0` - 모뎀 렌탈 - 모뎀
- `inventory_items_list: "[]"` - 모뎀 렌탈 - 모뎀

- `auto_renew: "false"` 或 `"true"` 选项 - 是否
- `relies_on_list: ""` - 依赖项列表 `service_type` 选项

配置项

配置项 `inventory_items_list`

1. 名称 - 配置项名称
2. 描述 - 配置项描述
3. 类型 - 配置项类型
4. 单位 - 配置项单位
5. 数量 - 配置项数量
6. 位置 - 配置项位置

配置项

配置项“WiFi 6 配置项”配置项

1. 名称 - 配置项“配置项”
2. 描述 - 配置项描述
3. 价格 - 配置项 £10.00
4. 配置项 - 配置项
 - `service_id` 配置项
 - `customer_id` 配置项
 - `item_state` "Assigned"
5. 数量 - 配置项数量
6. 位置 - 配置项位置 ONT
7. 价格 - 配置项 £10 配置项

配置项

配置项

1. 名称 - 配置项“配置项”
2. 配置项
 - 配置项
 - 配置项

- 14 日以内

3. 状態

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

4. 価格

- 14 日以内 £150 以内
- `item_state = "Lost"`
- £450+ 以内

状態

状態

- 状態
- 状態
- 状態 `contract_days` 状態

2 状態

状態“状態 20GB”状態OmniCRM 状態 Ansible 状態

状態

状態 → 状態 → 状態 ↓ ↓ 状態 ← 状態 ← 状態 Ansible 状態 ↓ ↓ 状態 → OCS 状態
 状態 → 状態 → 状態

状態

1. 状態

状態

- 状態“状態”
- 状態“状態 20GB”
- 状態

2. SIM

Placeholder text

- **SIM** - Placeholder SIM
 - Placeholder "SIM-00123 - ICCID: 8944..."
- Placeholder - Placeholder
 - Placeholder "+44 7700 900123"

Placeholder text

3. SIM

Placeholder text

- Placeholder £0.00 Placeholder
- Placeholder £15.00
- Placeholder £15.00 Placeholder
- Placeholder 30 Placeholder

Placeholder text

Placeholder 30 Placeholder

4. SIM

Placeholder "Placeholder" API

- Placeholder "Placeholder" status=1 Provision Placeholder
- Placeholder + Placeholder + Placeholder
- Placeholder Ansible Placeholder
- Placeholder provision_id Placeholder UI Placeholder

5. SIM

Placeholder text

Placeholder

```
{
  "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 □□□□□□□□□□□□□□□□

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

- SIM卡数据

第3步

```
- name: SIM卡
  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
}}"
```

第4步

第4步

```
- name: 获取 CRM API 返回的 SIM 卡信息
  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: 提取 SIM 卡的 IMSI
  set_fact:
    imsi: "{{ api_response_sim.json.itemtext2 }}"
    iccid: "{{ api_response_sim.json.itemtext1 }}"
```

输出

- SIM 卡 ID 5001
- SIM 卡信息
 - itemtext1 = ICCID[SIM 卡号]
 - itemtext2 = IMSI[IMSI 号]
- 输出结果

输出

- IMSI 0000 HSS0000000000000000
- ICCID 0000000000000000
- 000000MSISDN000000000000

00 5000000 UUID

```

- name: 00 UUID 00
  set_fact:
    uuid: "{{ 99999999 | random | to_uuid }}"

- name: 0000 UUID
  set_fact:
    service_uuid: "Local_Mobile_SIM_{{ uuid[0:8] }}"

```

00000000

- 0000 UUID
- 0000 Local_Mobile_SIM_a3f2c1d8 0 service_uuid

0000

- 00 UUID 0 OCS/CGRateS 00000000
- 0000000000
- 0000000000000000

00 600 OCS 000000

```

- name: OCS
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    headers:
      Content-Type: "application/json"
    body:
      {
        "method": "ApierV2.SetAccount",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "ActionPlanIds": [],
          "ExtraOptions": {
            "AllowNegative": false,
            "Disabled": false
          },
          "ReloadScheduler": true
        }]
      }
  register: ocs_create_response

```

Annotations

- CGRateS JSON-RPC API
- service_uuid
-
-

Variables

- OCS
-
-

7

```

- name: 20GB
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "DATA_20GB_Monthly",
            "Value": 21474836480,
            "ExpiryTime": "+720h",
            "Weight": 10,
            "DestinationIDs": "*any"
          }
        }]
      }

```

20GB

- 20GB
- 21474836480 $20 * 1024 * 1024 * 1024$
- 720 30
- 10

8

```
- name: 000000
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Account": "{{ service_uuid }}",
          "BalanceType": "*voice",
          "Balance": {
            "ID": "VOICE_Unlimited",
            "Value": 999999999,
            "ExpiryTime": "+720h"
          }
        }]
      }
}
```

- 999,999,999 0000000000000000
- 30 0000

00 900 CRM 00000000

```

- name: API
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/service/"
    method: PUT
    body_format: json
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      {
        "customer_id": "{{ customer_id }}",
        "product_id": "{{ product_id }}",
        "service_name": "Mobile - {{ phone_number }}",
        "service_type": "mobile",
        "service_uuid": "{{ service_uuid }}",
        "service_status": "Active",
        "service_provisioned_date": "{{ provision_datetime }}",
        "retail_cost": "{{ monthly_cost }}",
        "wholesale_cost": "{{
api_response_product.json.wholesale_cost }}",
        "icon": "fa-solid fa-sim-card"
      }
  register: service_creation_response

```

注意

- 服务UUID
- 服务UUID OCS
- 服务名称
- 服务类型“mobile”
- 服务ID

10

```

- name: SIM
  uri:
    url: "{{ crm_config.crm.base_url
  }}/crm/inventory/inventory_id/{{ hostvars[inventory_hostname]['SIM
  Card'] }}"
  method: PATCH
  body_format: json
  headers:
    Authorization: "Bearer {{ access_token }}"
  body:
    {
      "service_id": "{{
  service_creation_response.json.service_id }}",
      "customer_id": "{{ customer_id }}",
      "item_state": "Assigned"
    }

```

11

- SIM
- service_id SIM
- "SIM"
-

11

- SIM
-
-

11

```

- name: 0000000000
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/transaction/"
    method: PUT
    body_format: json
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      {
        "customer_id": "{{ customer_id }}",
        "service_id": "{{
service_creation_response.json.service_id }}",
        "title": "{{ package_name }} - 00",
        "description": "0000",
        "retail_cost": "{{ setup_cost }}",
        "wholesale_cost": "{{
api_response_product.json.wholesale_setup_cost }}"
      }

```

00000000

- 00 £0.00 0000000000000000
- 00 £1.00 000000
- 0000000000000000

00 **12**0000000000000000

```

rescue:
  - name: 删除OCS 套餐
    uri:
      url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
      method: POST
      body:
        {
          "method": "ApierV2.RemoveAccount",
          "params": [{
            "Account": "{{ service_uuid }}"
          }]
        }
  - name: 失败
    fail:
      msg: "删除OCS 套餐失败"

```

删除套餐

- 删除套餐
- 删除OCS 套餐
- 删除“套餐”
- 删除套餐

删除

- OCS 套餐
- 套餐
- 套餐

删除套餐

删除套餐

1. OCS 套餐CGRateS

- ID Local_Mobile_SIM_a3f2c1d8
- 套餐
 - 20GB 套餐 30 套餐

- 999M 30
- 999M 30

2. CRM

- ID 1234
- John Doe [customer_id: 123]
- 20GB [product_id: 42]
- "Mobile - +44 7700 900123"
- UUID [Local_Mobile_SIM_a3f2c1d8]
-
- £15.00 £5.00
- £10.00/

3.

- SIM 5001 1234 123
- 5002 1234 123

4.

-
-

5.

-
- 20GB
-
-
-

3

API 调用

调用 API 获取 20GB 套餐 18GB 套餐和 5GB 套餐

1. 获取套餐

- 调用 "Mobile - +44 7700 900123" API
- 返回 20GB 套餐 18GB 套餐 90%
- 返回 "套餐名称" 和 "价格"

2. 创建套餐

创建套餐

- `category = "addon"`
- `service_type = "mobile"` 套餐名称
- `residential = true` 是否住宅套餐
- `enabled = true`

返回 "5GB 套餐 - £5.00"

3. 更新套餐

- 调用 "5GB 套餐"
- 返回 £5.00
- 返回套餐名称

4. 套餐配置

调用 `play_topup_charge_then_action.yaml` API

```
{
  "product_id": 43, # 5GB 套餐
  "customer_id": 123,
  "service_id": 1234, # 套餐
  "access_token": "eyJ...",
  "data_gb": 5, # 套餐 provisioning_json_vars
  "days": 7 # 套餐 provisioning_json_vars
}
```

□□□□□□□□□□

- □□ `service_id` □□□□□□□□
- □□□□□
- □□□□□□□□□□□□

□□□□□□□□□□□□

□□ **1** □□□□□□□□□□

```
- name: □ CRM API □□□□□□
  uri:
    url: "http://localhost:5000/crm/service/service_id/{{
service_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    register: api_response_service

- name: □□□□□□
  set_fact:
    service_uuid: "{{ api_response_service.json.service_uuid }}"
    customer_id: "{{ api_response_service.json.customer_id }}"
```

□□□□

- □□ `service_uuid` □□□□□ OCS □□□□□□
- □□□□□□□□□□□□□□□□
- □□□□□□□□

□□ **2** □□□□□□□

```

- name: 创建支付方法
  uri:
    url: "http://localhost:5000/api/payments/methods/default?customer_id={{ customer_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: api_response_payment_method

- name: 设置支付方法 ID
  set_fact:
    payment_method_id: "{{ api_response_payment_method.json.data.payment_method_id }}"

- name: 创建支付
  uri:
    url: "http://localhost:5000/api/payments/charge"
    method: POST
    body_format: json
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      {
        "customer_id": "{{ customer_id | int }}",
        "amount": 5.00,
        "currency": "USD",
        "payment_method_id": "{{ payment_method_id }}",
        "metadata": {
          "description": "5GB 存储空间",
          "service_id": "{{ service_id | int }}",
          "product_id": "{{ product_id | int }}",
          "invoice": true
        }
      }
  register: charge_response

- name: 断言支付成功
  assert:
    that:
      - charge_response.json.success == true

```

创建支付

- Stripe 5GB
- £5.00
- £1.50
- Stripe
- Stripe
- Stripe

Stripe

- Stripe
- Stripe
- Stripe

3 OCS

```

- name: Stripe 5GB
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "DATA_5GB_Boost_{{ uuid }}",
            "Value": 5368709120,
            "ExpiryTime": "+168h",
            "Weight": 20
          }
        }]
      }

```

Stripe

- Stripe 5GB 5368709120
- 168 7

- 20년 - 20년

20년

- 2GB 25
- 5GB 7
- 7GB
- 20년

4년

```
- name: 20년
  uri:
    url: "http://localhost:5000/crm/transaction/"
    method: PUT
    body_format: json
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      {
        "customer_id": "{{ customer_id }}",
        "service_id": "{{ service_id }}",
        "title": "5GB",
        "description": "5GB 7",
        "retail_cost": 5.00,
        "wholesale_cost": 1.50
      }
}
```

20년

- £5.00
- £1.50
- 20년

20년

1. 20년
2. 20년
3. 20년 OCS

4. CRM 10000

5. 1000000000000000 7GB

10000

- 1000000£15 1000£5
- 1000000£5 1000£1.50

100000000000000

10000000000000000000000000000000

10000

```
{
  "product_name": "10GB 10000",
  "category": "addon",
  "retail_cost": 10.00,
  "contract_days": 30,
  "auto_renew": "true",
  "provisioning_play": "play_topup_charge_then_action"
}
```

100000000

```

- 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...

本章学习目标

通过本章学习，应达到以下目标：

1. 了解...“...”
2. 掌握...“...”
3. 熟悉...“...”
4. 能够...“...”

本章重点

1. ...“...”

- ...“...”
- ...“...”
- ...“...”
- ...“...”

2. ...“...”

本章主要介绍...

- ...“...”
- 7-30 ...“...”
- ...“...”
- ...“...”

3. ...“...”

本章主要介绍...

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

[[service.deprovisioning_play]]

4. Ansible []

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

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

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

  - name: [] [] [] [] [] [] [] [] [] [] [] [] [] []
    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

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

状態

状態

- `New` - 新
- `In Stock` - 在庫
- `Assigned` - 指定
- `Decommissioned` - 廃止
- `Damaged` - 破損

状態

状態

- SIM 状態“状態”
- 状態30 状態
- 状態“状態”

状態

状態

- 状態
- 状態
- 状態
- 状態

Mailjet & OmniCRM

OmniCRM & Mailjet

Mailjet

-
-
- 10
-
-

Mailjet OmniCRM-API/crm_config.yaml mailjet

```
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

API Parameters

OmniCRM API **10** Parameters for Mailjet Integration

- **from_email** - Sender email address
- **from_name** - Sender name
- **template_id** - Mailjet Template ID
- **subject** - Email subject

Example

Request

Response

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

Variables

- {{ var:customer_name }} - Customer name
- {{ var:email }} - Customer email
- {{ var:company_name }} - Company name

Links


- {{ var:login_url }} - Login URL
- {{ var:support_url }} - Support URL

Footer

□□□□□□□□□□□□□□

```
api_crmCommunicationCustomerInvoice:  
  from_email: "billing@yourcompany.com"  
  from_name: "Your Company Billing"  
  template_id: 6759851  
  subject: "□□□□ - "
```

□□□□□

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

□□□□□□□□

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

□□□□□□

□□ PDF □□□□□□□□□□□□□□

□□□□□□□

□□□□□□□□□□□□□□

```
api_crmCommunicationCustomerInvoiceReminder:  
  from_email: "billing@yourcompany.com"  
  from_name: "Your Company Billing"  
  template_id: 6759852  
  subject: "□□□□ - □□□□"
```

□□□□

- □□□□ X □□□□□□
- □□□□□□□□
- □□□□□□□□□□□□□□

□□□□□□□

- {{ var:customer_name }}
- {{ var:invoice_number }}
- {{ var:due_date }}
- {{ var:days_overdue }}
- {{ var:total_amount }}
- {{ var:pay_url }}

□□□□□□□□

□□□□□□□□□□□□□□

```
api_crmCommunicationUserWelcome:  
  from_email: "admin@yourcompany.com"  
  from_name: "Your Company Admin"  
  template_id: 5977510  
  subject: "□□□□□□"
```

□□□□□

- □□□□□□□□□□
- □□□□□□□□□□"□□□□□□"□□

□□□□□□□

- {{ var:user_name }} - □□□□□□□□
- {{ var:email }} - □□□□□□□□□□
- {{ var:role }} - □□□□□□
- {{ var:login_url }} - □□□□□□□□□□
- {{ var:temp_password }} - □□□□□□□□□□

- `{{ var:support_email }}` - IT サポート

メール本文

メール本文本文本文

```
api_crmCommunicationUserPasswordReset:  
  from_email: "noreply@yourcompany.com"  
  from_name: "Your Company Security"  
  template_id: 5977511  
  subject: "パスワードリセット"
```

メール本文

- パスワードリセットメール本文
- パスワードリセットURL

メール本文

- `{{ var:user_name }}`
- `{{ var:reset_url }}` - パスワードリセットURL (1 回有効)
- `{{ var:expiry_time }}` - 有効期限

メール本文

パスワードリセットURL 1 回有効

メール本文

パスワードリセットURL

```
api_crmCommunicationUserPasswordResetSuccess:  
  from_email: "noreply@yourcompany.com"  
  from_name: "Your Company Security"  
  template_id: 5977512  
  subject: "パスワードリセット完了"
```

メール本文

- 0000000000
- 0000000000

00000000

- {{ var:user_name }}
- {{ var:change_date }} - 00000000/00
- {{ var:ip_address }} - 000 IP 00000000
- {{ var:support_email }} - 000000000000

00000000

000000000000000000

```
api_crmCommunicationUserPasswordChange:
  from_email: "noreply@yourcompany.com"
  from_name: "Your Company Security"
  template_id: 5977513
  subject: "00000000"
```

000000

- 00000000/00000000
- 0000000000

00000000

- {{ var:user_name }}
- {{ var:change_date }}
- {{ var:changed_by }} - "00"00000000
- {{ var:support_email }}

00000000

000000000000000000

```
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: "████████████████"
```

██████

- ██████████
- ████████████████████
- ████████████████████

██████████

- `{{ var:customer_name }}`
- `{{ var:service_name }}` - ████████████████████

- `{{ var:expiry_date }}`
- `{{ var:balance_type }}` - “”“”“”“”
- `{{ var:renewal_url }}` - /

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

- 20%
- OCS
-

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

Mailjet

Mailjet

1 Mailjet

1. Mailjet

2. 200 OK → 204 No Content
3. 200 OK
4. 200 OK (with headers) → 204 No Content

2. Mailjet

Mailjet HTML

- -
- - `{{ var:customer_name }}` `{{ var:user_name }}`
- -
- -
- -
- -

