

# BI Query Bot

一個基於規則的中文自然語言轉 SQL 查詢系統  
畫面介紹：

BI Query Bot

0.1.0

GAS 1.1

topmapi.com

default

POST /query Query Data

Parameters

No parameters

Request body required

application/json

Example Value | Schema

```
{
  "question": "string"
}
```

Responses

Code	Description	Links
200	Successful Response	No links

Media type

application/json

Controls Accept header

Example Value | Schema

default

POST /query Query Data

Parameters

No parameters

Request body required

application/json

```
{
  "question": "統計每類產品的總銷售數量"
}
```

Execute

Clear

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8080/query' \
  -H 'accept: application/json' \
  -H 'content-type: application/json' \
  -d '{
    "question": "統計每類產品的總銷售數量"
  }'
```

Request URL

http://127.0.0.1:8080/query

Server response

Code	Details
200	<div>Response body</div> <pre>{   "sql_query": "SELECT product_name, SUM(sales_amount) as total_sales \n      FROM sales_data \n      GROUP BY product_name \n      ORDER BY total_sales DESC",   "results": [     {       "product_name": "手機電腦",       "total_sales": 330     },     {       "product_name": "智慧手機",       "total_sales": 330     },     {       "product_name": "智慧手錶",       "total_sales": 330     },     {       "product_name": "筆記型電腦",       "total_sales": 330     },     {       "product_name": "耳機",       "total_sales": 330     }   ] }</pre> <div>Response headers</div> <pre>content-length: 446 content-type: application/json date: Thu, 10 Apr 2025 10:23:29 GMT server: nginx</pre>

Download

## 一、系統架構:

### 1. 核心組件

```
bi_query_bot/  
├── app/  
│   ├── main.py          # FastAPI 主應用程式  
│   ├── llm_agent.py     # 自然語言轉 SQL 規則引擎  
│   ├── db.py            # 數據庫連接和初始化  
│   └── query_runner.py  # SQL 查詢執行和驗證
```

### 2. 主要功能模塊

- main.py: API 端點和請求處理
- llm\_agent.py: 中文自然語言轉換為 SQL 查詢
- db.py: 數據庫連接管理
- query\_runner.py: SQL 查詢驗證和執行

## 二、API 端點

1. 查詢端點 (POST /query)
2. 健康檢查 (GET /health)

## 三、錯誤處理

1. 400 無效請求
  - 空問題
  - 不安全的 SQL 操作
2. 500 服務器錯誤
  - 數據庫連結錯誤
  - 查詢執行失敗

## 四、啟動指令

- 啟動服務器

```
python -m uvicorn app.main:app --host 127.0.0.1 --port 8080 --reload
```

- API 文檔訪問

```
http://127.0.0.1:8080/docs
```

## 五、日誌紀錄

- 系統包含完整的日誌記錄：
- 請求接收記錄
- SQL 查詢生成記錄
- 查詢執行結果記錄
- 錯誤追蹤記錄

## 六、數據模型

1. QueryRequest :

```
python Apply to llm_agent.py

class QueryRequest(BaseModel):
    question: str # 必填，最小長度為1
```

## 2. QueryResponse :

```
python Apply to llm_agent.py

class QueryResponse(BaseModel):
    sql_query: str # 生成的SQL查詢
    results: List[Dict] # 查詢結果
    error: Optional[str] # 錯誤信息
```

## 七、功能使用

1. 月度銷售查詢
  - 問法：「上個月的總銷售額」或「上個月的銷售額」
  - 結果：會顯示上個月的總營收
2. 產品銷量統計
  - 問法：「統計每個產品的總銷售數量」或「查詢每個產品的銷售數量」
  - 結果：列出所有產品的銷售總量，按銷量降序排列
3. 銷售額排名查詢
  - 問法：「銷售額最高的前 X 名產品」（X 可以是任何數字）
  - 例如：「銷售額最高的前 5 名產品」或「銷售收入最高的前 10 名產品」
  - 結果：顯示指定數量的最高營收產品
4. 產品日均銷量查詢
  - 問法：「[產品名稱]的平均每日銷售量」
  - 例如：「iPhone 的平均每日銷售量」
  - 結果：顯示特定產品的平均每日銷售數量
5. 本月銷售查詢
  - 問法：「本月每天的總收入」「本月總銷售額」「本月每天的銷售額」
  - 結果：顯示本月每天的銷售收入數據
6. 歷史銷售趨勢
  - 問法：「過去 X 天的銷售趨勢」（X 可以是任何天數）
  - 例如：「過去 30 天的銷售趨勢」
  - 結果：顯示指定天數內的每日銷售收入

7. 預設查詢

- 如果您的問題不符合上述任何模式，系統會：
- 返回最近 10 筆銷售記錄
- 包含：銷售日期、產品名稱、銷售數量和收入

8. 使用建議

- 問題要簡單明確
- 可以靈活使用不同的說法（如「銷售額」、「收入」、「金額」等）
- 數字可以自由指定（如查詢前幾名或過去幾天的數據）
- 產品名稱要與數據庫中的名稱一致