

後端外包考題

目標與範圍（Scope）

請完成一個「最小可用」的服務管理後端系統，涵蓋：

- 1. 會員註冊 / 登入（JWT）
- 2. 服務（Service）的 CRUD 操作（權限保護）

資料庫結構設計

預約服務表（AppointmentServices）

```
CREATE TABLE IF NOT EXISTS public."AppointmentServices" (  
  id                                UUID                                NOT NULL PRIMARY  
KEY,  
  name                             VARCHAR(255)                       NOT NULL,  
  description                       TEXT,  
  price                             INTEGER                       NOT NULL,  
  "showTime"                       INTEGER,  
  "order"                           INTEGER                       DEFAULT 0,  
  "isRemove"                       BOOLEAN                       DEFAULT false,  
  "isPublic"                       BOOLEAN                       DEFAULT true,  
  "createdAt"                      TIMESTAMP WITH TIME ZONE NOT NULL,  
  "updatedAt"                      TIMESTAMP WITH TIME ZONE NOT NULL,  
);  
  
-- 索引  
CREATE INDEX IF NOT EXISTS appointment_services__shop_id  
ON public."AppointmentServices" ("ShopId");
```

欄位說明

| 欄位 | 說明 |
|-------------|--------------|
| name | 服務名稱 |
| description | 服務描述 |
| price | 實際價格 |
| showTime | 顯示時間 |
| order | 排序 |
| isRemove | 是否已軟刪除 |
| isPublic | 是否公開於 Client |

會員表（建議結構）

```
CREATE TABLE IF NOT EXISTS public."Users" (  
  id          UUID          NOT NULL PRIMARY KEY,  
  email       VARCHAR(255)  NOT NULL UNIQUE,  
  password    VARCHAR(255)  NOT NULL,  
  name        VARCHAR(255)  NOT NULL,  
  "createdAt" TIMESTAMP WITH TIME ZONE NOT NULL,  
  "updatedAt" TIMESTAMP WITH TIME ZONE NOT NULL  
);
```

不指定 API 路徑：僅描述所需能力與規格。

必用技術棧（Required Stack）

- 語言 / 框架：**TypeScript + Node.js**
- Web：**Koa**（若熟悉 **NestJS** 亦可）
- ORM：**Sequelize**（含 migration + seed）
- 資料庫：**PostgreSQL**（可用 Docker；或 SQLite，但需說明差異）
- 驗證 / 校驗：**Joi**

加分：

- 微服務框架 **Moleculer**
- 測試：**Jest**
- Lint：ESLint / Prettier
- CI/CD：**npm run test** 可自動執行

核心技能驗證重點

1. 專案理解能力

- 分層清楚：controllers / services / repositories / models / middlewares / utils
- 使用 **.env** 管理設定（JWT、DB 設定）
- 清楚的 **README.md**

2. API 開發能力

需具備以下行為與邏輯：

- 會員註冊 / 登入
 - 註冊：email/密碼/名稱（密碼需雜湊）
 - 登入：驗證 email/密碼，回傳 **JWT**
- 服務（Service）CRUD
 - 查詢服務列表（公開）
 - 查詢單一服務（公開）
 - 新增服務（需 JWT）
 - 更新服務（需 JWT）
 - 刪除服務（需 JWT）

回應格式建議但不限制：

```
成功: { "data": ... }  
失敗: { "error": { "code": "xxx", "message": "..."} }
```

3. 資料庫操作

- 提供 migration 與 seed
- 服務資料模型設計（如：id, name, description, price, duration 等）

4. 測試與品質

- 單元測試：涵蓋服務 CRUD 商業邏輯
- 整合測試：涵蓋註冊/登入、服務管理、reCAPTCHA 驗證
- 使用 Jest，可透過 `npm test` 執行

商業規則

- 會員可查看服務列表
- 會員登入後可管理（CRUD）服務
- JWT 驗證保護需要權限的 API

安全性需求

- 密碼需雜湊
- JWT middleware
- Joi 驗證請求
- 集中錯誤處理（避免內部堆疊外洩）
- 加分：Rate Limit、日誌

測試案例

1. 註冊/登入（成功 / 驗證失敗）
2. 服務 CRUD（公開查詢 / 需 JWT 的新增/更新/刪除）
3. 權限驗證（未登入存取受保護 API）
4. API 端點可透過 Postman 測試驗證

交付方式

1. GitHub 專案 + README（啟動、環境變數、DB 初始化、API 範例）
2. Postman Collection 或 API 文檔（提供完整的 API 測試範例）

完成定義（DoD）

- 註冊/登入、服務 CRUD 功能正確執行
- JWT 權限保護正常運作
- 所有 API 端點可透過 Postman 成功測試
- 通過測試案例