```

<!DOCTYPE html>
<html>
<head>
  <style>
    body { font-family: Arial, sans-serif; }
    .button { background-color: #4CAF50; color: white;
padding: 14px 28px; }
  </style>
</head>
<body>
  ![Logo](https://yourcompany.com/logo.png)

  <h2>○○○○○○</h2>

  <p>○○ {{ var:user_name }}</p>

  <p>○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○</p>

  <a href="{{ var:reset_url }}" class="button">○○○○</a>

  <p>○○○○ {{ var:expiry_time }} ○○○</p>

  <p>○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○</p>

  <hr>
  <p style="font-size: 12px; color: #666;">
    ○○○ | support@yourcompany.com<br/>
    123 ○○○○○○○○○
  </p>
</body>
</html>

```

3 ○○○○○ ID

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

4. Mailjet Configuration

1. Mailjet API Key and Secret
2. Mailjet Account
3. Mailjet Template
4. Mailjet User

5. OmniCRM Configuration

Configuration file: `crm_config.yaml`

```
mailjet:  
  api_key: your_api_key  
  api_secret: your_secret  
  
  api_crmCommunicationUserPasswordReset:  
    from_email: "noreply@yourcompany.com"  
    from_name: "Your Company Security"  
    template_id: 5977511  
    subject: "Security Alert"
```

Restart OmniCRM API

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

6. Troubleshooting

OmniCRM configuration issues with Mailjet

Check the following:

- Mailjet API Key
- Mailjet Account
- Mailjet Template
- Mailjet User

- 環境変数

環境変数

環境変数の設定

- 環境変数
- 設定/取得
- 環境変数の名前

環境変数

環境変数の設定

- 環境変数 API キー Mailjet 環境変数 ID
- 環境変数
 - 環境変数 `crm_config.yaml` 環境変数 `api_key` 環境変数 `api_secret`
 - 環境変数 Mailjet 環境変数
 - 環境変数 ID 環境変数 Mailjet
 - 環境変数 API 環境変数

環境変数の取得

- 環境変数 環境変数 OmniCRM 環境変数
- 環境変数
 - 環境変数 環境変数
 - 環境変数 `{{ var:variable_name }}` 環境変数
 - 環境変数 OmniCRM 環境変数 API 環境変数
 - 環境変数 Mailjet 環境変数

環境 PDF 環境

- 環境 PDF 環境
- 環境
 - 環境
 - 環境

- PDF 15MB Mailjet
- PDF

Mailjet

- API
- - Mailjet API 200
 - OmniCRM-API
 -
 -

- `administration_configuration` - Mailjet
- `payments_invoices` -
- `authentication_flows` -
- `customer_care` -
- - Mailjet API

- Mailjet <<https://dev.mailjet.com/>>
- Mailjet API <<https://dev.mailjet.com/email/reference/>>



OmniCRM 50

API

OmniCRM API を Prometheus で監視するための Prometheus scrape config を提供します。
/crm/metrics を Prometheus が scrape できるように設定します。

API

API

- API - 監視対象の API/サービス
- API - 監視対象の API
- API - OCS/CGRateS, Stripe, Mailjet API
- API - 監視対象の API
- **HTTP** API - API を監視するための Prometheus scrape config

API

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

API: Prometheus scrape config

API: Prometheus scrape config

API

1. API

Ansible を使用して Prometheus scrape config を自動的に生成するための [Provisioning System](#) `Ansible Playbooks`

omnicrm_provision_jobs_total

API: API

{}:

- `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 設定項目に関する [System Architecture](#)

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 の設定項目

omnicrm_db_pool_size

例: 10

例: 設定項目

omnicrm_db_pool_overflow

例: 10

例: 設定項目

omnicrm_db_pool_connections_checked_out

例: 10

例: 設定項目

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

omnicrm_ocs_api_errors_total

Table: Table

Table:

- `method` - OCS API Method
- `error_type` - Error Type: `timeout`, `connection_error`, `json_error`, `http_error`, `...`

Table: OCS/CGRateS API OCS API Error Summary Table

omnicrm_ocs_balance_queries_total

Table: Table

Table:

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

Table: OCS 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

- `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 API의 HTTP 요청 총 횟수를 나타냅니다.

`flask_http_request_exceptions_total`

타입: `counter`

메트릭:

- `method` - HTTP 메서드
- `endpoint` - Flask 엔드포인트

이 메트릭은 HTTP 요청 중 예외가 발생한 횟수를 나타냅니다.

8. API `omnicrm_api_errors_total`

API `omnicrm_api_errors_total` returns the total number of errors.

`omnicrm_api_errors_total`

Response: `int`

Fields:

- `endpoint` - API endpoint
- `error_type` - Error type
- `status_code` - HTTP status code

Response: `int`

`omnicrm_api_auth_failures_total`

Response: `int`

Fields:

- `auth_method` - Authentication method: `jwt`, `api_key`, `ip_whitelist`
- `reason` - Reason for failure

Response: `int`

9. `omnicrm_api`

`omnicrm_api`

Response: `int`

Fields:

- `version` - API version
- `environment` - Environment: `production`, `staging`, `development`

Response: OmniCRM API version and environment information

□□□□

update_db_pool_metrics(engine)

□ 30 □□□□□□□□□□□□□□□□□□

Prometheus □□

□□□□

□ OmniCRM □□□□ `prometheus.yml` □

```
scrape_configs:  
  - job_name: 'omnicrm-api'  
    scrape_interval: 15s  
    scrape_timeout: 10s  
    static_configs:  
      - targets: ['omnicrm-api:5000']  
    metrics_path: '/crm/metrics'
```

□□□□

□□□□□□

```

- alert: HighProvisionFailureRate
  expr: |
    (
      rate(omnicrm_provision_jobs_total{status="failed"}[5m]) /
      rate(omnicrm_provision_jobs_total[5m])
    ) > 0.1
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "High Provision Failure Rate"
    description: "{{ $value | humanizePercentage }} High Provision Failure Rate"

```

High Provision Failure Rate

```

- alert: DatabasePoolExhausted
  expr: omnicrm_db_pool_overflow > 0
  for: 2m
  labels:
    severity: critical
  annotations:
    summary: "Database Pool Exhausted"
    description: "Database Pool Exhausted"

```

Database Pool Exhausted

```

- alert: SlowDatabaseQueries
  expr: |
    histogram_quantile(0.99,
      rate(omnicrm_db_query_duration_seconds_bucket[5m])
    ) > 1.0
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "Slow Database Queries"
    description: "99th Percentile of Database Query Duration is {{ $value }}s"

```

OCS API

```

- alert: OCSAPIDown
  expr: |
    (
      rate(omnicrm_ocs_api_calls_total{status="failed"}[5m]) /
      rate(omnicrm_ocs_api_calls_total[5m])
    ) > 0.5
  for: 2m
  labels:
    severity: critical
  annotations:
    summary: "OCS API 🚫🚫🚫🚫"
    description: "{{ $value | humanizePercentage }} 🚫 OCS API 🚫🚫
🚫"

```

Stripe 🚫🚫🚫

```

- alert: StripePaymentFailures
  expr: rate(omnicrm_stripe_payment_failures_total[5m]) > 5
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "Stripe 🚫🚫🚫🚫🚫"
    description: "{{ $value }}" 🚫🚫🚫🚫

```

🚫🚫🚫🚫

🚫🚫🚫🚫

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

2. Prometheus - KPI

- Stripe KPI
- KPI
- KPI

3. Prometheus - KPI

- API KPI
- KPI
- API KPI

4. Prometheus - KPI

- KPI
- KPI
- API KPI

API

Prometheus KPI:

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

API Thanos Cortex VictoriaMetrics KPI

API

API

API /metrics KPI nginx KPI

API:

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

Prometheus 配置:

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

配置

配置

- 配置
- Stripe 配置
- OCS 配置

配置

Prometheus 配置

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

>10,000 配置

配置

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](#)



Payment System Guide

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

Payment System Guide

- [Stripe](#) [PayPal](#)
 - [Stripe](#) [PayPal](#) [Stripe](#) [PayPal](#) [Stripe](#) [PayPal](#)
 - [Stripe](#)
 - [PayPal](#)
- **PayPal** [PayPal](#)

OmniCRM

OmniCRM

- [stripe](#) [paypal](#)
- [Stripe](#) [PayPal](#)

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

00000000

000000

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

00000

00000000000000

0000000000000000

0000000000000000 60 000000000000

00000000

0000000000000000

□□□□

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

□□□□□□□□

□□□□□□□□□□□□ : □□□□□□□□

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

□□□□□□□□

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

□□□□□□□□□□□□□□□□

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

□□□□□□

- 信用卡 - 信用卡CVC Stripe 支付
- 信用卡 - 信用卡
- 卡号/卡 - 信用卡

信用卡

- 信用卡 - 信用卡

信用卡

- 信用卡支付集成 iframe Stripe Elements / PayPal SDK
- OmniCRM 信用卡支付集成
- PCI DSS 信用卡支付集成
- 信用卡支付集成

信用卡支付集成

信用卡 "Add Payment Method" 集成

1. 信用卡支付
 - 信用卡支付集成
 - 信用卡支付集成
 - 信用卡 CVC 支付
2. 信用卡
 - 信用卡支付集成 OmniCRM
 - 信用卡支付集成 pm_1A2B3C4D Stripe
 - 信用卡支付 OmniCRM
3. 信用卡支付
 - OmniCRM 信用卡支付集成
 - 信用卡支付集成
 - 信用卡支付 OmniCRM 支付

信用卡支付

信用卡支付

信用卡 Visa 卡号 1234 信用卡支付集成

□□□□□□□□□□□□

□□□□□□

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

□□□□□□□□

□□□□□□□□

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

□□□□□□□□

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

Visa □□ 5678 □□□□□□□□□□□□

□□□□□□□□□□□□

□□□□□□

□□□□□□□□

□□□□□□□□□□□□□□□□

刪除付款方法

請按照以下步驟操作

步驟一

- 進入「我的帳戶」
- 選擇「付款方法」
- 點選 **"Delete"**

步驟二

請確認

刪除付款方法

您將刪除 Visa 卡 1234 有效期至 12/2026

⚠️ 刪除後，該卡將不再用於任何自動扣款。

[刪除] [取消]

刪除 "Delete Payment Method" 後

注意事項

請注意

刪除付款方法後

將發生

- 該卡將不再用於任何自動扣款
- 該卡將不再顯示在您的付款方法列表中
- 如果您需要重新添加該卡，請聯繫您的銀行

XXXXXXXXXX

OmniCRM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXX

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

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

◆◆ XXX

- XXXX - XXXX 30/60/90 XXXXX
- XXXX - XXXXX
- XXXX - XXXXXXXXXXXXXXX

XXX

- XX - XXXXXXXXXXXXXXXXXXXXXXX
- XX - XXXXXXXXXXXXXXXXXXXXXXX Mailjet

📁 📁

📁 📁 Mailjet 📁 📁 📁 📁 📁 📁

- 📁 📁 **60** 📁 - 📁 📁 📁
- 📁 📁 **30** 📁 - 📁 📁 📁
- 📁 📁 **7** 📁 - 📁 📁 📁
- 📁 📁 - 📁 📁 📁 📁 📁

📁 📁

📁 📁 📁 📁 📁 📁

Mailjet 📁 📁 📁 📁

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

📁 📁 `integrations_mailjet` 📁 📁 📁 📁 📁 📁 📁 📁

📁 📁 📁 📁 📁 📁 📁 📁

📁 📁 📁 📁 📁

1. 📁 📁 📁 📁 📁 📁 📁 📁
2. 📁 📁 📁 📁 📁 📁
3. 📁 📁 **Billing → Payment Methods**
4. 📁 📁 **"Add Payment Method"**
5. 📁 📁 📁 📁 📁 📁 📁 📁 📁 📁 📁 📁 📁 📁 📁 📁
6. 📁 📁 📁 📁 📁
7. 📁 📁 📁 / 📁 📁 📁 📁

📁 📁 📁 📁 📁

📁 📁 📁 📁 📁 📁 📁 📁

1. 背景
2. 背景 **Billing → Payment Methods**
3. 背景
4. 背景
5. 背景
6. 背景

背景

背景

- 背景
- 背景
- 背景

背景

背景

背景

1. 背景
 - 背景
 - 背景
 - 背景
 - 背景
2. 背景
 - 背景
 - 背景
 - 背景“背景”背景
3. 背景
 - 背景 - 背景
 - 背景 - 背景
 - 背景 - 背景

背景

□ **1-3** □□□□□□

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

□ **4-7** □□

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

□ **8-14** □□

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

□ **15** □□□□□

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

□□□□□□□□

□□□□□□□□

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

□□□□□□□□

□□□□□□□□□□□□

1. □□□□□□□□□□

2. 商家
3. 商家代辦商
4. 商家代辦商代辦商
5. 商家代辦商代辦商代辦商

商家代辦商

商家

OmniCRM 商家代辦商

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

OmniCRM 商家代辦商

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

OmniCRM 商家代辦商

- 商家代辦商
- CVV/CVC 碼
- 商家代辦商
- PIN 碼

PCI 概要

概要

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

Payment System Guide <payment_system_guide> 概要

概要

1. 概要

概要

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

2. 概要

概要

1. 概要 60 概要
2. 概要
3. Billing → Payment Methods
4. 概要 12/2025
5. "Add Payment Method"

6. 0000 12/2028 0000
7. 00000000
8. 0000
9. 00000000

0000 **3**00000000000000

000 00000000“000000”

1. 00000000
2. 00000000000000
3. 00000000000000 01/2025
4. 000“00000000000000”
5. 00000000000000
6. 00000 **Customers → [Customer] → Billing**
7. 00 **"Add Payment Method"**
8. 00000000000000
9. 00000000
10. 00000000
11. 00000000
12. 00000000“0000000000”

0000 **4**00000000000000

000 0000000000000000

1. 0000000000 Visa 00 1111
2. 0000000000000000
3. 000000 Mastercard 00 2222000000
4. 000000 Amex 00 333300000000
5. 0000000000000000 Mastercard
6. 000 Visa 0000000000

📅 5📅📅📅📅📅📅📅📅

📅📅📅📅📅

1. 📅📅📅📅 **Billing** → **Expiring Cards**
2. 📅📅📅“📅📅📅 30 📅”
3. 📅📅 15 📅📅📅📅📅📅📅
4. 📅📅📅 → **"Send Reminder Emails"**
5. Mailjet 📅📅📅📅📅📅📅📅📅
6. 📅📅📅📅📅📅📅
7. 📅📅📅📅📅📅
8. 📅📅📅📅📅📅
9. 📅📅📅📅📅📅

📅📅📅📅

“📅📅📅📅📅📅”

- 📅📅 📅📅📅📅📅📅📅📅📅📅📅📅📅📅📅📅
- 📅📅📅📅
 - 📅📅📅📅📅
 - 📅📅📅📅📅📅📅
 - 📅📅📅📅📅📅
 - 📅📅📅📅📅📅📅📅📅

“📅📅📅📅📅”📅📅📅📅

- 📅📅 📅📅📅📅 API 📅📅📅📅📅
- 📅📅📅📅
 - 📅📅📅📅📅
 - 📅📅📅📅📅
 - 📅📅📅📅📅📅📅📅📅📅📅
 - 📅📅📅📅📅📅📅📅📅
 - 📅📅📅📅📅📅📅📅📅

○○○○○○○○○○○○○○○○

- ○○○ ○○○○○○○○○○○○○○○
- ○○○○
 - ○○○○○○○○○○○
 - ○○○○○○○○○○○○○○○○○

○○○○○○○○○○○○“○○○○○○”○○

- ○○○ ○○○○○○○○○○○○○
- ○○○○
 - ○○○○
 - ○○○○○○○○○○○○○○○○○
 - ○○○○○○○○○○○○○○○○○

○○○○○○○○

- ○○○ ○○○○○○○○○○○
- ○○○○
 - ○○○○○○○○○○○
 - ○○○○○○○○○○○○○
 - ○○○○○○○○○○○○○

○○○○○○○○○○

- ○○○ ○○○○ 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 API



OmniCRM API integrates with various payment gateways including Stripe, PayPal, Square, Adyen, Braintree, and others.

API endpoints and documentation

Ansible Playbook for API integration. See [Playbooks](#) for details.



1. [API](#)
2. [API](#)
3. [API](#)
4. [API](#)
5. [API](#)
6. [API](#)
7. [API](#)
8. [API](#)

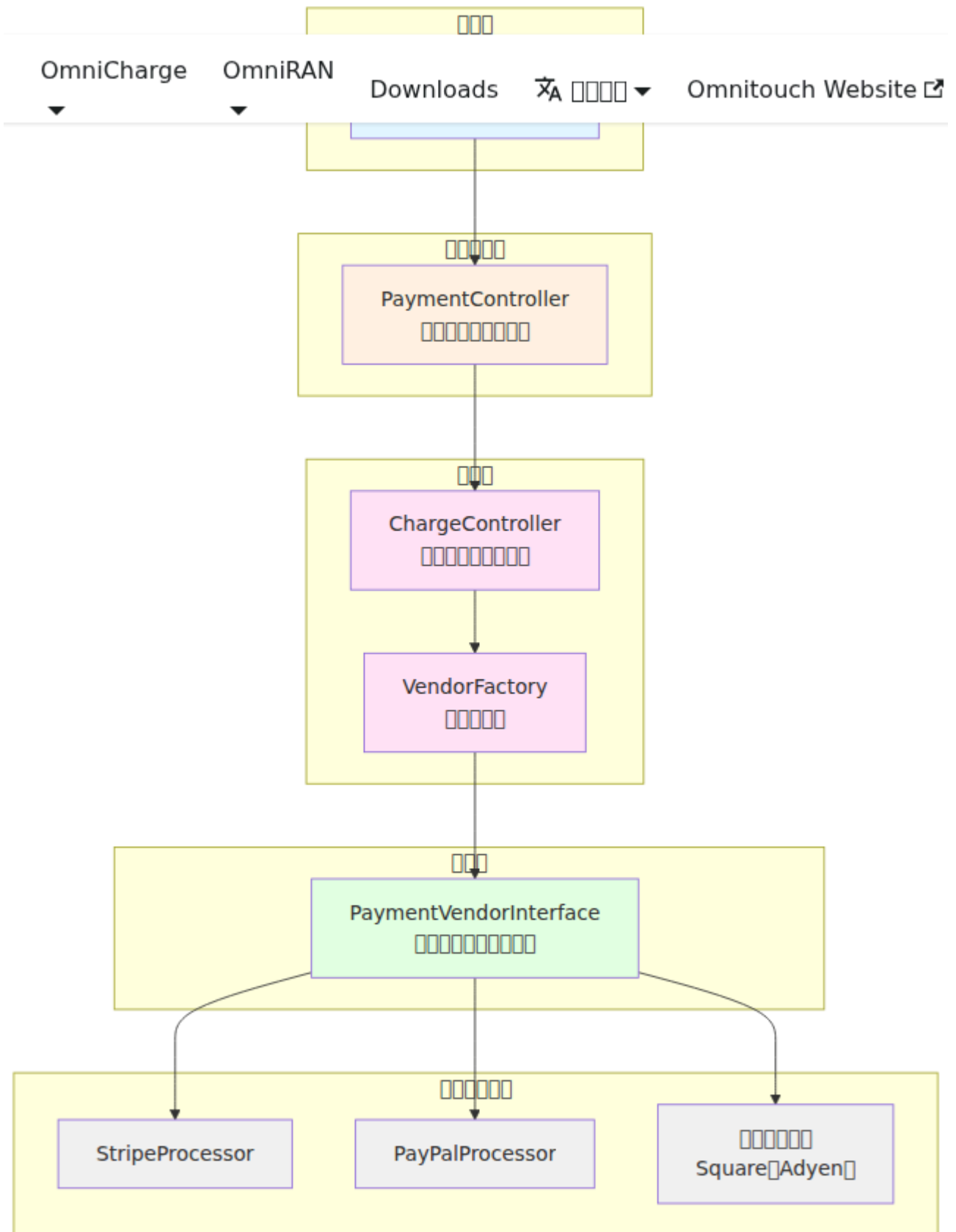


API endpoints and documentation

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

- □ □□□□□□□
- □ □□□□□□□API□□□□□

□□□



Payment Gateway

Payment Gateway: Square, Adyen, Braintree, OmniCRM

Payment Gateway: Stripe, PayPal, **API**

Payment Gateway



Payment Gateway

1. `PaymentVendorInterface`
2. `VendorFactory`
3. `PaymentGateway`
4. `API`

Payment Gateway

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

Payment Gateway

Payment Gateway 1-2



Stripe

Stripe 3D

1. Stripe API

- <https://stripe.com>
- → API
- pk_live_... pk_test_...
- sk_live_... sk_test_...

- pk_test_/sk_test_
- pk_live_/sk_live_
- API

2.

crm_config.yaml

