## 專案結構

[保持原 structure.txt 不變]

## 安裝步驟

### 後端安裝步驟：

1. 建立虛擬環境：

```bash

# 創建專案目錄

mkdir crypto-analytics

cd crypto-analytics

# 創建後端目錄並進入

mkdir backend

cd backend

# 創建並激活虛擬環境

python -m venv venv

# Windows:

venv\Scripts\activate

# Linux/Mac:

source venv/bin/activate

```

2. 安裝核心依賴：

```bash

# 更新pip

python -m pip install --upgrade pip

# 安裝核心依賴

pip install fastapi~=0.103.0

pip install uvicorn~=0.23.0

pip install sqlalchemy~=2.0.0

pip install pydantic~=2.0.0

pip install pydantic-settings~=2.0.0

```

3. 安裝數據庫依賴：

```bash

# PostgreSQL

pip install psycopg2-binary~=2.9.0

pip install alembic~=1.12.0

# Redis

pip install redis~=4.6.0

```

4. 安裝數據處理依賴：

```bash

pip install pandas~=2.0.0

pip install numpy~=1.24.0

pip install scipy~=1.11.0

pip install ccxt~=4.0.0 # 加密貨幣交易所API

```

5. 安裝工具依賴：

```bash

pip install python-dotenv~=1.0.0

pip install httpx~=0.24.1

pip install websockets~=11.0.3

pip install python-jose[cryptography]~=3.3.0

```

6. 安裝開發依賴：

```bash

pip install pytest~=7.4.0

pip install pytest-asyncio~=0.21.1

pip install black~=23.7.0

pip install isort~=5.12.0

```

### 數據庫設置：

1. PostgreSQL 設置：

- 下載 PostgreSQL 15.x：https://www.postgresql.org/download/windows/

- 運行安裝程序：

- 設置端口：5434

- 設置密碼：[您的密碼]

- 創建數據庫：

- 使用 psql 或 pgAdmin 創建 crypto\_analytics 數據庫

2. Redis 設置：

- 下載 Redis for Windows：https://github.com/microsoftarchive/redis/releases

- 下載 Redis-x64-3.0.504.msi

- 安裝 Redis：

- 運行 MSI 安裝程序

- 使用默認端口 6379

- 驗證安裝：

- 檢查 Windows 服務中的 Redis 是否運行中

### 前端安裝步驟：

1. 創建Next.js專案：

```bash

# 返回到專案根目錄

cd ..

# 創建前端應用

npx create-next-app@latest frontend --typescript --tailwind --eslint

cd frontend

```

2. 安裝UI依賴：

```bash

# UI組件

npm install @radix-ui/react-dialog @radix-ui/react-dropdown-menu @radix-ui/react-tabs

npm install @radix-ui/react-popover @radix-ui/react-select

npm install class-variance-authority clsx tailwind-merge

# Icons

npm install lucide-react

```

3. 安裝圖表依賴：

```bash

# 圖表庫

npm install recharts lightweight-charts

npm install d3 @types/d3

```

4. 安裝數據管理依賴：

```bash

# 狀態管理

npm install zustand

# API請求

npm install axios

# 數據獲取

npm install @tanstack/react-query

```

5. 安裝工具依賴：

```bash

# 工具庫

npm install date-fns lodash

npm install @types/lodash

# 表單處理

npm install react-hook-form @hookform/resolvers zod

```

6. 安裝開發依賴：

```bash

npm install -D @types/node @types/react @types/react-dom

npm install -D prettier prettier-plugin-tailwindcss

```

### 開發順序建議：

1. 基礎設置：

- 設置 core/config.py（環境變量配置）

- 設置 core/database.py（PostgreSQL 連接）

- 設置 core/cache.py（Redis 連接）

- 設置 core/logging.py（日誌配置）

2. 數據模型設計：

- 創建 models/market.py

- 創建 schemas/market.py

- 設計模型關係

3. 數據庫遷移：

- 初始化 Alembic

- 配置數據庫連接

- 創建並執行遷移

4. 數據採集服務：

- 實現 Binance API 連接

- 創建數據採集邏輯

- 設置 Redis 緩存策略

- 驗證數據存儲

5. 測試用例：

- 數據庫連接測試

- Redis 連接測試

- 模型操作測試

- 數據採集測試

6. API 開發：

- 實現 API 端點

- 添加錯誤處理

- 實現數據驗證

- 添加緩存層

7. 前端開發：

- 設置基礎架構

- 實現數據顯示

- 添加用戶交互

### 數據存儲策略：

1. PostgreSQL：

- 市場基礎數據

- 分析結果

- 歷史數據

- 使用者數據（如果有）

2. Redis：

- API 請求限制追踪

- 頻繁訪問數據的緩存

- 臨時計算結果

- 即時市場數據緩存

### 注意事項：

1. 使用 .env 文件管理所有配置

2. 實現完整的錯誤處理

3. 設置合理的緩存策略

4. 定期備份 PostgreSQL 數據

5. 監控 Redis 內存使用

6. 遵循代碼風格指南

7. 編寫完整的測試用例

8. 記錄 API 文檔