

# Day12 - 請支援 AI - Function Calling (中)

只是問時間似乎無法體會 Function Call 有甚麼特別之處，今天我們模擬從後端取得資料並透過 AI 幫我們分析，看看外掛程式該如何撰寫

**程式目標: 詢問 AI 公司每個產品去年的銷量，並統計占銷量的比例是多少**

1. 撰寫外掛程式 ProductFunction : 實作 java.util.function

```
public class ProductFunction implements Function<ProductFunctionRequest, Response> {
    // 假設從後端可取得產品每年的銷量，若只是 POJO 可以使用 Java 14 才有的 record
    public record Product(String year, String model, Integer sales) {}

    @Override
    public Response apply(Request request) {
        // 模擬從後端取得產品資訊，使用 List 回傳多筆資料
        return new Response(List.of(
            new Product("2022", "PD-1405", 12500),
            new Product("2023", "PD-1234", 10000),
            new Product("2023", "PD-1235", 1500),
            new Product("2023", "PD-1385", 15000),
            new Product("2024", "PD-1255", 15000),
            new Product("2024", "PD-1300", 12000),
            new Product("2024", "PD-1405", 12500),
            new Product("2024", "PD-1235", 15000),
            new Product("2024", "PD-1385", 15000)
        ));
    }

    @JsonInclude(Include.NON_NULL)
    @JsonClassDescription("公司產品銷售列表")
    public record Request(
        // 參數可帶入年份及產品，目前沒特別處理，參數可放在跟後端請求資料時篩選
        @JsonProperty(required = false, value = "year") String year,
        @JsonProperty(required = false, value = "product") String product
    ) {}
}
```

```

        ) {
    }
    //回應資料若有多筆，可以使用 List 回傳，AI 也能根據這些資料搭配 Pr
    public record Response(List<Product> products) {
    }
}

```

2. 註冊外掛程式: 寫一隻 Config 並使用 FunctionCallbackWrapper 將外掛程式包進 Builder 內，這裡須給予幾個設定
  - a. Function Name
  - b. Description
  - c. Response回傳內容

```

//有幾隻 Function 就有幾個 Bean，這裡的描述盡量寫清楚，也可以使用中文
@Configuration
public class AiConfig {
    @Bean
    public FunctionCallback productSalesInfo() {
        return FunctionCallbackWrapper.builder(new ProductFuncT
            .WithName("ProductSalesInfo")
            .withDescription("Get the products sales volume
            .withResponseConverter((response) -> response.p
            .build());
    }
    @Bean
    public FunctionCallback currentDateTime() {
        return FunctionCallbackWrapper.builder(new CurrentDateT
            .WithName("CurrentDateTime") //Function Name
            .withDescription("Get the Date Time") //Descri
            .withResponseConverter((response) -> response.c
            //回傳結果要如何轉換，一個參數通常都直接使用 .toString
            .build());
    }
}

```

3. 在 Option 中定義可被調用的 Function Name，這裡設定的名稱需與上一步一樣

```

@GetMapping("/func")
public String func(String prompt) {
    return chatModel.call(
        new Prompt(prompt,
            OpenAiChatOptions.builder()
                // Funciton可以放多筆，也能依據 API 接口放上合適的 F
                .withFunction("ProductSalesInfo")
                .withFunction("CurrentDateTime")
                .build())
        ).getResult().getOutput().getContent();
}

```

## 程式補充說明

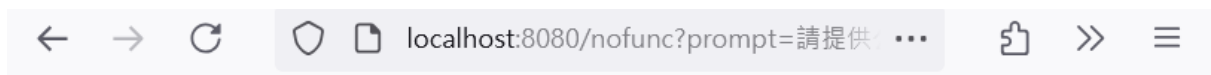
- 產品的 Class 不一定要放在 Function 內，這裡可以放 Entity，取得資料的方式可以透過資料庫，若要綁定 @Service 物件，記得也把 Function Class 設為 @Component
- 傳入的參數可以作為後端資料篩選的參數，減少資料也能減少送給 AI 的 Token 數量
- Spring MVC 的部分可以搭配 Spring Security，有些機密資料不能讓沒權限的人取得
- Options 中可以放多個 Function

看到這邊應該能開始體會和 Spring Boot 框架整合有甚麼好處了，從 Spring MVC / Spring Data JPA / Spring Security 都能整合進來，企業引入 AI 的同時還能兼顧安全性以及資料的彈性

## 測試結果

我詢問的問題是 請提供公司2023年產品銷量, 並舉出特別的地方

未使用Function Call



抱歉，我無法提供**2023**年公司的具體產品銷量數據，因為我的數據截至於**2021**年，且無法獲取即時或最新的商業數據。不過，我可以幫助你分析一些行業趨勢、產品發展或市場表現等相關話題。如果你有特定的公司或產品想要了解，可以告訴我，我會盡量提供相關的背景資訊或知識。

## 使用Function Call



在**2023**年，公司產品的銷量如下：1. **\*\*PD-1234\*\***: 10,000 台 2. **\*\*PD-1235\*\***: 1,500 台 3. **\*\*PD-1385\*\***: 15,000 台 ### 特別的地方 - **\*\*PD-1385\*\***的銷量為15,000台，顯示出這款產品在市場上的受歡迎程度，可能是因為其獨特的功能或者促銷策略。 - **\*\*PD-1234\*\***的銷量為10,000台，是公司其中一款主力產品，表明其穩定的市場需求。 - **\*\*PD-1235\*\***的銷量僅為1,500台，這可能需要進一步分析原因，是否是市場競爭、產品定位或其他因素導致銷量不佳。

## 回顧今天學到的內容:

1. 與後端結合的 Function 寫法
2. 多個 Function 的設定方式
3. 與 Spring 其他模組的整合方式