```
payment_vendors:  
  stripe:  
    api_key: "sk_live_YOUR_SECRET_KEY_HERE"  
    publishable_key: "pk_live_YOUR_PUBLISHABLE_KEY_HERE"
```

```
export STRIPE_SECRET_KEY="sk_live_YOUR_SECRET_KEY_HERE"  
export STRIPE_PUBLISHABLE_KEY="pk_live_YOUR_PUBLISHABLE_KEY_HERE"
```

3.

.env

```
REACT_APP_STRIPE_PUBLISHABLE_KEY=pk_live_YOUR_PUBLISHABLE_KEY_HERE
```

Placeholder text line

PayPal

PayPal configuration text

Placeholder text line

Placeholder text line

```
payment_vendors:
  paypal:
    client_id: "AXx_YOUR_CLIENT_ID_HERE"
    client_secret: "ELx_YOUR_CLIENT_SECRET_HERE"
    mode: "live" # "sandbox"
```

PCI

OmniCRM PCI configuration text

- Placeholder text item 1
- Placeholder text item 2
- Placeholder text item 3
- Placeholder text item 4

Placeholder text line

OmniCRM Stripe API configuration text

Placeholder text line

Placeholder text line



Merchant - Square

```

{
  "vendor": "square",
  "vendor_payment_method_id": "sq_card_abc123",
  "payment_type": "card"
}

```

Merchant - Stripe

Merchant

Merchant

PaymentMethod

Merchant - Stripe

```

{
  "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", // □□□□□□□□ID
  "amount": 200.00,
  "currency": "USD",
  "status": "succeeded", // 'succeeded', 'failed',
  'refunded'
  "captured_at": "2025-12-27T10:30:00Z",
  "vendor_response": {}, // □□□□□□□□□□
  "meta": {}
}
```

WalletAccount

□□□□□□□□□□□□□□□□

```
{
  "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

□□□□□□□□□□□□□□

```
{
  "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", // □□□□□□
  "reference_id": 103,
  "meta": {},
  "created_at": "2025-12-27T10:35:00Z"
}
```

□□□□

□□□□□□□□□□□□□□□□**□□□□**□

00

API ID invoice_id
“”

** **

0000

```

{
  "invoice_id": 12345,
  "invoice_number": "INV-2025-000001", // INV-YYYY-NNNNNN
  "customer_id": 123,
  "title": "",
  "paid": true, // 
  "void": false, // 
  "payment_reference": "ch_xxxxx", // /ID
  "payment_type": "stripe_capture", // 
  "payment_time": "2025-12-27T10:30:00",
  "start_date": "2025-12-01",
  "end_date": "2025-12-31",
  "due_date": "2026-01-15",
  "retail_cost": 500.00, // 
  "wholesale_cost": 250.00
}

```

INV-YYYY-NNNNNN 11

000000

retail_cost

0000

- “”
-
-
-

0000

- 零售店交易
- 零售店交易
- 零售店交易 - 零售店交易

零售店交易

```
// 1. 零售店交易
{
  "transaction_id": 7001,
  "invoice_id": 12345,
  "retail_cost": 100.00, // 零售 = 零售
  "title": "零售"
}

// 2. 零售店交易
{
  "transaction_id": 7002,
  "invoice_id": 12345,
  "retail_cost": -100.00, // 零售 = 零售
  "title": "零售12345",
  "payment_type": "stripe_capture",
  "payment_reference": "ch_xxxxx"
}

// 零售$0.00 → 零售
```

零售店交易 payment_reference payment_type payment_time

- 零售店交易 invoice_id 零售店交易 Hayden
- 零售店交易/零售店交易
- 零售店交易

零售

零售店交易

API

Stripe API 的端点

端点

1. 创建支付意图
2. 创建支付链接
3. 创建支付方法
4. 创建支付方法
 - 创建支付方法
 - 更新支付方法

Stripe API 的端点 `Stripe API` `Stripe API`

API

端点 URL `/api/payments/`

端点 `**API**`

端点

端点

端点 `POST /api/payments/methods`

端点

```
curl -X POST https://your-domain.com/api/payments/methods \
-H "Content-Type: application/json" \
-H "Authorization: Bearer YOUR_API_KEY" \
-d '{
  "customer_id": 123,
  "vendor": "stripe",
  "payment_token": "pm_xxxxx",
  "is_default": false,
  "card_nickname": "My Visa Card"
}'
```

□□□□

```
{
  "customer_id": 123,
  "vendor": "stripe", // 'stripe', 'paypal'□□□□
  "payment_token": "pm_xxxxx", // SDK□□□□□□□□□□
  "is_default": false, // □□□□□□□□□□
  "card_nickname": "My Visa Card" // □□□□□□□□
}
```

□□□201□□□□□□

```
{
  "success": true,
  "message": "Success",
  "data": {
    "payment_method_id": 789,
    "customer_id": 123,
    "vendor": "stripe",
    "payment_type": "card",
    "card_brand": "visa",
    "card_last4": "4242",
    "card_exp_month": 12,
    "card_exp_year": 2025,
    "card_nickname": "My Visa Card",
    "is_default": false,
    "status": "active"
  }
}
```

Success

Success

GET /api/payments/methods?customer_id={id}

curl

```
curl -X GET "https://your-domain.com/api/payments/methods?
customer_id=123" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

200 OK

```
{
  "success": true,
  "data": [
    {
      "payment_method_id": 789,
      "customer_id": 123,
      "vendor": "stripe",
      "payment_type": "card",
      "card_brand": "visa",
      "card_last4": "4242",
      "is_default": true
    },
    {
      "payment_method_id": 790,
      "customer_id": 123,
      "vendor": "paypal",
      "payment_type": "paypal",
      "paypal_email": "user@example.com",
      "is_default": false
    }
  ]
}
```

□□□□□□□□

□□□□□□□□□□

□□□ GET /api/payments/methods/default?customer_id={id}

□□□

```
curl -X GET "https://your-domain.com/api/payments/methods/default?
customer_id=123" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

□□□200 OK□□

```
{
  "success": true,
  "data": {
    "payment_method_id": 789,
    "vendor": "stripe",
    "payment_type": "card",
    "card_brand": "visa",
    "card_last4": "4242",
    "is_default": true
  }
}
```

□□□□□□□□

□□□□□□□□□□

□□□ PUT /api/payments/methods/set-default

□□□

```
curl -X PUT https://your-domain.com/api/payments/methods/set-
default \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "payment_method_id": 790
  }'
```

□□□200 OK□□

```
{
  "success": true,
  "message": "□□□□□□□□□□",
  "data": {
    "payment_method_id": 790,
    "is_default": true
  }
}
```

DELETE

DELETE

DELETE /api/payments/methods/{payment_method_id}?customer_id={id}

curl

```
curl -X DELETE "https://your-domain.com/api/payments/methods/789?customer_id=123" \
-H "Authorization: Bearer YOUR_API_KEY"
```

200 OK

```
{
  "success": true,
  "message": "Deleted successfully"
}
```

API

API

1. API

API

POST /api/payments/charge

curl

```
curl -X POST https://your-domain.com/api/payments/charge \  
-H "Content-Type: application/json" \  
-H "Authorization: Bearer YOUR_API_KEY" \  
-d '{  
  "customer_id": 123,  
  "amount": 50.00,  
  "currency": "USD",  
  "payment_method_id": 789,  
  "metadata": {  
    "order_id": "12345"  
  }  
'
```

□□□□

```
{  
  "customer_id": 123,  
  "amount": 50.00,  
  "currency": "USD", // □□□USD  
  "payment_method_id": 789, // □□ - □□□□□□□□□□  
  "vendor": "stripe", // □□□□payment_token□□□□  
  "payment_token": "pm_xxxxx", // □□ - □□□□□  
  "save_method": false, // □□□□□□□□□□□□□□□□  
  "metadata": {  
    "order_id": "12345"  
  }  
}
```

□□□200 OK□□


```
{
  "success": true,
  "message": "成功",
  "data": {
    "transaction_id": "ch_xxxxx",
    "capture_id": 101,
    "amount": 50.00,
    "currency": "USD",
    "status": "succeeded"
  }
}
```

2. 如何接收支付通知

如何接收支付通知

通过 `POST /api/payments/invoice`

步骤

1. 创建支付通知
2. 接收支付通知
3. 验证支付通知  是否有效
4. 处理支付通知
5. 更新支付通知

步骤

POST /api/payments/authorize/hold

요청 파라미터

1. 고객 ID
2. 결제 금액 - 결제 단위
3. 결제 방법 ID
4. 예약 ID (선택 사항)

요청 예시

```
curl -X POST https://your-domain.com/api/payments/authorize/hold \
-H "Content-Type: application/json" \
-H "Authorization: Bearer YOUR_API_KEY" \
-d '{
  "customer_id": 123,
  "amount": 200.00,
  "payment_method_id": 789,
  "use_wallet": true,
  "metadata": {
    "reservation_id": "RES-001"
  }
}'
```

응답 예시

```
{
  "customer_id": 123,
  "amount": 200.00,
  "currency": "USD",
  "payment_method_id": 789,           // 결제 방법 ID
  "vendor": "stripe",               // 결제 수단 payment_token
  "payment_token": "pm_xxxxx",     // 결제 토큰
  "save_method": false,
  "use_wallet": true,               // 지갑 사용 여부 true
  "metadata": {
    "reservation_id": "RES-001"
  }
}
```

200 OK

```
{
  "success": true,
  "message": "Success",
  "data": {
    "authorization_id": 301,
    "vendor_authorization_id": "auth_xxxxx",
    "amount": 200.00,
    "currency": "USD",
    "status": "authorized",
    "wallet_balance": 150.00,
    "wallet_to_use": 150.00,           // wallet
    "card_amount": 50.00,           // card
    "message": "Card $50 and wallet $200"
  }
}
```

2

POST /api/payments/capture/{authorization_id}

1. Success - OK
2. Success
3. Success
4. Success/Success

```
curl -X POST https://your-domain.com/api/payments/capture/301 \
-H "Content-Type: application/json" \
-H "Authorization: Bearer YOUR_API_KEY" \
-d '{
  "amount": 200.00,
  "metadata": {
    "invoice": true,
    "title": "测试",
    "description": "3测试",
    "wholesale_cost": 100.00,
    "contract_days": 3,
    "send_email": true
  }
}'
```

测试

```
{
  "amount": 200.00, // 金额 - 测试金额
  "metadata": {
    "invoice": true, // 是否生成发票
    "create_transaction": true, // 是否创建交易
    "title": "测试",
    "description": "3测试",
    "wholesale_cost": 100.00,
    "contract_days": 3,
    "send_email": true // 是否发送邮件
  }
}
```

测试200 OK

```
{
  "success": true,
  "message": "成功",
  "data": {
    "capture_id": 103,
    "transaction_id": "ch_xxxxx",
    "authorization_id": 301,
    "amount": 200.00,
    "currency": "USD",
    "status": "succeeded",
    "wallet_credit": { // 入账
      "ledger_id": 401,
      "amount": 50.00
    },
    "wallet_debit": { // 扣款
      "ledger_id": 402,
      "amount": 200.00
    },
    "transaction": { // 交易
      "transaction_id": 7001
    },
    "invoice": { // 发票
      "invoice_id": 12345,
      "invoice_number": "INV-2025-000001"
    }
  }
}
```

3

POST /api/payments/release/{authorization_id}

```
curl -X POST https://your-domain.com/api/payments/release/301 \
-H "Authorization: Bearer YOUR_API_KEY"
```

200 OK

```
{
  "success": true,
  "message": "成功",
  "data": {
    "authorization_id": 301,
    "vendor_authorization_id": "auth_xxxxx",
    "status": "released",
    "released_at": "2025-12-27T10:45:00Z"
  }
}
```

成功

4. 成功

成功

POST /api/payments/topup

成功 → 成功 → 成功

成功

```
curl -X POST https://your-domain.com/api/payments/topup \
-H "Content-Type: application/json" \
-H "Authorization: Bearer YOUR_API_KEY" \
-d '{
  "customer_id": 123,
  "amount": 30.00,
  "payment_method_id": 789,
  "service_uuid": "svc-uuid-123",
  "imsi": "123456789012345",
  "days": 30,
  "metadata": {
    "is_rental": false
  }
}'
```

成功/成功

```
curl -X POST https://your-domain.com/api/payments/topup \  
-H "Content-Type: application/json" \  
-H "Authorization: Bearer YOUR_API_KEY" \  
-d '{  
  "customer_id": 1,  
  "amount": 5.00,  
  "vendor": "stripe",  
  "payment_token": "pm_xxxxx",  
  "service_uuid": "hotspot-uuid",  
  "imsi": "123456789012345",  
  "days": 1,  
  "metadata": {  
    "is_rental": true,  
    "billing_email": "user@example.com"  
  }  
'
```

