

Day13 - Function Calling 最終組合技 (下)

原本 Function Call 只打算寫兩篇，不過最近參考其他框架發現 Function Calling 是會被反覆調用的，當你的 Prompt 有可匹配的 Function 時，Spring AI 會先去調用並取得資料，之後會再拿取得的資料確認是否還有可匹配的 Function，並將目前取得的結果再次呼叫可匹配的 Function
來看看怎麼活用這樣的技巧吧

程式目標: 詢問 AI 公司去年最熱銷的產品，並列出該產品所有型號

在不知道 Function Calling 會重複調用時你可能會把產品改成一個 Class，並將所有產品型號放在 List 變數中，這樣做雖然也沒問題，不過長久下來會浪費許多額外花費，因為程式會將所有產品資料包含型號一起送給 AI 處理，當資料多的時候就形成額外的花費

運用重複調用的特性，我們可以增加一隻查詢產品型號的 Function，Request 的參數是產品名稱，Response 則放產品型號的 List，若 Prompt 有提到需要列出產品型號時，Spring AI 就會主動再去調用這隻查詢型號的 Function，下面直接看 Function 如何增加

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

```
public class ProductDetailsFunction implements Function<Product, List<ProductDetail>> {
    public record ProductDetail(String product, List<String> models) {}
    @Override
    public Response apply(Request request) {
        //模擬資料，企業通常會透過DB或是其他API查詢內容
        List<ProductDetail> productModels= List.of(
            new ProductDetail("PD-1405", List.of("1405-001", "1405-002")),
            new ProductDetail("PD-1234", List.of("1234-1", "1234-2")),
            new ProductDetail("PD-1235", List.of("1235-4", "1235-5")),
            new ProductDetail("PD-1385", List.of("1385-1", "1385-2")),
            new ProductDetail("PD-1255", List.of("1255-1", "1255-2")),
            new ProductDetail("PD-1300", List.of("1300-1", "1300-2"))
        );
    }
}
```

```

        );
        //模擬查詢後回傳的結果
        ProductDetail models = productModels.stream().filter(
            return new Response(models);
    }

    @JsonInclude(Include.NON_NULL)
    @JsonClassDescription("產品型號列表")
    public record Request(
        //參數只需帶入產品
        @JsonProperty(required = false, value = "product") @J
    ) {
    }
    //回傳的結果最好包含產品一起回傳，若只回傳型號清單 AI 比較容易失誤
    public record Response(ProductDetail models) {
    }
}

```

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 ProductFunc
            .WithName("ProductSalesInfo")
            .withDescription("Get the products sales volume")
            .withResponseConverter((response) -> response.p
            .build());
    }
}

//新增的Bean，描述盡量清楚，關鍵字可加上中文讓 AI 更好判斷

```

```

@Bean
public FunctionCallback productDetailsInfo() {
    return FunctionCallbackWrapper.builder(new ProductDetail
        .WithName("ProductDetailsInfo")
        .withDescription("Get the product's model(產品型
        .withResponseConverter((response) -> response.m
        .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("ProductDetailsInfo")
                .build())
        ).getResult().getOutput().getContent();
}

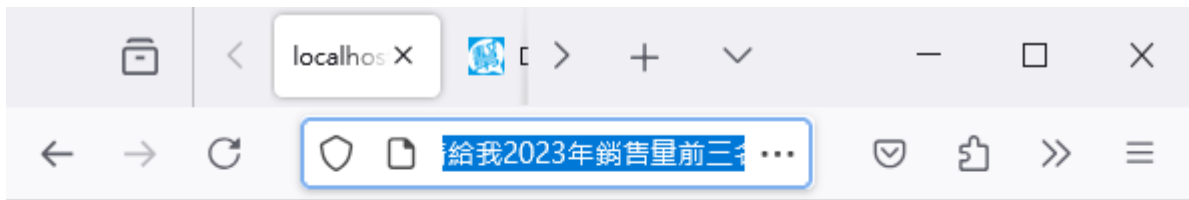
```

程式補充說明

- Request 只需放入產品名稱，這裡被二次調用時會拿第一次產品銷量的產品名稱作為參數傳入
- 在 ProductDetailsInfo 模擬的部分，實際可改由 Spring Data JPA 或是透過 RestClient 呼叫 API 取得資料
- Config 跟 Options 中記得放入新的程式

測試結果

直接看測試結果，這次詢問的問題是 請給我2023年銷售量前三名的產品，並列出該產品所有型號，使用表格方式呈現



以下是2023年銷售量前三名的產品及其所有型號的表格：

產品型號	銷售量	所有型號
PD-1385	15000	1385-1, 1385-2, 1385-3
PD-1234	10000	1234-1, 1234-2, 1234-3, 1234-4
PD-1235	1500	1235-4, 1235-5

如有需要進一步的資訊，請告訴我！

表格的部分是採用 Markdown 格式，若有前端程式搭配就能呈現表格的內容

產品型號	銷售量	所有型號
PD-1385	15000	1385-1, 1385-2, 1385-3
PD-1234	10000	1234-1, 1234-2, 1234-3, 1234-4
PD-1235	1500	1235-4, 1235-5

回顧今天學到的內容：

1. 多層 Function Calling 的撰寫方式
2. Prompt 中若能匹配 Function，Spring AI 就會主動調用

凱文大叔個人的心得：AI 有了 Function Calling 便能與企業資訊系統整合，以前需要人工處理的 BI 現在能透過 AI 的推理能力協助完成，目前示範的程式雖然都只是呈現在網頁上，不過 Function 其實能呼叫任何程式，也能將分析後的資料送給其他報表程式來呈現結果，甚至結合 ERP 執行下單的動作，怎麼運用就要靠大家的想像力了