

前端頁面 API 整合說明文件(千瑜)

本文件依照前端頁面流程，說明各頁面所需使用的 API 及其規格。

Base URL: <http://localhost:3000/api>

前置作業

→ 到 .env 檔設定連線資料庫所需資訊

一、登入頁 (Login Page)

1. 使用者登入

- 功能說明: 使用者輸入 Email 與密碼進行登入，成功後取得 Token。
- 對應 API: POST /auth/login
- API 合約:

- Request Body:

```
{  
  "email": "user@example.com",  
  "password": "password123"  
}
```

- Response (Success):

```
{  
  "user_id": 1,  
  "display_name": "Jasmine",  
  "email": "user@example.com",  
  "token": "eyJhbGciOiJIUzI1NiIsIn..."  
}
```

- Response (Error):

- 401: {"error": "INVALID_CREDENTIALS", "message": "Email 或密碼錯誤"}

二、註冊頁 (Sign-up Page)

1. 註冊新帳號

- 功能說明: 建立新使用者，註冊成功後回傳 Token 。
- 對應 API: POST /auth/signup
- API 合約:

- Request Body:

```
{  
  "email": "user@example.com",  
  "password": "password123",  
  "display_name": "Jasmine"
```

- ```
 }

 ○ Response (Success):
 {
 "user_id": 1,
 "display_name": "Jasmine",
 "email": "user@example.com",
 "token": "eyJhbGciOiJIUzI1NilsIn..."
 }

 ○ Response (Error):
 ■ 400: {"error": "EMAIL_TAKEN", "message": "此 Email 已被註冊"}
```

### 三、第一次卡片設定頁 / 編輯卡片 (Card Setup Wizard)

此頁面用於讓使用者從系統支援的卡片中，挑選自己擁有的卡片 (最多 3 張)。

#### 1. 取得所有可用卡片 (下拉選單資料源)

- 功能說明: 顯示系統支援的所有銀行與卡片清單，供使用者選擇。
- 對應 API: GET /cards
- API 合約:
  - **Response:**

```
[
 {
 "card_id": 101,
 "bank_name": "A 銀行",
 "card_name": "A 銀行無限卡"
 },
 ...
]
```

#### 2. 取得我目前的卡片 (僅用於「編輯模式」)

- 功能說明: 若使用者是從主頁點「編輯卡片」進來，需先呼叫此 API 回填目前的設定。
- 對應 API: GET /user/cards
- Header: Authorization: Bearer <token>
- API 合約:
  - **Response:**

```
[
 {
 "user_card_id": 1,
 "card_id": 101,
 "bank_name": "A 銀行",
 ...
]
```

```
 "card_name": "A 銀行無限卡"
 }
]
```

### 3. 儲存我的卡片設定

- 功能說明: 將使用者選好的卡片 ID 送回後端儲存。
- 對應 API: PUT /user/cards
- Header: Authorization: Bearer <token>
- API 合約:
  - Request Body:

```
{
 "card_ids": [101, 102] // 陣列長度最多 3
}
```
  - Response (Success):

```
{
 "user_id": 1,
 "card_ids": [101, 102]
}
```
  - Response (Error):
    - 400: {"error": "INVALID\_INPUT", "message": "最多只能選三張卡"}

## 四、使用者主頁 (Dashboard)

### 1. 頂部個人資訊區塊 (Top Info Section)

- 功能說明: 顯示使用者名稱與本月消費概況。
- 資料來源:
  1. 使用者名稱: 來自 GET /user/me。
  2. 本月消費總額: 來自 GET /dashboard 回傳的 summary 物件。
- 對應 API 1: GET /user/me
  - Response: {"user\_id": 1, "display\_name": "Jasmine", ...}
- 對應 API 2: GET /dashboard (部分欄位)
  - Response: {"summary": { "current\_month\_spending": 5000 }, ...}

### 2. 我的信用卡區塊 (My Cards Section)

- 功能說明: 顯示使用者持有的信用卡及其本月累積回饋資訊。
- 對應 API: GET /dashboard
- 資料位置: 回傳 JSON 中的 cards 陣列。
- Header: Authorization: Bearer <token>
- API 合約:

- **Response:**

```
{
 "summary": { ... },
 "cards": [
 {
 "card_id": 101,
 "bank_name": "A 銀行",
 "card_name": "無限卡",
 "month_spending": 3000,
 "reward_detail": [
 { "reward_mode": "cash", "reward_amount": 30 }
]
 }
]
}
```

### 3. 近期消費紀錄列表 (Transaction List)

- 功能說明: 顯示底部的消費紀錄表格, 支援換頁與篩選。
- 對應 API: GET /get\_transactions
- Header: Authorization: Bearer <token>
- 參數說明 (Query Parameters):

| 參數名稱      | 必填 | 預設值  | 說明           | 範例      |
|-----------|----|------|--------------|---------|
| page      | 否  | 1    | 目前頁碼         | 1       |
| limit     | 否  | 10   | 每頁顯示筆數       | 10      |
| year      | 否  | 2025 | 篩選年份         | 2025    |
| month     | 否  | 11   | 篩選月份         | 11      |
| card_name | 否  | (無)  | 篩選特定卡片<br>名稱 | A 銀行無限卡 |
| mcc_type  | 否  | (無)  | 篩選特定消費<br>類別 | dining  |

- API 合約:
  - **Response:**

```
{
 "recent_transactions": [
 {
 "transaction_id": 101,
 "date": "2025-11-15T12:00:00Z",
 "amount": 3000,
 "category": "dining"
 }
]
}
```

```

 "transaction_id": 501,
 "date": "2025-11-20",
 "merchant_name": "Starbucks",
 "amount": 150,
 "card_name": "A 銀行無限卡",
 "reward_mode": "points",
 "reward_amount": 5
 }
],
"page": 1,
"limit": 10,
"total": 50,
"filter": {
 "year": 2025,
 "month": 11,
 ...
}
}

```

#### 4. 篩選器資料源 : 消費類別 (用於消費紀錄列表)

- 功能說明: 提供消費紀錄列表上方「類別篩選」下拉選單的選項內容。
- 對應 API: GET /mcc\_types
- Header: Authorization: Bearer <token>
- API 合約:
  - Response:
 

```
{
 "mcc_types": ["dining", "transportation", "online_shopping"]
}
```

### 五、新增消費頁 (Add Transaction)

#### 1. 選擇商家 (Step 1)

- 功能說明: 商家下拉選單。
- 對應 API: GET /merchants
- API 合約:
  - Response:
 

```
[
 { "merchant_id": 1, "merchant_name": "Starbucks" },
 { "merchant_id": 2, "merchant_name": "Uber" }
```

]

## 2. 選擇分類 (Step 2 - 連動)

- 功能說明: 選了商家後, 撈取該商家支援的分類。
- 對應 API: GET /merchants/:merchantId/categories
- API 合約:
  - Response:

```
[
 { "category_id": 10, "category_name": "餐飲" }
]
```

## 3. 選擇產品 (Step 3 - 連動)

- 功能說明: 選了分類後, 撈取該分類下的產品。
- 對應 API: GET /merchants/:merchantId/categories/:categoryId/products
- API 合約:
  - Response:

```
[
 {
 "product_id": 99,
 "product_name": "拿鐵",
 "default_price": 150
 }
]
```