200 OK

```
{  
  "success": true,  
  "message": "Success",  
  "data": {  
    "transaction_id": "ch_xxxxx",  
    "authorization_id": 302,  
    "capture_id": 104,  
    "amount": 30.00,  
    "status": "succeeded",  
    "provision_result": {  
      "success": true,  
      "topup_result": {...},  
      "service_uuid": "svc-uuid-123",  
      "imsi": "123456789012345",  
      "days": 30  
    },  
    "payment_method_saved": false // false  
  }  
}
```

5. 請求の送信

このステップでは、APIにデータを送信します。

curl `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"
  }
}'
```

このステップでは、APIにデータを送信します。

```
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"
  }
}'
```

curl `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"
}'
```

curl

```
{
  "transaction_id": "ch_xxxxx", // 交易ID
  "vendor": "stripe", // 'stripe', 'paypal'
  "amount": 50.00, // 金额 - 退款金额
  "reason": "customer_request" // 原因
}
```

返回200 OK

```
{
  "success": true,
  "message": " refunded ",
  "data": {
    "refund_id": "re_xxxxx",
    "amount": 50.00,
    "status": "succeeded"
  }
}
```

API

URL `/api/wallet/`

GET `/api/wallet/balance?customer_id={id}`

```
curl -X GET "https://your-domain.com/api/wallet/balance?
customer_id=123" \
-H "Authorization: Bearer YOUR_API_KEY"
```

200 OK

```
{
  "customer_id": 123,
  "balance": 150.50,
  "currency": "USD"
}
```

GET /api/wallet/info?customer_id={id}

```
curl -X GET "https://your-domain.com/api/wallet/info?customer_id=123" \
-H "Authorization: Bearer YOUR_API_KEY"
```

200 OK

```
{
  "customer_id": 123,
  "wallet": {
    "wallet_account_id": 456,
    "balance": 150.50,
    "currency": "USD",
    "auto_recharge_enabled": true,
    "auto_recharge_amount": 100.00,
    "auto_recharge_threshold": 10.00,
    "low_balance_warning_threshold": 10.00
  }
}
```

POST /api/wallet/topup

/PayPal →

```
curl -X POST https://your-domain.com/api/wallet/topup \  
-H "Content-Type: application/json" \  
-H "Authorization: Bearer YOUR_API_KEY" \  
-d '{  
  "customer_id": 123,  
  "amount": 100.00,  
  "payment_method_id": 789  
'
```

请求

```
{  
  "customer_id": 123,  
  "amount": 100.00,  
  "currency": "USD", // 币种USD  
  "payment_method_id": 789 // 支付方式 - 信用卡  
}
```

返回200 OK

```
{  
  "success": true,  
  "payment": {  
    "transaction_id": "ch_xxxxx",  
    "amount": 100.00  
  },  
  "wallet": {  
    "ledger_id": 503,  
    "balance_after": 250.50  
  },  
  "message": "充值100.00成功"  
}
```

返回数据

成功返回数据

```
GET /api/wallet/transactions?customer_id={id}&limit={n}&offset={n}&type={type}
```

curl

```
curl -X GET "https://your-domain.com/api/wallet/transactions?customer_id=123&limit=50&offset=0&type=credit" \
-H "Authorization: Bearer YOUR_API_KEY"
```

parameters

- `customer_id` ID
- `limit` 50
- `offset` 0
- `type` 'credit' | 'debit' | 'refund' | 'adjustment'

200 OK

```
{
  "customer_id": 123,
  "count": 2,
  "transactions": [
    {
      "ledger_id": 501,
      "transaction_type": "credit",
      "amount": 100.00,
      "balance_before": 150.50,
      "balance_after": 250.50,
      "description": "□□□",
      "reference_type": "payment_capture",
      "reference_id": 103,
      "created_at": "2025-12-27T10:35:00Z"
    },
    {
      "ledger_id": 502,
      "transaction_type": "debit",
      "amount": 50.00,
      "balance_before": 250.50,
      "balance_after": 200.50,
      "description": "□□□□",
      "reference_type": "service_charge",
      "reference_id": 789,
      "created_at": "2025-12-27T11:00:00Z"
    }
  ]
}
```

API

API

メソッド	API	レスポンス
POST	<code>/api/payments/methods</code>	成功
GET	<code>/api/payments/methods?customer_id={id}</code>	成功
GET	<code>/api/payments/methods/default?customer_id={id}</code>	成功
PUT	<code>/api/payments/methods/set-default</code>	成功
DELETE	<code>/api/payments/methods/{id}?customer_id={id}</code>	成功

API

メソッド	API	レスポンス
POST	<code>/api/payments/charge</code>	成功
POST	<code>/api/payments/invoice</code>	成功
POST	<code>/api/payments/topup</code>	成功
POST	<code>/api/payments/authorize/hold</code>	成功
POST	<code>/api/payments/capture/{id}</code>	成功
POST	<code>/api/payments/release/{id}</code>	成功
POST	<code>/api/payments/rental</code>	成功
POST	<code>/api/payments/refund</code>	成功

□□□□

□□	□□	□□
GET	<code>/api/wallet/balance?customer_id={id}</code>	□□□□□□
GET	<code>/api/wallet/info?customer_id={id}</code>	□□□□□□ + □□
POST	<code>/api/wallet/topup</code>	□□□□□□□□□□
GET	<code>/api/wallet/transactions?customer_id={id}</code>	□□□□□□

PayPal□□□□

◆◆◆ □	□□	□□
POST	<code>/api/payments/paypal/vault/setup-token</code>	□Card Fields SDK□ □PayPal□□□□
POST	<code>/api/payments/paypal/vault/finalize</code>	□□PayPal□□□□□□□□ □□
POST	<code>/api/payments/paypal/vault/update-setup-token</code>	□□3DS/SCA□□□□□□ □□

□□□□

□□**1**□□□□□□□□□□

□□□□□□□□□□□□□□□□30□□□□□□

```
# 创建支付记录
curl -X POST https://your-domain.com/api/payments/invoice \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 99.99,
    "metadata": {
      "service_id": 456,
      "contract_days": 30,
      "description": "30天会员"
    }
  }'
```

返回

```
{
  "success": true,
  "message": "支付成功",
  "data": {
    "service_amount": 99.99,
    "wallet_portion_used": 50.00, // 钱包$50
    "card_portion_used": 49.99, // 卡片$49.99
    "final_balance": 0.00
  }
}
```

2. 创建订单

创建订单时，需要指定商品 ID 和数量。同时，还需要指定支付方式。如果支付方式为钱包支付，则需要指定钱包 ID。如果支付方式为卡片支付，则需要指定卡片 ID。创建订单的 API 接口如下：

```
# 実行コマンド
curl -X POST https://your-domain.com/api/payments/topup \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 149.99,
    "payment_method_id": 789,
    "service_uuid": "fiber-svc-uuid",
    "imsi": "123456789012345",
    "days": 30,
    "metadata": {
      "service_type": "fiber_internet"
    }
  }'
```

結果

1. 請求額\$149.99
2. 請求内容
3. 請求額 → 請求内容
4. 請求額 → 請求内容

3. 請求内容

請求内容にWiFiが含まれていることを確認する

```
# 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

4 充值失败 + 返回

充值失败

```

# 1. authorize - authorize
curl -X POST https://your-domain.com/api/payments/authorize/hold \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 500.00,
    "use_wallet": true,
    "metadata": {
      "reservation_id": "RES-2025-001"
    }
  }'
```

```

# 2. check status {"authorization_id": 301, "status": "authorized"}
```

```

# 3. capture payment - capture
curl -X POST https://your-domain.com/api/payments/capture/301 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "metadata": {
      "invoice": true,
      "title": "Reservation - 301",
      "description": "3020001202700300",
      "send_email": true
    }
  }'
```

```

# 4. release payment - release
curl -X POST https://your-domain.com/api/payments/release/301 \
  -H "Authorization: Bearer YOUR_API_KEY"
```

5. Stripe

Visa

```
# 1 Stripe.js Stripe
# const {token} = await stripe.createToken(cardElement);

# 2
curl -X POST https://your-domain.com/api/payments/methods \
-H "Content-Type: application/json" \
-H "Authorization: Bearer YOUR_API_KEY" \
-d '{
  "customer_id": 123,
  "vendor": "stripe",
  "payment_token": "pm_xxxxx",
  "is_default": true,
  "card_nickname": "Visa"
}'
```

6

\$100

```
curl -X POST https://your-domain.com/api/wallet/topup \
-H "Content-Type: application/json" \
-H "Authorization: Bearer YOUR_API_KEY" \
-d '{
  "customer_id": 123,
  "amount": 100.00
}'
```

```
{
  "success": true,
  "payment": {
    "transaction_id": "ch_xxxxx",
    "amount": 100.00
  },
  "wallet": {
    "balance_after": 250.50
  },
  "message": "成功支付100.00元"
}
```

成功支付

成功支付

成功支付400元

```
{
  "error": "成功支付50.00元"
}
```

成功支付400元

```
{
  "error": "成功支付"
}
```

成功支付400元

```
{
  "error": "customer_id amount"
}
```

404

```
{  
  "error": "999"  
}
```

500

```
{  
  "error": "500"  
}
```

404

404

404

1. Stripe/PayPal

404

PayPal

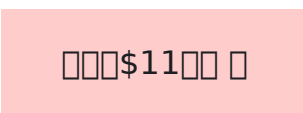
POST /api/payments/refund

404

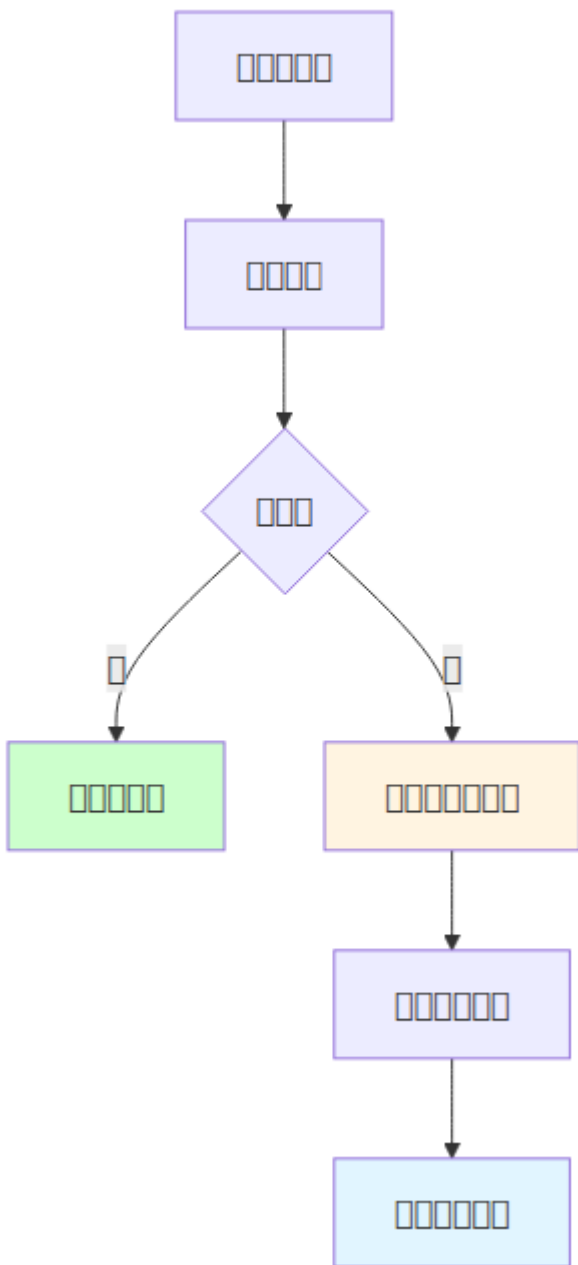
1. 100
- 2.
- 3.
4. 100 5-10 +
 - o 100
 - o
 - o
 - o

1 \$1 + \$10

\$1 \$10



OmniCRM



API

```
curl -X POST https://your-domain.com/api/payments/invoice \
-H "Content-Type: application/json" \
-H "Authorization: Bearer YOUR_API_KEY" \
-d '{
  "customer_id": 123,
  "amount": 10.00,
  "metadata": {
    "addon_id": 456
  }
}'
```

API

```
{
  "success": true,
  "message": "成功",
  "data": {
    "customer_id": 123,
    "service_amount": 10.00,
    "routing_mode": "hybrid",
    "initial_balance": 1.00,
    "wallet_portion_used": 1.00, // 財布$1
    "card_portion_used": 9.00, // カード$9
    "charged_amount": 9.00, // ← カード
    "wallet_credited": 9.00, // ← 財布
    "wallet_debited": 10.00, // ← カード
    "final_balance": 0.00
  }
}
```

2\$50 + \$30

\$50 \$30

□□□□\$50

□□□□
\$30 - \$50 = \$0

□□□

□□□□

□□□□

□□□□-\$30

□□□□\$20
□□□□\$0
□□□□\$30 □

API□□□

```

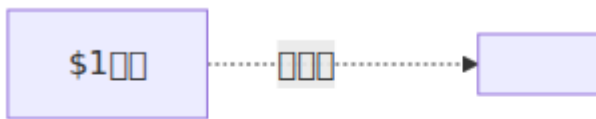
{
  "success": true,
  "data": {
    "service_amount": 30.00,
    "initial_balance": 50.00,
    "charged_amount": 0,           // ← 0000
    "wallet_debited": 30.00,
    "final_balance": 20.00,
    "payment_method_used": false  // ← 0000
  }
}

```

300000000000

00000000\$500000000000\$1500

1000000 POST /api/payments/authorize/hold00



API000

```

{
  "authorization_id": 301,
  "amount": 500.00,
  "status": "authorized",
  "wallet_balance": 150.00,
  "wallet_to_use": 150.00,
  "card_amount": 350.00,
  "message": "0000$3500000000000000000000$50000000"
}

```

2000000 POST /api/payments/capture/30100

□□□□

□□□□\$350

□□□□+\$350
□□□\$500

□□□□-\$500
□□□□

□□□□\$0
□□□□\$350

□□□□

- □ □□□□□□□\$350□□□□\$500
- □ □□□□□□□□□□□
- □ □□□□□□□□□
- □ □□□□□□□□□

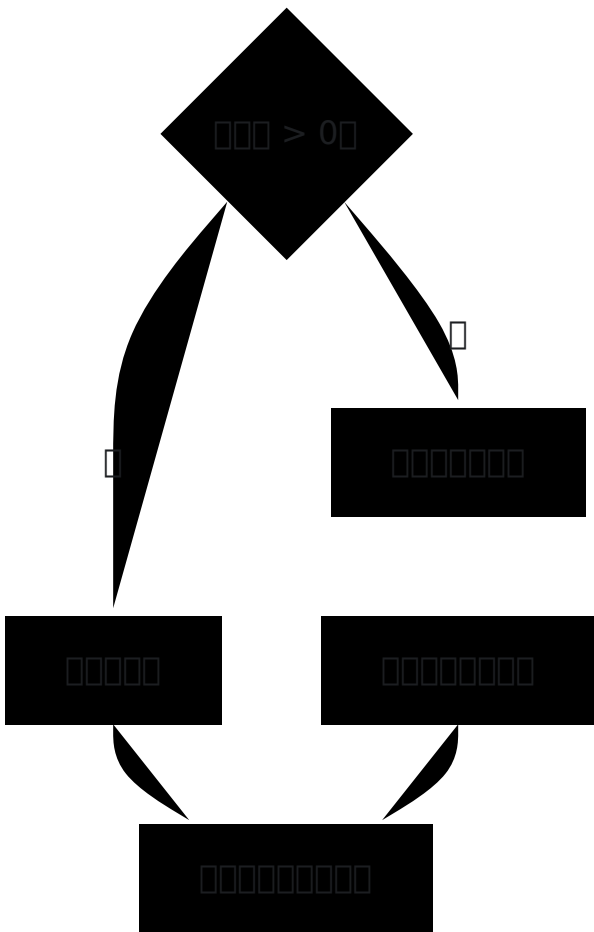
□□□□

□□□□□□□□□□□□

total

total > 0

```
total
wallet_portion = min(
    total,
    card_limit
)
card_portion = total -
    wallet_portion
```



00

00000

1. 0000000000
2. 00000000 - 00000
3. 0000 > 000000000000
4. 00000000000
5. 00000000000
6. 00000000000000000000

0000000

00000000000000000000

0000 - 0000000000000000

```
curl -X POST https://your-domain.com/api/payments/charge \  
-H "Content-Type: application/json" \  
-H "Authorization: Bearer YOUR_API_KEY" \  
-d '{  
  "customer_id": 123,  
  "amount": 100.00,  
  "metadata": {  
    "routing_mode": "bypass"  
  }  
'
```

000000\$100000000\$10000000\$100000\$0

0000000000000000/000000000000

0000

1. 0000000000000000000000000000

2. ...
3. ...
4. ...
5. ...
6. ...
7. ...7...
8. ...

Playbook

Ansible playbook

1. ...try/catch...
2. **authorization_id** - ...
3. **API** - ...
4. ...2...
5. ...

Playbook

API

API Authorization

