

電商訂單管理系統 - PHP 後端面試題

背景說明

您需要為電商平台開發訂單管理系統的後端 API。系統需要處理大量訂單資料，並提供 RESTful API 供前端使用。

技術要求

- 使用 PHP 7.0+ 開發
- 實作 RESTful API
- 使用 MySQL 資料庫
- 可使用任何 PHP 框架（Laravel、Symfony 等）或純 PHP
- 資料回傳格式為 JSON

資料庫設計

必要資料表

請設計以下資料表結構（可參考建議欄位，但鼓勵優化）：

基本要求

1. **users** - 用戶資料表
2. **products** - 商品資料表
3. **orders** - 訂單主表
4. **order_items** - 訂單明細表

建議欄位結構

```
sql
```

```
-- 僅供參考，可自行調整優化
```

```
users: id, name, email, created_at, updated_at
```

```
products: id, name, price, stock, created_at, updated_at
```

```
orders: id, user_id, order_number, status, total_amount, created_at, updated_at
```

```
order_items: id, order_id, product_id, quantity, unit_price, subtotal
```

設計重點

- 適當的主鍵和外鍵設計
- 必要的索引以提升查詢效能
- 訂單狀態的合理設計

- 資料類型的正確選擇

API 功能要求

必須實作的 API 端點

1. 訂單管理

| | | |
|------|-------------------------|----------------|
| GET | /api/orders | - 取得訂單列表（支援分頁） |
| GET | /api/orders/{id} | - 取得單一訂單詳情 |
| POST | /api/orders | - 建立新訂單 |
| PUT | /api/orders/{id}/status | - 更新訂單狀態 |

2. 訂單統計（大量資料處理重點）

| | | |
|-----|-------------------|------------------|
| GET | /api/orders/stats | - 訂單基本統計（總數、總金額） |
|-----|-------------------|------------------|

3. 商品管理

| | | |
|-----|---------------|--------|
| GET | /api/products | - 商品列表 |
|-----|---------------|--------|

大量資料處理要求

1. 資料量設定

- 我們將提供包含 1,000 筆訂單的測試資料
- 包含 100 個商品資料
- 包含 200 個用戶資料

2. 效能優化要求

- 訂單列表 API 必須支援分頁（每頁 20 筆）
- 統計 API 需要處理大量資料並在合理時間內回應（< 3 秒）

3. 篩選功能

訂單列表需支援以下篩選：

- 依訂單狀態篩選（可選實作）
- 依日期範圍篩選（可選實作）

具體實作要求

1. 訂單建立 API

json

POST /api/orders

```
{
  "user_id": 1,
  "items": [
    {
      "product_id": 1,
      "quantity": 2
    },
    {
      "product_id": 3,
      "quantity": 1
    }
  ]
}
```

需要驗證：

- 商品是否存在
- 庫存是否足夠
- 自動計算訂單總金額
- 自動生成訂單編號

2. 統計 API 範例回應

json

GET /api/orders/stats

```
{
  "total_orders": 1000,
  "total_amount": 15069192,
  "today_orders": 12,
  "today_amount": 281866
}
```

3. 錯誤處理

實作適當的 HTTP 狀態碼和錯誤訊息：

- 400 Bad Request - 請求格式錯誤
- 404 Not Found - 資源不存在
- 422 Unprocessable Entity - 驗證失敗
- 500 Internal Server Error - 伺服器錯誤

評分標準

資料庫設計 (25%)

- 資料表結構合理性
- 主鍵、外鍵設計正確
- 索引設計（加分項）
- 資料類型選擇適當

基礎功能 (40%)

- 核心 API 端點實作完整性
- RESTful API 規範遵循
- 訂單建立和查詢功能

大量資料處理 (25%)

- 分頁功能實作
- 基本統計功能
- 資料庫查詢效能

程式碼品質 (10%)

- 程式碼結構清晰
- 基本錯誤處理
- JSON 格式回應

額外加分項

- 完整的資料驗證
- 適當的 HTTP 狀態碼使用
- API 文件或註釋

測試資料

我們將提供以下檔案：

- `database_setup.sql` - 資料庫初始化腳本（僅建立空資料庫）
- `generate_test_data.php` - PHP 測試資料生成腳本

使用步驟：

1. 執行 `database_setup.sql` 建立資料庫
2. 完成資料庫設計並建立資料表

3. 修改 `generate_test_data.php` 中的資料庫連線設定

4. 執行：`php generate_test_data.php`

生成的測試資料：

- 200 個用戶
- 100 個商品
- 1000 筆訂單（包含今天約15筆訂單）
- 約2000筆訂單明細

重要提醒：測試資料生成腳本是基於建議的資料表結構和欄位名稱編寫的。如果您的設計有差異，請相應調整腳本或確保欄位名稱一致。

這套測試資料可以有效驗證應試者的實作是否正確，並測試其處理大量資料的能力。

提交要求

1. 程式碼 - 完整的專案檔案
2. 資料庫 **Schema** - SQL 建表語句
3. **README** - 專案安裝和執行說明
4. **API 測試** - 可用 Postman 或 curl 指令示範

注意事項

- 請確保資料庫 Schema 設計能夠匯入測試資料
- API 回應時間應保持在合理範圍內
- 程式碼需具備良好的可讀性
- 我們將使用標準測試資料來驗證 API 功能正確性

祝您面試順利！