```
Authorization: Bearer YOUR_API_KEY
```

API

Payment Gateway

Integration

- **Stripe** - Stripe ACH
- **PayPal** - PayPal Card Fields SDK

Deployment

Deployment to production

Automation

Tools

- **Playbooks** - Ansible playbook
 - Infrastructure
 - CI/CD
 - Deployment
 - Monitoring

Configuration

- **Stripe** - Stripe configuration
- **PayPal** - PayPal configuration
- **Card Fields** - Card Fields configuration
- **Payment Gateway** - Payment Gateway configuration
- **Payment Gateway** - Payment Gateway configuration

Deployment

- **Stripe** → Stripe configuration
- **Stripe/PayPal** → Stripe/PayPal configuration

- 000000 → 0000000
 - 000000 → 000Playbook00
-



API API API API

Transactions </payments_transaction>

Mailjet <integrations_mailjet> PDF

OmniCRM API

OmniCRM API HTML templates Jinja2 configuration

Templates

OmniCRM API invoice templates/

Templates

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

Config

OmniCRM API crm_config.yaml

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

Configuration

OmniCRM API Jinja2 configuration

□□□□

- `{{ 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>
```

部署命令

命令 1

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

命令 2 HTML/CSS

命令 `your_company_invoice_template.html` 部署

部署步骤

1. 部署

```
<!-- 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>Payment</h3>
  <p><strong>Pay to: </strong> https://yourcompany.com/pay </p>
</div>
<p><strong>Account Details</strong></p>
<ul>
  <li>Your Company Ltd</li>
  <li>BSB 123-456</li>
  <li>987654321</li>
  <li>{{ invoice_number }}</li>
</ul>
</div>
```

5. Terms

```
<div class="terms">
  <h4>Terms</h4>
  <p>Payment due within 30 days</p>
  <p>Late payment fee of 2%</p>
  <p>Contact billing@yourcompany.com</p>
</div>
```

3. Configuration

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 deployment
2. Deploy

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 %}

```

テンプレート

ファイル名

```

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 商品金额 tax_percentage 税率 0%

商品名称

商品数量

```

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

```

PDF 生成

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

使用 **CSS**

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

注意

- 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. □□□□□□
 - □□□□□□□□
 - □□□□□□□□

- 〇〇〇〇〇〇〇
- 〇〇〇〇〇〇〇〇〇

3. PDF 〇〇〇

- 〇〇〇〇〇〇〇<5MB〇
- 〇〇〇〇〇〇〇
- 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
- 〇〇〇〇〇
- 〇〇〇〇〇〇〇〇〇〇〇

5. 〇〇〇〇〇〇〇〇

- PDF 〇〇〇〇〇〇〇〇〇
- 〇〇〇〇〇〇〇 Mailjet 〇〇〇15MB〇
- 〇 Gmail〇Outlook〇Apple Mail 〇〇〇

〇〇〇〇〇〇〇〇〇〇〇〇〇〇

〇〇〇〇〇 API 〇〇〇〇〇〇〇〇〇

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

〇〇〇〇〇〇〇〇

〇〇〇〇〇〇〇

- 〇〇〇 〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
- 〇〇〇 〇〇〇

PDF 〇〇〇〇〇

- 〇〇〇 〇〇〇〇 CSS 〇〇
- 〇〇〇 〇〇 CSS 2.1 〇〇〇〇〇〇〇 WeasyPrint 〇〇〇 CSS

URL

- URL 格式
- HTTPS URL

Table Layout

- Table Layout
- `table-layout: fixed`

Font

- Font
- Arial, Times New Roman

PDF

- HTML to PDF (WeasyPrint)
- PDF to HTML (WeasyPrint)

PDF

PDF, OmniCRM, PDF, PDF

PDF

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

2. 查詢 - 查詢所有已過期
 - 查詢 invoice_id 查詢 PDF
 - 查詢所有已過期 PDF
 - 增加 last_accessed 查詢所有已過期
3. 查詢 - 查詢所有已過期 PDF 查詢
 - 查詢所有已過期/查詢所有已過期
 - 查詢所有已過期
 - 查詢所有已過期

查詢

- 查詢 - 查詢所有已過期 PDF 查詢
- 查詢 - 查詢所有已過期 PDF 查詢
- 查詢 - 查詢 PDF 查詢 CPU 查詢
- 查詢 - 查詢所有已過期 PDF 查詢

查詢

查詢 PDF 查詢所有已過期 PDF 查詢

- 查詢所有已過期 90 查詢
- 查詢所有已過期 30 查詢
- 查詢所有已過期 PDF 查詢

API 查詢

查詢 API 查詢 UI 查詢

- 查詢所有已過期 PDF 查詢
- 查詢所有已過期 PDF 查詢
- 查詢所有已過期 PDF 查詢

查詢所有已過期 PDF 查詢

```
-- 查詢所有已過期 PDF 查詢 MySQL 查詢
DELETE FROM Invoice_PDF_Cache;
```

crm_config.yaml

crm

crm

crm

1. crm → [crm]
2. crm
3. crm

crm

1. crm → crm
2. crm

crm

crm

crm

- crm - crm

- 1000000 - 1000000000000000000000
- 1000000 - 1000000000000000000000
- 1000000 - 1000000000000000000000

1000000

- 1000 1000000 "k" 1000000\$1.5k
- 1000,000 1000000 "M" 1000000\$2.3M
- 1000,000,000 1000000 "B" 1000000\$1.1B

1000000

- "1000" "1000" 1000000000000000000000
- 1000000000000000000000
- 1000000000000000000000
- 1000000000000000

1000000

1000000000000000000000

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

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

" + " "



XX

XXXXXXXXXX

- XXXX - XXXX IDXXXXXXXXXXXXXXXXXXXXXXXXXXXX
- XXXX - XXXXXXXXXXXXXXXXXXXXXXX
 - XXXX
 - XXXXX
 - XXXX
 - XXXXXXXXXXXXX \$10.00 (10%)
 - XXXXXXXXXXXXX "-"
- XXXX - XXXXXXX
 - XXXX
 - XXXXXXXXXXXXXXXXXXXXX
 - XXXXXXXXXXXXXXX
 - XXXXXXXXXXXX + XX
- XXXX - XXXXXXXXXXXXXXX
 - XX **PDF** - XXXX PDFXXXXXXXXXX
 - XXXXXXX - XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 - XXXX - XXXXXXXXXXXXXXXXXXXXXXX
 - XX - XX Stripe XXXXXXXXXXXX Stripe XX

- 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
 - 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-xxxx-xxxx-xxxx"

退款

退款 ()

退款

- 退款
- 退款 **Stripe**
- `payment_reference` Stripe ID
- 退款

退款

1. Stripe
2. ()
- 3.

4. " "
5. Stripe
 - Stripe API
 - Stripe
 - `refund_reference`
6. " "

7. 0000000

???.00000000

- 0000000000000000
- 0000 "0000"
- 0000000000
- 000000000000003-7 000000
- Stripe 0000000000

000000

- 0000000000POS 000000000000000000
- 000000000000000000
- 00000000

00000000

00

000000000000000000

- 00 ID
- 0000
- 0000000000000000

00

00000000000000

000000

- 0000 - 000000000000
- 0000 - 00000000000000

000000

- [] - []
- [] - []

[]

[]

- **ID** - [] ID []/[]
- [] - []
- [] - []
- [] - []
- [] - []/[]
- [] - []

[] ↔ []

[]

[]10 25 50 [] 100 []

[]

[] **1**[]

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

7.
 -
 -
 -
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. 𐀀𐀁𐀂𐀃𐀄𐀅𐀆𐀇𐀈𐀉𐀊𐀋𐀌𐀍𐀎𐀏𐀐𐀑𐀒𐀓𐀔𐀕𐀖𐀗𐀘𐀙𐀚𐀛𐀜𐀝𐀞𐀟𐀠𐀡𐀢𐀣𐀤𐀥𐀦𐀧𐀨𐀩𐀪𐀫𐀬𐀭𐀮𐀯𐀰𐀱𐀲𐀳𐀴𐀵𐀶𐀷𐀸𐀹𐀺𐀻𐀼𐀽𐀾𐀿𐁀𐁁𐁂𐁃𐁄𐁅𐁆𐁇𐁈𐁉𐁊𐁋𐁌𐁍𐁎𐁏𐁐𐁑𐁒𐁓𐁔𐁕𐁖𐁗𐁘𐁙𐁚𐁛𐁜𐁝𐁞𐁟𐁠𐁡𐁢𐁣𐁤𐁥𐁦𐁧𐁨𐁩𐁪𐁫𐁬𐁭𐁮𐁯𐁰𐁱𐁲𐁳𐁴𐁵𐁶𐁷𐁸𐁹𐁺𐁻𐁼𐁽𐁾𐁿𐂀𐂁𐂂𐂃𐂄𐂅𐂆𐂇𐂈𐂉𐂊𐂋𐂌𐂍𐂎𐂏𐂐𐂑𐂒𐂓𐂔𐂕𐂖𐂗𐂘𐂙𐂚𐂛𐂜𐂝𐂞𐂟𐂠𐂡𐂢𐂣𐂤𐂥𐂦𐂧𐂨𐂩𐂪𐂫𐂬𐂭𐂮𐂯𐂰𐂱𐂲𐂳𐂴𐂵𐂶𐂷𐂸𐂹𐂺𐂻𐂼𐂽𐂾𐂿𐃀𐃁𐃂𐃃𐃄𐃅𐃆𐃇𐃈𐃉𐃊𐃋𐃌𐃍𐃎𐃏𐃐𐃑𐃒𐃓𐃔𐃕𐃖𐃗𐃘𐃙𐃚𐃛𐃜𐃝𐃞𐃟𐃠𐃡𐃢𐃣𐃤𐃥𐃦𐃧𐃨𐃩𐃪𐃫𐃬𐃭𐃮𐃯𐃰𐃱𐃲𐃳𐃴𐃵𐃶𐃷𐃸𐃹𐃺𐃻𐃼𐃽𐃾𐃿𐄀𐄁𐄂𐄃𐄄𐄅𐄆𐄇𐄈𐄉𐄊𐄋𐄌𐄍𐄎𐄏𐄐𐄑𐄒𐄓𐄔𐄕𐄖𐄗𐄘𐄙𐄚𐄛𐄜𐄝𐄞𐄟𐄠𐄡𐄢𐄣𐄤𐄥𐄦𐄧𐄨𐄩𐄪𐄫𐄬𐄭𐄮𐄯𐄰𐄱𐄲𐄳𐄴𐄵𐄶𐄷𐄸𐄹𐄺𐄻𐄼𐄽𐄾𐄿𐅀𐅁𐅂𐅃𐅄𐅅𐅆𐅇𐅈𐅉𐅊𐅋𐅌𐅍𐅎𐅏𐅐𐅑𐅒𐅓𐅔𐅕𐅖𐅗𐅘𐅙𐅚𐅛𐅜𐅝𐅞𐅟𐅠𐅡𐅢𐅣𐅤𐅥𐅦𐅧𐅨𐅩𐅪𐅫𐅬𐅭𐅮𐅯𐅰𐅱𐅲𐅳𐅴𐅵𐅶𐅷𐅸𐅹𐅺𐅻𐅼𐅽𐅾𐅿𐆀𐆁𐆂𐆃𐆄𐆅𐆆𐆇𐆈𐆉𐆊𐆋𐆌𐆍𐆎𐆏𐆐𐆑𐆒𐆓𐆔𐆕𐆖𐆗𐆘𐆙𐆚𐆛𐆜𐆝𐆞𐆟𐆠𐆡𐆢𐆣𐆤𐆥𐆦𐆧𐆨𐆩𐆪𐆫𐆬𐆭𐆮𐆯𐆰𐆱𐆲𐆳𐆴𐆵𐆶𐆷𐆸𐆹𐆺𐆻𐆼𐆽𐆾𐆿𐇀𐇁𐇂𐇃𐇄𐇅𐇆𐇇𐇈𐇉𐇊𐇋𐇌𐇍𐇎𐇏𐇐𐇑𐇒𐇓𐇔𐇕𐇖𐇗𐇘𐇙𐇚𐇛𐇜𐇝𐇞𐇟𐇠𐇡𐇢𐇣𐇤𐇥𐇦𐇧𐇨𐇩𐇪𐇫𐇬𐇭𐇮𐇯𐇰𐇱𐇲𐇳𐇴𐇵𐇶𐇷𐇸𐇹𐇺𐇻𐇼𐇽𐇾𐇿𐈀𐈁𐈂𐈃𐈄𐈅𐈆𐈇𐈈𐈉𐈊𐈋𐈌𐈍𐈎𐈏𐈐𐈑𐈒𐈓𐈔𐈕𐈖𐈗𐈘𐈙𐈚𐈛𐈜𐈝𐈞𐈟𐈠𐈡𐈢𐈣𐈤𐈥𐈦𐈧𐈨𐈩𐈪𐈫𐈬𐈭𐈮𐈯𐈰𐈱𐈲𐈳𐈴𐈵𐈶𐈷𐈸𐈹𐈺𐈻𐈼𐈽𐈾𐈿𐉀𐉁𐉂𐉃𐉄𐉅𐉆𐉇𐉈𐉉𐉊𐉋𐉌𐉍𐉎𐉏𐉐𐉑𐉒𐉓𐉔𐉕𐉖𐉗𐉘𐉙𐉚𐉛𐉜𐉝𐉞𐉟𐉠𐉡𐉢𐉣𐉤𐉥𐉦𐉧𐉨𐉩𐉪𐉫𐉬𐉭𐉮𐉯𐉰𐉱𐉲𐉳𐉴𐉵𐉶𐉷𐉸𐉹𐉺𐉻𐉼𐉽𐉾𐉿𐊀𐊁𐊂𐊃𐊄𐊅𐊆𐊇𐊈𐊉𐊊𐊋𐊌𐊍𐊎𐊏𐊐𐊑𐊒𐊓𐊔𐊕𐊖𐊗𐊘𐊙𐊚𐊛𐊜𐊝𐊞𐊟𐊠𐊡𐊢𐊣𐊤𐊥𐊦𐊧𐊨𐊩𐊪𐊫𐊬𐊭𐊮𐊯𐊰𐊱𐊲𐊳𐊴𐊵𐊶𐊷𐊸𐊹𐊺𐊻𐊼𐊽𐊾𐊿𐋀𐋁𐋂𐋃𐋄𐋅𐋆𐋇𐋈𐋉𐋊𐋋𐋌𐋍𐋎𐋏𐋐𐋑𐋒𐋓𐋔𐋕𐋖𐋗𐋘𐋙𐋚𐋛𐋜𐋝𐋞𐋟𐋠𐋡𐋢𐋣𐋤𐋥𐋦𐋧𐋨𐋩𐋪𐋫𐋬𐋭𐋮𐋯𐋰𐋱𐋲𐋳𐋴𐋵𐋶𐋷𐋸𐋹𐋺𐋻𐋼𐋽𐋾𐋿𐌀𐌁𐌂𐌃𐌄𐌅𐌆𐌇𐌈𐌉𐌊𐌋𐌌𐌍𐌎𐌏𐌐𐌑𐌒𐌓𐌔𐌕𐌖𐌗𐌘𐌙𐌚𐌛𐌜𐌝𐌞𐌟𐌠𐌡𐌢𐌣𐌤𐌥𐌦𐌧𐌨𐌩𐌪𐌫𐌬𐌭𐌮𐌯𐌰𐌱𐌲𐌳𐌴𐌵𐌶𐌷𐌸𐌹𐌺𐌻𐌼𐌽𐌾𐌿𐍀𐍁𐍂𐍃𐍄𐍅𐍆𐍇𐍈𐍉𐍊𐍋𐍌𐍍𐍎𐍏𐍐𐍑𐍒𐍓𐍔𐍕𐍖𐍗𐍘𐍙𐍚𐍛𐍜𐍝𐍞𐍟𐍠𐍡𐍢𐍣𐍤𐍥𐍦𐍧𐍨𐍩𐍪𐍫𐍬𐍭𐍮𐍯𐍰𐍱𐍲𐍳𐍴𐍵𐍶𐍷𐍸𐍹𐍺𐍻𐍼𐍽𐍾𐍿𐎀𐎁𐎂𐎃𐎄𐎅𐎆𐎇𐎈𐎉𐎊𐎋𐎌𐎍𐎎𐎏𐎐𐎑𐎒𐎓𐎔𐎕𐎖𐎗𐎘𐎙𐎚𐎛𐎜𐎝𐎞𐎟𐎠𐎡𐎢𐎣𐎤𐎥𐎦𐎧𐎨𐎩𐎪𐎫𐎬𐎭𐎮𐎯𐎰𐎱𐎲𐎳𐎴𐎵𐎶𐎷𐎸𐎹𐎺𐎻𐎼𐎽𐎾𐎿𐏀𐏁𐏂𐏃𐏄𐏅𐏆𐏇𐏈𐏉𐏊𐏋𐏌𐏍𐏎𐏏𐏐𐏑𐏒𐏓𐏔𐏕𐏖𐏗𐏘𐏙𐏚𐏛𐏜𐏝𐏞𐏟𐏠𐏡𐏢𐏣𐏤𐏥𐏦𐏧𐏨𐏩𐏪𐏫𐏬𐏭𐏮𐏯𐏰𐏱𐏲𐏳𐏴𐏵𐏶𐏷𐏸𐏹𐏺𐏻𐏼𐏽𐏾𐏿𐐀𐐁𐐂𐐃𐐄𐐅𐐆𐐇𐐈𐐉𐐊𐐋𐐌𐐍𐐎𐐏𐐐𐐑𐐒𐐓𐐔𐐕𐐖𐐗𐐘𐐙𐐚𐐛𐐜𐐝𐐞𐐟𐐠𐐡𐐢𐐣𐐤𐐥𐐦𐐧𐐨𐐩𐐪𐐫𐐬𐐭𐐮𐐯𐐰𐐱𐐲𐐳𐐴𐐵𐐶𐐷𐐸𐐹𐐺𐐻𐐼𐐽𐐾𐐿𐑀𐑁𐑂𐑃𐑄𐑅𐑆𐑇𐑈𐑉𐑊𐑋𐑌𐑍𐑎𐑏𐑐𐑑𐑒𐑓𐑔𐑕𐑖𐑗𐑘𐑙𐑚𐑛𐑜𐑝𐑞𐑟𐑠𐑡𐑢𐑣𐑤𐑥𐑦𐑧𐑨𐑩𐑪𐑫𐑬𐑭𐑮𐑯𐑰𐑱𐑲𐑳𐑴𐑵𐑶𐑷𐑸𐑹𐑺𐑻𐑼𐑽𐑾𐑿𐒀𐒁𐒂𐒃𐒄𐒅𐒆𐒇𐒈𐒉𐒊𐒋𐒌𐒍𐒎𐒏𐒐𐒑𐒒𐒓𐒔𐒕𐒖𐒗𐒘𐒙𐒚𐒛𐒜𐒝𐒞𐒟𐒠𐒡𐒢𐒣𐒤𐒥𐒦𐒧𐒨𐒩𐒪𐒫𐒬𐒭𐒮𐒯𐒰𐒱𐒲𐒳𐒴𐒵𐒶𐒷𐒸𐒹𐒺𐒻𐒼𐒽𐒾𐒿𐓀𐓁𐓂𐓃𐓄𐓅𐓆𐓇𐓈𐓉𐓊𐓋𐓌𐓍𐓎𐓏𐓐𐓑𐓒𐓓𐓔𐓕𐓖𐓗𐓘𐓙𐓚𐓛𐓜𐓝𐓞𐓟𐓠𐓡𐓢𐓣𐓤𐓥𐓦𐓧𐓨𐓩𐓪𐓫𐓬𐓭𐓮𐓯𐓰𐓱𐓲𐓳𐓴𐓵𐓶𐓷𐓸𐓹𐓺𐓻𐓼𐓽𐓾𐓿𐔀𐔁𐔂𐔃𐔄𐔅𐔆𐔇𐔈𐔉𐔊𐔋𐔌𐔍𐔎𐔏𐔐𐔑𐔒𐔓𐔔𐔕𐔖𐔗𐔘𐔙𐔚𐔛𐔜𐔝𐔞𐔟𐔠𐔡𐔢𐔣𐔤𐔥𐔦𐔧𐔨𐔩𐔪𐔫𐔬𐔭𐔮𐔯𐔰𐔱𐔲𐔳𐔴𐔵𐔶𐔷𐔸𐔹𐔺𐔻𐔼𐔽𐔾𐔿𐕀𐕁𐕂𐕃𐕄𐕅𐕆𐕇𐕈𐕉𐕊𐕋𐕌𐕍𐕎𐕏𐕐𐕑𐕒𐕓𐕔𐕕𐕖𐕗𐕘𐕙𐕚𐕛𐕜𐕝𐕞𐕟𐕠𐕡𐕢𐕣𐕤𐕥𐕦𐕧𐕨𐕩𐕪𐕫𐕬𐕭𐕮𐕯𐕰𐕱𐕲𐕳𐕴𐕵𐕶𐕷𐕸𐕹𐕺𐕻𐕼𐕽𐕾𐕿𐖀𐖁𐖂𐖃𐖄𐖅𐖆𐖇𐖈𐖉𐖊𐖋𐖌𐖍𐖎𐖏𐖐𐖑𐖒𐖓𐖔𐖕𐖖𐖗𐖘𐖙𐖚𐖛𐖜𐖝𐖞𐖟𐖠𐖡𐖢𐖣𐖤𐖥𐖦𐖧𐖨𐖩𐖪𐖫𐖬𐖭𐖮𐖯𐖰𐖱𐖲𐖳𐖴𐖵𐖶𐖷𐖸𐖹𐖺𐖻𐖼𐖽𐖾𐖿𐗀𐗁𐗂𐗃𐗄𐗅𐗆𐗇𐗈𐗉𐗊𐗋𐗌𐗍𐗎𐗏𐗐𐗑𐗒𐗓𐗔𐗕𐗖𐗗𐗘𐗙𐗚𐗛𐗜𐗝𐗞𐗟𐗠𐗡𐗢𐗣𐗤𐗥𐗦𐗧𐗨𐗩𐗪𐗫𐗬𐗭𐗮𐗯𐗰𐗱𐗲𐗳𐗴𐗵𐗶𐗷𐗸𐗹𐗺𐗻𐗼𐗽𐗾𐗿𐘀𐘁𐘂𐘃𐘄𐘅𐘆𐘇𐘈𐘉𐘊𐘋𐘌𐘍𐘎𐘏𐘐𐘑𐘒𐘓𐘔𐘕𐘖𐘗𐘘𐘙𐘚𐘛𐘜𐘝𐘞𐘟𐘠𐘡𐘢𐘣𐘤𐘥𐘦𐘧𐘨𐘩𐘪𐘫𐘬𐘭𐘮𐘯𐘰𐘱𐘲𐘳𐘴𐘵𐘶𐘷𐘸𐘹𐘺𐘻𐘼𐘽𐘾𐘿𐙀𐙁𐙂𐙃𐙄𐙅𐙆𐙇𐙈𐙉𐙊𐙋𐙌𐙍𐙎𐙏𐙐𐙑𐙒𐙓𐙔𐙕𐙖𐙗𐙘𐙙𐙚𐙛𐙜𐙝𐙞𐙟𐙠𐙡𐙢𐙣𐙤𐙥𐙦𐙧𐙨𐙩𐙪𐙫𐙬𐙭𐙮𐙯𐙰𐙱𐙲𐙳𐙴𐙵𐙶𐙷𐙸𐙹𐙺𐙻𐙼𐙽𐙾𐙿𐚀𐚁𐚂𐚃𐚄𐚅𐚆𐚇𐚈𐚉𐚊𐚋𐚌𐚍𐚎𐚏𐚐𐚑𐚒𐚓𐚔𐚕𐚖𐚗𐚘𐚙𐚚𐚛𐚜𐚝𐚞𐚟𐚠𐚡𐚢𐚣𐚤𐚥𐚦𐚧𐚨𐚩𐚪𐚫𐚬𐚭𐚮𐚯𐚰𐚱𐚲𐚳𐚴𐚵𐚶𐚷𐚸𐚹𐚺𐚻𐚼𐚽𐚾𐚿𐛀𐛁𐛂𐛃𐛄𐛅𐛆𐛇𐛈𐛉𐛊𐛋𐛌𐛍𐛎𐛏𐛐𐛑𐛒𐛓𐛔𐛕𐛖𐛗𐛘𐛙𐛚𐛛𐛜𐛝𐛞𐛟𐛠𐛡𐛢𐛣𐛤𐛥𐛦𐛧𐛨𐛩𐛪𐛫𐛬𐛭𐛮𐛯𐛰𐛱𐛲𐛳𐛴𐛵𐛶𐛷𐛸𐛹𐛺𐛻𐛼𐛽𐛾𐛿𐜀𐜁𐜂𐜃𐜄𐜅𐜆𐜇𐜈𐜉𐜊𐜋𐜌𐜍𐜎𐜏𐜐𐜑𐜒𐜓𐜔𐜕𐜖𐜗𐜘𐜙𐜚𐜛𐜜𐜝𐜞𐜟𐜠𐜡𐜢𐜣𐜤𐜥𐜦𐜧𐜨𐜩𐜪𐜫𐜬𐜭𐜮𐜯𐜰𐜱𐜲𐜳𐜴𐜵𐜶𐜷𐜸𐜹𐜺𐜻𐜼𐜽𐜾𐜿𐝀𐝁𐝂𐝃𐝄𐝅𐝆𐝇𐝈𐝉𐝊𐝋𐝌𐝍𐝎𐝏𐝐𐝑𐝒𐝓𐝔𐝕𐝖𐝗𐝘𐝙𐝚𐝛𐝜𐝝𐝞𐝟𐝠𐝡𐝢𐝣𐝤𐝥𐝦𐝧𐝨𐝩𐝪𐝫𐝬𐝭𐝮𐝯𐝰𐝱𐝲𐝳𐝴𐝵𐝶𐝷𐝸𐝹𐝺𐝻𐝼𐝽𐝾𐝿𐞀𐞁𐞂𐞃𐞄𐞅𐞆𐞇𐞈𐞉𐞊𐞋𐞌𐞍𐞎𐞏𐞐𐞑𐞒𐞓𐞔𐞕𐞖𐞗𐞘𐞙𐞚𐞛𐞜𐞝𐞞𐞟𐞠𐞡𐞢𐞣𐞤𐞥𐞦𐞧𐞨𐞩𐞪𐞫𐞬𐞭𐞮𐞯𐞰𐞱𐞲𐞳𐞴𐞵𐞶𐞷𐞸𐞹𐞺𐞻𐞼𐞽𐞾𐞿𐟀𐟁𐟂𐟃𐟄𐟅𐟆𐟇𐟈𐟉𐟊𐟋𐟌𐟍𐟎𐟏𐟐𐟑𐟒𐟓𐟔𐟕𐟖𐟗𐟘𐟙𐟚𐟛𐟜𐟝𐟞𐟟𐟠𐟡𐟢𐟣𐟤𐟥𐟦𐟧𐟨𐟩𐟪𐟫𐟬𐟭𐟮𐟯𐟰𐟱𐟲𐟳𐟴𐟵𐟶𐟷𐟸𐟹𐟺𐟻𐟼𐟽𐟾𐟿𐠀𐠁𐠂𐠃𐠄𐠅𐠆𐠇𐠈𐠉𐠊𐠋𐠌𐠍𐠎𐠏𐠐𐠑𐠒𐠓𐠔𐠕𐠖𐠗𐠘𐠙𐠚𐠛𐠜𐠝𐠞𐠟𐠠𐠡𐠢𐠣𐠤𐠥𐠦𐠧𐠨𐠩𐠪𐠫𐠬𐠭𐠮𐠯𐠰𐠱𐠲𐠳𐠴𐠵𐠶𐠷𐠸𐠹𐠺𐠻𐠼𐠽𐠾𐠿𐡀𐡁𐡂𐡃𐡄𐡅𐡆𐡇𐡈𐡉𐡊𐡋𐡌𐡍𐡎𐡏𐡐𐡑𐡒𐡓𐡔𐡕𐡖𐡗𐡘𐡙𐡚𐡛𐡜𐡝𐡞𐡟𐡠𐡡𐡢𐡣𐡤𐡥𐡦𐡧𐡨𐡩𐡪𐡫𐡬𐡭𐡮𐡯𐡰𐡱𐡲𐡳𐡴𐡵𐡶𐡷𐡸𐡹𐡺𐡻𐡼𐡽𐡾𐡿𐢀𐢁𐢂𐢃𐢄𐢅𐢆𐢇𐢈𐢉𐢊𐢋𐢌𐢍𐢎𐢏𐢐𐢑𐢒𐢓𐢔𐢕𐢖𐢗𐢘𐢙𐢚𐢛𐢜𐢝𐢞𐢟𐢠𐢡𐢢𐢣𐢤𐢥𐢦𐢧𐢨𐢩𐢪𐢫𐢬𐢭𐢮𐢯𐢰𐢱𐢲𐢳𐢴𐢵𐢶𐢷𐢸𐢹𐢺𐢻𐢼𐢽𐢾𐢿𐣀𐣁𐣂𐣃𐣄𐣅𐣆𐣇𐣈𐣉𐣊𐣋𐣌𐣍𐣎𐣏𐣐𐣑𐣒𐣓𐣔𐣕𐣖𐣗𐣘𐣙𐣚𐣛𐣜𐣝𐣞𐣟𐣠𐣡𐣢𐣣𐣤𐣥𐣦𐣧𐣨𐣩𐣪𐣫𐣬𐣭𐣮𐣯𐣰𐣱𐣲𐣳𐣴𐣵𐣶𐣷𐣸𐣹𐣺𐣻𐣼𐣽𐣾𐣿𐤀𐤁𐤂𐤃𐤄𐤅𐤆𐤇𐤈𐤉𐤊𐤋𐤌𐤍𐤎𐤏𐤐𐤑𐤒𐤓𐤔𐤕𐤖𐤗𐤘𐤙𐤚𐤛𐤜𐤝𐤞𐤟𐤠𐤡𐤢𐤣𐤤𐤥𐤦𐤧𐤨𐤩𐤪𐤫𐤬𐤭𐤮𐤯𐤰𐤱𐤲𐤳𐤴𐤵𐤶𐤷𐤸𐤹𐤺𐤻𐤼𐤽𐤾𐤿𐥀𐥁𐥂𐥃𐥄𐥅𐥆𐥇𐥈𐥉𐥊𐥋𐥌𐥍𐥎𐥏𐥐𐥑𐥒𐥓𐥔𐥕𐥖𐥗𐥘𐥙𐥚𐥛𐥜𐥝𐥞𐥟𐥠𐥡𐥢𐥣𐥤𐥥𐥦𐥧𐥨𐥩𐥪𐥫𐥬𐥭𐥮𐥯𐥰𐥱𐥲𐥳𐥴𐥵𐥶𐥷𐥸𐥹𐥺𐥻𐥼𐥽𐥾𐥿𐦀𐦁𐦂𐦃𐦄𐦅𐦆𐦇𐦈𐦉𐦊𐦋𐦌𐦍𐦎𐦏𐦐𐦑𐦒𐦓𐦔𐦕𐦖𐦗𐦘𐦙𐦚𐦛𐦜𐦝𐦞𐦟𐦠𐦡𐦢𐦣𐦤𐦥𐦦𐦧𐦨𐦩𐦪𐦫𐦬𐦭𐦮𐦯𐦰𐦱𐦲𐦳𐦴𐦵𐦶𐦷𐦸𐦹𐦺𐦻𐦼𐦽𐦾𐦿𐧀𐧁𐧂𐧃𐧄𐧅𐧆𐧇𐧈𐧉𐧊𐧋𐧌𐧍𐧎𐧏𐧐𐧑𐧒𐧓𐧔𐧕𐧖𐧗𐧘𐧙𐧚𐧛𐧜𐧝𐧞𐧟𐧠𐧡𐧢𐧣𐧤𐧥𐧦𐧧𐧨𐧩𐧪𐧫𐧬𐧭𐧮𐧯𐧰𐧱𐧲𐧳𐧴𐧵𐧶𐧷𐧸𐧹𐧺𐧻𐧼𐧽𐧾𐧿𐨀𐨁𐨂𐨃𐨄𐨅𐨆𐨇𐨈𐨉𐨊𐨋𐨌𐨍𐨎𐨏𐨐𐨑𐨒𐨓𐨔𐨕𐨖𐨗𐨘𐨙𐨚𐨛𐨜𐨝𐨞𐨟𐨠𐨡𐨢𐨣𐨤𐨥𐨦𐨧𐨨𐨩𐨪𐨫𐨬𐨭𐨮𐨯𐨰𐨱𐨲𐨳𐨴𐨵𐨶𐨷𐨹𐨺𐨸𐨻𐨼𐨽𐨾𐨿𐩀𐩁𐩂𐩃𐩄𐩅𐩆𐩇𐩈𐩉𐩊𐩋𐩌𐩍𐩎𐩏𐩐𐩑𐩒𐩓𐩔𐩕𐩖𐩗𐩘𐩙𐩚𐩛𐩜𐩝𐩞𐩟𐩠𐩡𐩢𐩣𐩤𐩥𐩦𐩧𐩨𐩩𐩪𐩫𐩬𐩭𐩮𐩯𐩰𐩱𐩲𐩳𐩴𐩵𐩶𐩷𐩸𐩹𐩺𐩻𐩼𐩽𐩾𐩿𐪀𐪁𐪂𐪃𐪄𐪅𐪆𐪇𐪈𐪉𐪊𐪋𐪌𐪍𐪎𐪏𐪐𐪑𐪒𐪓𐪔𐪕𐪖𐪗𐪘𐪙𐪚𐪛𐪜𐪝𐪞𐪟𐪠𐪡𐪢𐪣𐪤𐪥𐪦𐪧𐪨𐪩𐪪𐪫𐪬𐪭𐪮𐪯𐪰𐪱𐪲𐪳𐪴𐪵𐪶𐪷𐪸𐪹𐪺𐪻𐪼𐪽𐪾𐪿𐫀𐫁𐫂𐫃𐫄𐫅𐫆𐫇𐫈𐫉𐫊𐫋𐫌𐫍𐫎𐫏𐫐𐫑𐫒𐫓𐫔𐫕𐫖𐫗𐫘𐫙𐫚𐫛𐫜𐫝𐫞𐫟𐫠𐫡𐫢𐫣𐫤𐫦𐫥𐫧𐫨𐫩𐫪𐫫𐫬𐫭𐫮𐫯𐫰𐫱𐫲𐫳𐫴𐫵𐫶𐫷𐫸𐫹𐫺𐫻𐫼𐫽𐫾𐫿𐬀𐬁𐬂𐬃𐬄𐬅𐬆𐬇𐬈𐬉𐬊𐬋𐬌𐬍𐬎𐬏𐬐𐬑𐬒𐬓𐬔𐬕𐬖𐬗𐬘𐬙𐬚𐬛𐬜𐬝𐬞𐬟𐬠𐬡𐬢𐬣𐬤𐬥𐬦𐬧𐬨𐬩𐬪𐬫𐬬𐬭𐬮𐬯𐬰𐬱𐬲𐬳𐬴𐬵𐬶𐬷𐬸𐬹𐬺𐬻𐬼𐬽𐬾𐬿𐭀𐭁𐭂𐭃𐭄𐭅𐭆𐭇𐭈𐭉𐭊𐭋𐭌𐭍𐭎𐭏𐭐𐭑𐭒𐭓𐭔𐭕𐭖𐭗𐭘𐭙𐭚𐭛𐭜𐭝𐭞𐭟𐭠𐭡𐭢𐭣𐭤𐭥𐭦𐭧𐭨𐭩𐭪𐭫𐭬𐭭𐭮𐭯𐭰𐭱𐭲𐭳𐭴𐭵𐭶𐭷𐭸𐭹𐭺𐭻𐭼𐭽𐭾𐭿𐮀𐮁𐮂𐮃𐮄𐮅𐮆𐮇𐮈𐮉𐮊𐮋𐮌𐮍𐮎𐮏𐮐𐮑𐮒𐮓𐮔𐮕𐮖𐮗𐮘𐮙𐮚𐮛𐮜𐮝𐮞𐮟𐮠𐮡𐮢𐮣𐮤𐮥𐮦𐮧𐮨𐮩𐮪𐮫𐮬𐮭𐮮𐮯𐮰𐮱𐮲𐮳𐮴𐮵𐮶𐮷𐮸𐮹𐮺𐮻𐮼𐮽𐮾𐮿𐯀𐯁𐯂𐯃𐯄𐯅𐯆𐯇𐯈𐯉𐯊𐯋𐯌𐯍𐯎𐯏𐯐𐯑𐯒𐯓𐯔𐯕𐯖𐯗𐯘𐯙𐯚𐯛𐯜𐯝𐯞𐯟𐯠𐯡𐯢𐯣𐯤𐯥𐯦𐯧𐯨𐯩𐯪𐯫𐯬𐯭𐯮𐯯𐯰𐯱𐯲𐯳𐯴𐯵𐯶𐯷𐯸𐯹𐯺𐯻𐯼𐯽𐯾𐯿𐰀𐰁𐰂𐰃𐰄𐰅𐰆𐰇𐰈𐰉𐰊𐰋𐰌𐰍

- **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 "認證"
10. 認證
11. 認證 Stripe 認證

認證 6 認證

1. 認證
2. 認證
3. 認證 認證
4. 認證
 - 認證 ID 認證

- 0000000000
 - 0000000000
 - 0000000000
5. 0000000000000000
 6. 0000000 PDF0000000000 "00 **PDF**" 00
 7. 00000000000000000000 "0000000" 00
 8. 0000000000

0000 **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

0000 **8**0000000000

1. 00000000000000
2. 000000 00 0000
3. 0000000000
4. 0000000000000000
5. 0000000000
6. 0000000000 "0000" 00
7. 00 "00" 00000
8. 000000 "0000 2025-01-15"

9. `return` " "
10. `return` " "
11. `return`
12. `return` " "

9

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

-

-
-

PDF

- WeasyPrint
- HTML API

Stripe

- Stripe API
- Stripe

データベース

- データベースの設計
- データベースの最適化 **Stripe** データベースの最適化

メール配信

- データ Mailjet API の連携
- データ `crm_config.yaml` の Mailjet の連携

決済

- データ オンライン/POS/決済の連携 **Stripe** の連携
- データ オンライン **Stripe** の連携

PDF 生成

- データ PDF の生成
- データ データ PDF の削除 `DELETE FROM Invoice_PDF_Cache WHERE invoice_id = X;`

レポート

- データ レポートの生成
- データ レポートの最適化

セキュリティ

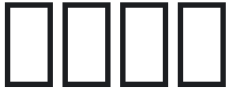
- データ セキュリティの対策
- データ セキュリティの検証

パフォーマンス

- データ パフォーマンスの最適化
- データ パフォーマンスの検証

デバッグ

- データ デバッグの手法
- データ デバッグのツール



□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□”□□□□

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

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□

□□□

□□□

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

□□□□

□□□□□□□□□□□□□□

□□□□□□

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

□□□□□□□□

1. 1000 00 → 100000000
2. 10000000000

1000000000

10000000000000000000000

1000000

- 1000 - 10000000000000000000
- 1000000 - 1000000000000000
- 100000 - 1000000000000000
- 1000000 - 1000000000000000

1000000

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

100000

100000000000000000000

□□□□

- **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** - 000ID00000/000000
- 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

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 \$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 (-)

00000000

- 00000000
- 00000000000000000000
- 00ID0000000000000000
- 0000000000 (✓)
- 000000

000000

0000000000000000

1. 00000000000000000000
2. 000000000000
3. 000000000000
4. 0000 invoice_id 000000
5. 0000000000"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. 0000000000
3. 0000000000000000 : 0
4. 00 "0000"
5. 0000000000
6. 0000000000
7. 000000000000

□□□□

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

Playbooks

Ansible playbooks OmniCRM

API API API

OmniCRM Ansible Playbooks CRM API

1. -
2. -

Playbooks

retail_cost wholesale_cost retail_setup_cost
playbook

Ansible playbook

- api_response_product.json.retail_cost
- -
- -
- -
- -
- -
- -
- -

□□□□□□□□□□

```
# □□ retail_cost □ $99.00□□□□□□ $79.00 □□□□□□
- name: □□□□□□□□□□□□□□□□ retail_cost□
  set_fact:
    charge_amount: 79.00

- name: □□□□□□□
  uri:
    url: "http://localhost:5000/crm/payments/authorize/hold"
    method: POST
    body_format: json
    body:
      customer_id: "{{ customer_id | int }}"
      amount: "{{ charge_amount | float }}" # □□□□□□□□□□□□
retail_cost
# ...
```

□□□□□□□□□□

```
# □□□□□□□□□□□□
- name: □□□□□□□□
  uri:
    url: "http://localhost:5000/crm/customer/{{ customer_id }}"
    # ...
  register: customer_info

- name: □□□□□□□□
  set_fact:
    base_price: "{{ api_response_product.json.retail_cost | float
  }}"
    discount_percent: >-
      {% if customer_info.json.tier == 'gold' %}20
      {% elif customer_info.json.tier == 'silver' %}10
      {% else %}0{% endif %}

- name: □□□□□□□□□□□□
  set_fact:
    final_price: "{{ (base_price | float) * (1 - (discount_percent
  | float / 100)) | round(2) }}"
```

ansible-playbook

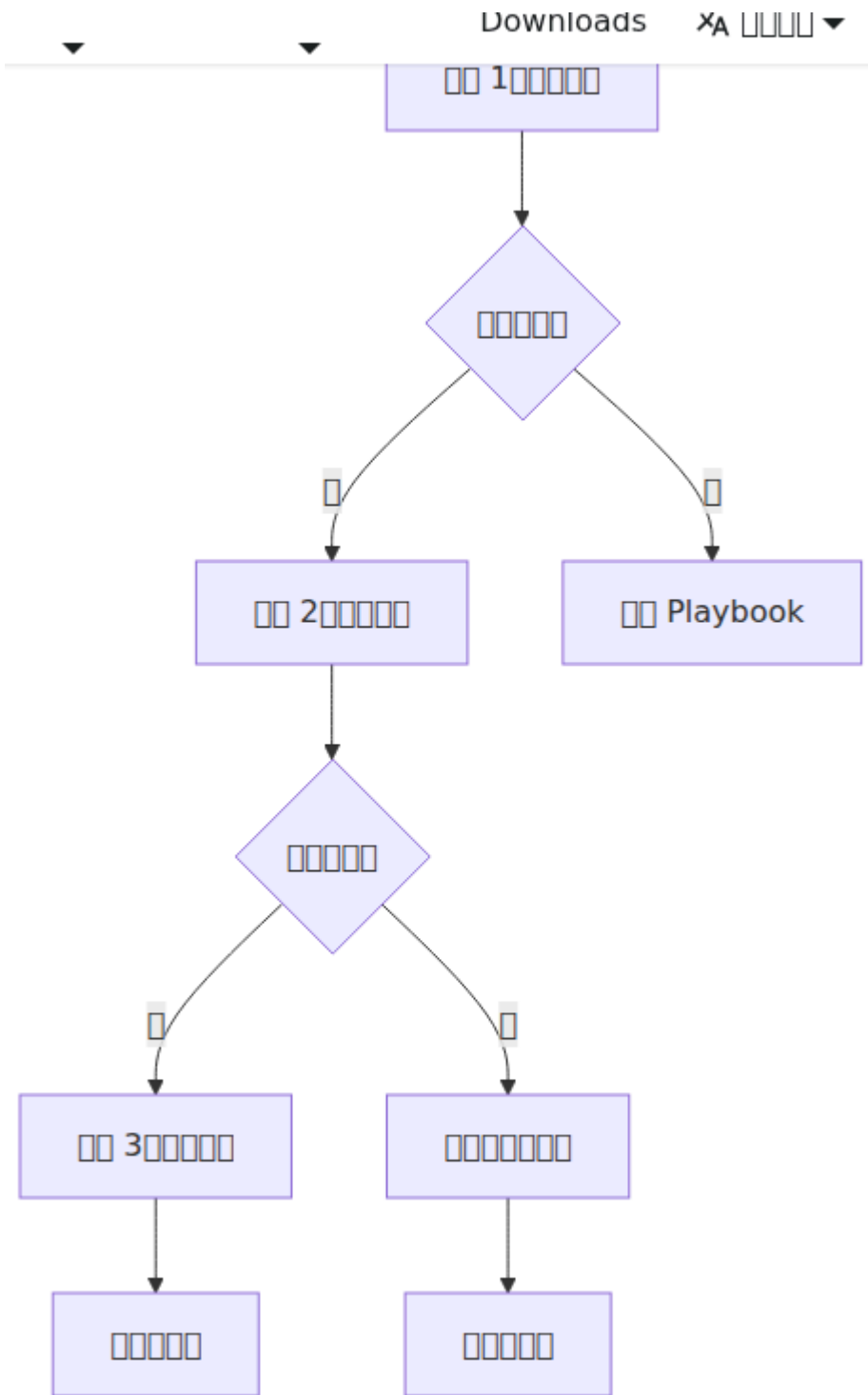
```
# 例として wholesale_cost と retail_cost
- name: 例として
  set_fact:
    charge_amount: >-
      {% if customer_info.json.is_wholesale_partner %}
      {{ api_response_product.json.wholesale_cost | float }}
      {% else %}
      {{ api_response_product.json.retail_cost | float }}
      {% endif %}
```

ansible-playbook 例としてCRM 例として/例として
ansible-playbooks 例として

例として ? ? ? 例として

例として

□□□□



□□□□

□□□□□□□□□□

□□ **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 }}"

# 支払い方法
- name: 支払い方法
  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: 支払い方法 ID を取得
  set_fact:
    payment_method_id: "{{ api_response_payment_methods.json |
json_query(query) }}"
  vars:
    query: "data[?is_default==`true`].payment_method_id | [0]"

# =====
# 支払い方法
# =====

- name: "1回払いの支払い方法を作成"
  uri:
    url: "http://localhost:5000/crm/payments/authorize/hold"
    method: POST
    headers:
      Content-Type: "application/json"
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "customer_id": "{{ customer_id | int }}",
        "amount": "{{ monthly_cost | float }}",
        "currency": "{{ crm_config.currency | default('AUD')
}}",
        "payment_method_id": "{{ payment_method_id | int }}",
        "metadata": {
          "description": "{{ package_name }} on {{
api_response_service.json.service_name }}"

```

```

        "service_id": "{{
api_response_service.json.service_id | int }}",
        "site_id": "{{ api_response_service.json.site_id |
int }}",
        "product_id": "{{ product_id }}",
        "user_id": "{{ (initiating_user | int) if
(initiating_user is defined and initiating_user is not none) else
omit }}",
        "title": "{{ package_name }}",
        "wholesale_cost": "{{ wholesale_cost | float }}",
        "invoice": true,
        "contract_days": "{{ days | int }}",
        "send_email": true
    }
}
return_content: yes
register: api_response_authorization

- name: []
  assert:
    that:
      - api_response_authorization.status == 200
      - api_response_authorization.json.success == true

- name: [] authorization_id []/[]
  set_fact:
    authorization_id: "{{
api_response_authorization.json.data.authorization_id }}"

# [] 200CS []
- block:
  - name: "[] 2000 CGRateS []"
    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 {{ authorization_id }}"

```

API

\$1 \$10
 \$9

1. /

POST /crm/payments/authorize/hold

\$150 \$500 \$350

```
{
  "customer_id": 123,
  "amount": 49.99,
  "currency": "AUD",
  "payment_method_id": 456,
  "metadata": {
    "description": "商品",
    "service_id": 789,
    "site_id": 12,
    "product_id": "34",
    "user_id": 5,
    "title": "商品",
    "wholesale_cost": 25.00,
    "invoice": true,
    "contract_days": 30,
    "send_email": true
  }
}
```

商品

- **customer_id**商品 - 商品 ID
- **amount**商品 - 商品
- **currency**商品 - 商品
- **payment_method_id**商品 - 商品
- **metadata**商品 - 商品
 - **description** - 商品
 - **service_id** - 商品
 - **site_id** - 商品
 - **product_id** - 商品
 - **user_id** - 商品
 - **title** - 商品
 - **wholesale_cost** - 商品
 - **invoice** - 商品 `true`商品
 - **contract_days** - 商品
 - **send_email** - 商品 `true`商品

000

```
{
  "success": true,
  "message": "000000000000",
  "data": {
    "authorization_id": 301,
    "vendor_authorization_id": "auth_xxxxx",
    "amount": 500.00,
    "currency": "USD",
    "status": "authorized",
    "wallet_balance": 150.00,
    "wallet_to_use": 150.00,
    "card_amount": 350.00,
    "message": "0000 $3500000000000000 $500 000000"
  }
}
```

000

- 00 authorization_id 000000000000
- 00 card_amount 0000000000
- 0000000000000000000000

2. 0000

000 POST /crm/payments/capture/{authorization_id}

00000000000000000000000000000000 invoice: true 0000000000


0000

```
{
  "metadata": {
    "provisioning_status": "success",
    "cgr_action": "Action_Topup_Standard",
    "additional_info": "00000000"
  }
}
```

□□□

```
{
  "success": true,
  "data": {
    "payment_id": "pay_xyz789",
    "transaction_id": 1234
  }
}
```

□□□□□□

1. □□□□□□□□□□□□□□□□
2. **□□□**□□□□□□
3. □□□□□□□□□□
4. □□□□□□□□□□ `invoice: true`
 - □□□□□□□□□□ = □□□
 - □□□□□□□□□□□□
 - □□□□□□□□□□ = □□□□□□
 - □□□□□□□□□□□□□□□□□□□□□□
 - □□□□□□□□□□□
5. □□□□□□□□□□□
6. □□ `send_email: true`□□□□□□□□□□□□□□□□

□□□\$500 □□□ \$150 □□□□□□

- □□□□\$350
- □□□□□+\$350□□□□□□□ \$500□
- □□□□□-\$500□□□□□□□
- □□□□□\$0

□□□□□□□□□□ □□□□□□□□□□

3. □□□□□□□

□□□ `POST /crm/payments/release/{authorization_id}`

API 返回的响应体示例如下：

成功

```
{
  "metadata": {
    "release_reason": "provisioning_failed",
    "error_details": "OCS 失败"
  }
}
```

失败

```
{
  "success": true,
  "message": "成功"
}
```

注意事项

1. 确保设备已正确配置
2. 检查网络连接是否正常
3. 确认设备固件版本
4. 检查设备状态
5. 检查设备日志

如需进一步帮助，请参考 API 文档。

API 接口

通过 API 调用设备接口，API 接口如下：

API 接口

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
    }
  }
}
```

□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```

- 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


```

- name: 0000
  uri: ...
  register: api_response_authorization

- name: 0000000
  assert:
    that:
      - api_response_authorization.status == 200
      - api_response_authorization.json.success == true
    fail_msg: "00000000{{ api_response_authorization.json }}"

```

3. 0000 ID

0000 authorization_id 0000/00000

```

- name: 00 authorization_id 0000/00
  set_fact:
    authorization_id: "{{
api_response_authorization.json.data.authorization_id }}"

```

4. 000000000

00000000000000000000

```

metadata:
  description: "00000000000"
  service_id: "{{ service_id | int }}"
  product_id: "{{ product_id }}"
  user_id: "{{ initiating_user | int }}"
  title: "00000000"
  wholesale_cost: "{{ wholesale_cost | float }}"
  invoice: true # 000000000000
  send_email: true # 000000

```

5. 000000000

0000000000000000000000

```
- name: 0000000000
  set_fact:
    final_cost: "{{ (base_cost | float * multiplier | float) |
round(2) }}"
```

6. 0000000000

0000000000000000000000000000000000

```
- name: 0000000000
  set_fact:
    payment_method_id: "{{ api_response_payment_methods.json |
json_query(query) }}"
  vars:
    query: "data[?is_default==`true`].payment_method_id | [0]"

- name: 0000000000
  assert:
    that:
      - payment_method_id is defined
      - payment_method_id != ""
      - payment_method_id != None
    fail_msg: "000000 {{ customer_id }} 00000000"
```

00000

00 **1**0000000000

0000000000000000

1. 0000000000
2. 0000
3. 0 OCS/CGRateS 000000
4. 0 CRM 00000000
5. 0000
6. 00000000000000

000 play_topup_charge_then_action.yaml 0000000

00 200000000000

0000000000000000

1. 0000
2. 000000
3. 00000000000000000000
4. 00000000000000

000 play_simple_service.yaml 00 202-232 00000000

00 300000/0000

00000000000000

1. 000000
2. 00 CGRateS 00
3. 000000
4. 0000000000

00 4000 **ActionPlan** 000000

0000000000

1. 0000 *http_post 0000000000
2. 0000 *monthly 000 ActionPlan
3. 0 ActionPlan 00000
4. CGRateS 0000000000
5. 00 playbook 000000

□□□□

□□□□

□□□ □□□□□□□□

□□□□

- □□□□□□□□□□
- □□□□
- □□□□□□
- □□ ID □□□

□□□□□ □□□□□□□□□□□□□□

□□□□□□□□□□□□

□□□ □□□□□□□□□□□□

□□□ □□□□□□□□□□□ - □□□□□□□□□□□□□□

□□□□□

- □□□□□□□□□□□ 7 □□
- □□□□□□□□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□

□□□□□□□□□□

□□□ □□□□□□□□□□□□

□□□ □□□□□□□□□□ `invoice: true`

□□□□□ □□□



- □□□□□□□□□□ `invoice: true`□□□

- 0000000000000000

0000000000

000 0000000000000000

00000

- 0000000000000000/0000/00000
- 00000   0000000000
- 000000

00000

- 0000000000000000000000000000
- 0000000000000000
- 0000000000
- 000000000000

0000000000

0000

0000000000000000

1. 00000000 - 00000000/PayPal

```

- name: 請求
  uri:
    url: "http://localhost:5000/crm/payments/refund"
    method: POST
    headers:
      Content-Type: "application/json"
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "transaction_id": "{{ vendor_transaction_id }}",
        "vendor": "stripe",
        "amount": "{{ refund_amount | float }}",
        "reason": "customer_request"
      }

```

2. 請求 - 請求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` - □□□□□□□□□□

OmniCRM RBAC

OmniCRM RBAC

OmniCRM RBAC RBAC view_customer create_inventory

RBAC

RBAC

1. RBAC
2. RBAC
3. RBAC
4. RBAC – “RBAC...”

RBAC

- RBAC – OmniCRM RBAC
- RBAC – view_customer delete_product

-   — `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

○○○○○○○○○○○○○○○○○○

○○○○○○○○Web UI/API○○○○○○○○○○

○○Web UI

1. ○○○○○○○○ - ○○○○○○○○○○○○○○○○○
2. ○○“○○○○” - ○○○○○○○○○

3. ○○○○○○○○○○
 - ○○○ - ○○○○○○○○○○○○○○○
 - ○○○○ - ○○○○○○○○○○○○○○○○○○○
 - ○○ - ○○○○○○○○○○○○○○○○○○○○○
 - ○○ - ○○○○○○○○○○○
 - ○○○ - ○○○○○○○○○○○
 - ○○ - ○○○○○○○○○○○
 - ○○○○ - ○○○○○○○○○
 - ○○ - ○○○○○○○○○○○○○○○○○○○
 - ○○○○○ - ○○○○○○○○○○○○○○○○○○○○○○○

4. ○○“○○○○” - ○○○○○○○○○○○○○○○○○○○

5. ○○○○○○ - ○○○○○○○○○○○○○○○○○○○

○○○○○

- 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 Endpoints

GET /auth/users/{user_id}

Response

- user_id
- username
- email
- password
- role

Request

Request parameters: user_id (required)

Response

Web UI

1. User ID → user_id
2. Username: **username**
3. Email: email
 - user - email: "Tier2_Support"
 - user - email: email
4. Password: **password**
5. Role: role

API

POST /auth/roles

Request body: admin

Response

API

POST /auth/roles/{role_id}/permissions

```
{
  "permission_id": 123
}
```

□□□□□□□□

```
{
  "permission_ids": [123, 124, 125]
}
```

□□□□□“□□□□”□□

□□□□□□□□□□□□□□□□□□

1. □□□□□

```
POST /auth/roles
{
  "name": "Provisioning_Specialist",
  "description": "Can provision services and manage customer
services"
}
```

2. □□□□□

```
POST /auth/roles/45/permissions
{
  "permission_ids": [
    1, # view_customer
    20, # view_customer_service
    21, # create_customer_service
    22, # update_customer_service
    30, # view_provision
    31, # create_provision
    40, # view_inventory
    50, # view_product
  ]
}
```

□□□□□□□□

□□Web UI□

1. □□□□□□□□
2. □□□□□□□□□□□□□□□□**"X"□"□□"□□**□□
3. □□□□

□□API□

□□□ DELETE /auth/roles/{role_id}/permissions/{permission_id}

□□□

```
DELETE /auth/roles/45/permissions/31
```

□□□□□□□□ create_provision □□□

□□□□□□□□

□□□□□□□□

□□Web UI□

1. □□□□□□□□ → □□□□□□
2. □□□□□□□□
3. □□□□□□□□
4. □□**"□□"□□**

API

PUT /auth/roles/{role_id}

```
{  
  "name": "Senior_Support",  
  "description": "Senior support team with full customer access"  
}
```

API

DELETE /auth/roles/{role_id}

データベースに「role」を追加

データベース

Web UI

1. ログイン → ログイン
2. ログイン
3. ログイン/パスワード
4. **"role"*

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!"  
}
```

Web UI

Admin Panel

Web UI

1. Admin Panel → Admin
2. Admin Panel**"Admin"**
3. Admin Panel
4. Admin**"Admin"**
5. Admin Panel
6. Admin Panel

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. Settings → Security
2. Enable 2FA
3. Select authentication method: **"2FA"**.
 - TOTP (QR)
 - 10-digit code
5. Select authenticator: Google Authenticator / Authy (QR)
6. Scan QR code
7. Enter code
8. 2FA is enabled

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. 存储验证状态

□□□□

- 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 [redacted] - [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

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

权限名称

- `admin` — 系统最高权限
- `can_impersonate` — 权限名称/权限名称
- `grafana_access` — 访问Grafana权限名称
Grafana权限名称

权限名称

- 权限
 - view_customer[] create_customer[] update_customer[] delete_customer
 - 权限名称 权限名称
- 权限
 - view_customer_site[] create_customer_site[]
update_customer_site[] delete_customer_site
 - 权限名称 权限名称
- 权限

- view_customer_contact[] create_customer_contact[]
update_customer_contact[] delete_customer_contact
- [] [] [] [] [] [] [] [] [] []
- [] [] [] [] [] [] [] [] [] [] <administration_attributes> []
 - view_customer_attribute[] create_customer_attribute[]
update_customer_attribute[] delete_customer_attribute
 - [] [] [] [] [] [] [] [] [] []
- [] [] [] [] [] [] [] [] [] [] <administration_tags> []
 - view_customer_tag[] create_customer_tag[] update_customer_tag[]
delete_customer_tag
 - [] [] [] [] [] [] [] [] [] []
- [] [] [] []
 - view_customer_service[] create_customer_service[]
update_customer_service[] delete_customer_service
 - [] [] [] [] [] [] [] [] [] []
- [] [] [] []
 - view_customer_activity[] create_customer_activity[]
update_customer_activity[] delete_customer_activity
 - [] [] [] [] [] [] [] [] [] []



- **Stripe** []
 - view_customer_stripe_card[] create_customer_stripe_card[]
update_customer_stripe_card[] delete_customer_stripe_card
 - [] [] [] [] [] [] [] [] [] [] Stripe []
- [] []
 - view_customer_transaction[] create_customer_transaction[]
update_customer_transaction[] delete_customer_transaction
 - [] [] [] [] [] [] [] [] [] []
- [] []
 - view_customer_invoice[] create_customer_invoice[]
update_customer_invoice[] delete_customer_invoice
 - [] [] [] [] [] [] [] [] [] []

□□

- 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。角色权限是指将用户分为不同的角色，每个角色拥有不同的权限。细粒度权限是指将权限细化到具体的操作或数据上。

权限管理

角色	权限	备注
管理员 角色	admin 角色 can_impersonate 角色 CRUD 操作	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。	权限管理
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 "impersonate..." if impersonation is successful
- Requires impersonation permissions — admin
- Requires `admin` and `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

- 0000000000000000
- 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. 保存

□□

□□□□□□□□□□□□□□□□□□□□

□□□□□

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

□□□□□

□□□□□□□□□□□□□□□□□□□□

□□□□□□□

- □□□ - □□Stripe□□□□□□□□□□/□□□
- □□□□ - □□□□□□□□□
- □□□ - □□□□□□□□□□
- □□□□□ - □□□□4□□□□□□□□□□□

□□□ □□□□ <payment_methods> □□□□□□□□□□□□□□□□

□□ / □□

□□□□□□□□□□□□□□□□□□□□

□□□□□

1. □□□ □□ → □□□□ → “□□□”
2. □□□□□□□□□□

3. 0000000000000000
4. 0000000000
5. 0000000
6. 0000
7. 0000000

000000

- 0000 - 1GB 5GB 10GB 20GB 50GB 00
- 0000 - 0000000
- 000 - 000
- 0000 - 000000 £5 £10 £20 £50 £100

000 000000 <features_topup_recharge> 0000000000000000

00000000

00000000000000000000

00000000

- 0000 - 0000000
- 00**IP**00 - 0000000000IP
- 0000 - IPTV0000000000
- 0000 - 00000000000000
- 0000 - 00000000

00000000

1. 000 00 → 0000 → “0000”
2. 000000000000000000
3. 000000000000 “000000”
4. 0000000000 + 0000
5. 00000
6. 0000000000

□□□ □□□□ - □□□□ <csa_modify> □□□□□□□□□□□□□□□□

□□□□□

□□□□□□□□□□□□□□□□

□□□□□□□

- □□□□□□□□□□
- □□□□□□□
- □□□□□□□□7□□3□□1□□
- □□□□□□□□□□□□□□□□
- □□□□□/□□
- □□□□□□□
- □□□□□□□□□□□□

□□□□□□□

- □□□□□□□
- □□□□□□□□
- □□□□□□□□□10%□
- □□□□□□□□
- □□□□□□□□□□□□□□□□□□□□

□□□□□

□□□□□□□

- □□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□□<20%□□□□
- □□□□□□□□□□□□□

□□□□□□

□□□□□□□□◀◀

□□□□□□

- □□□□□□□□ ↔ □□□
- □□□□□□□□
- □□□□□□□□□□□□□□□□
- □□□□
- □□□□
- ◀◀◀ □□□□□□
- □□□□□□

□□□□□

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

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□



1. □□□ □□ → □□□□
2. □□ □□□ □□□
3. □□□□□□□□□□□□
4. □□ “□□□□□□□□” □□
5. □□□□□□□□□□□□□□□□
6. □□□□□□□□□□□□□□□□
7. □□□□□□□□□□□□□□□□

□□□

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

□□□□□□

- □□ `can_impersonate` □ `admin` □□
- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□□□□□

□□□ □□□□ - □□□□ `<customer_care>` □□□□□□□□□□

□□□□□□

□□□□□□□□□□□□□□□□□□□□

□□□□□□

- 網路連線
- 網路存取
- 網路存取權限
- 網路存取權限
- 網路eSIM存取權限
- 網路存取
- GPS存取權限

網路Web存取 (PWA)

- 網路存取權限
- 網路存取權限
- 網路存取權限
- 網路存取權限

網路存取權限

網路存取權限

網路存取

1. 網路存取 “網路”
2. 網路存取
3. 網路存取權限1
4. 網路存取
5. 網路存取權限
6. 網路存取

網路存取

- 網路8
- 網路存取
- 網路存取
- 網路存取
- 網路存取 (!@#\$%^&*)

□□□□

□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.
- 4.
5. API

API

-
-
-
-
-

API <concepts_api>

"API"

□□□□

- □□□□ - □□□□
- □□□ - □□□□□□□□□□
- **API** □□ - □□□□□□
- □□□□ - □□ 2FA □□□

□□□□

□□□□□□□□□□

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

□□□□□□

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

□□□

□□□□

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

□□□□

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

□□□□

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

□□□□□□

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

API

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

API

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

API

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

API

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

API

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

OmniCRM API

OmniCRM API

📁📁📁📁📁

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

📁📁📁📁

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

📁📁📁📁📁📁 | 📁 **Ansible** 📁📁📁

📁📁📁📁📁

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

📁📁📁📁

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

📁📁📁📁📁 | 📁📁📁 | 📁📁📁

📁📁📁📁📁

📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁📁

📁📁📁📁

- 📁📁📁📁📁📁📁📁
- 📁📁📁📁📁📁📁

- 000000000000
- 00000000
- 000000000000
- Google 000000000000

000000 | 0000 | 0000

0000

0000000000000000 - 0 SIM 00000000 IP 0000

00000

- 00 SIM 00000000
- 00000000 IP 0000
- 000000000000
- 000000000000
- 00000000
- 00000000

000000

0000000000

00 000000 0000000000000000

000000000

- 000000000000
- 000000000000
- 00000000
- 00 PDF 000000
- 00000000
- 0000000000

00000000 | 00000000

API

API 的基礎知識

認證

- **JWT** 認證
- **(2FA)** 認證 TOTP 認證
- **(RBAC)** 認證
- 認證
- 認證
- 認證

認證 | **RBAC** 認證 | **2FA** 認證

集成

OmniCRM 集成

- **CGRateS** - 集成
- **Stripe** - 集成
- **Mailjet** - 集成
- **Google** 廣告 - 集成
- **RESTful API** - 集成

API 集成 | **Stripe** 集成 | **Mailjet** 集成

部署

部署的基礎知識

部署

- 部署 - 部署
- 部署 - 部署 + 部署 + 部署
- 部署 - 部署
- 部署 - 部署

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXX | XXXXXXXX

XXXXXXXXXX

XXXXXXXXXX OmniCRM 与 XXXXXXXX XXXXXXXXXXXX

XXXXXX

- XXXXXXXXXXXXXXXXXXXX
- XXXXXXXX
- XXXXXXXXXXXXXXXXXXXX
- XXXXXXXX

XXXXXXXXXX

XXXXXX ? ? XXXXX

与 XXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXX

- XXXXXXXX
- XXXXXXXXXXXX
- XXXXXXXXXXXX
- XXXXX

XXXXX | XXXXX

OmniCRM - ██████████ BSS/OSS

██████2025-12-23