

Day3 - Hello AI World

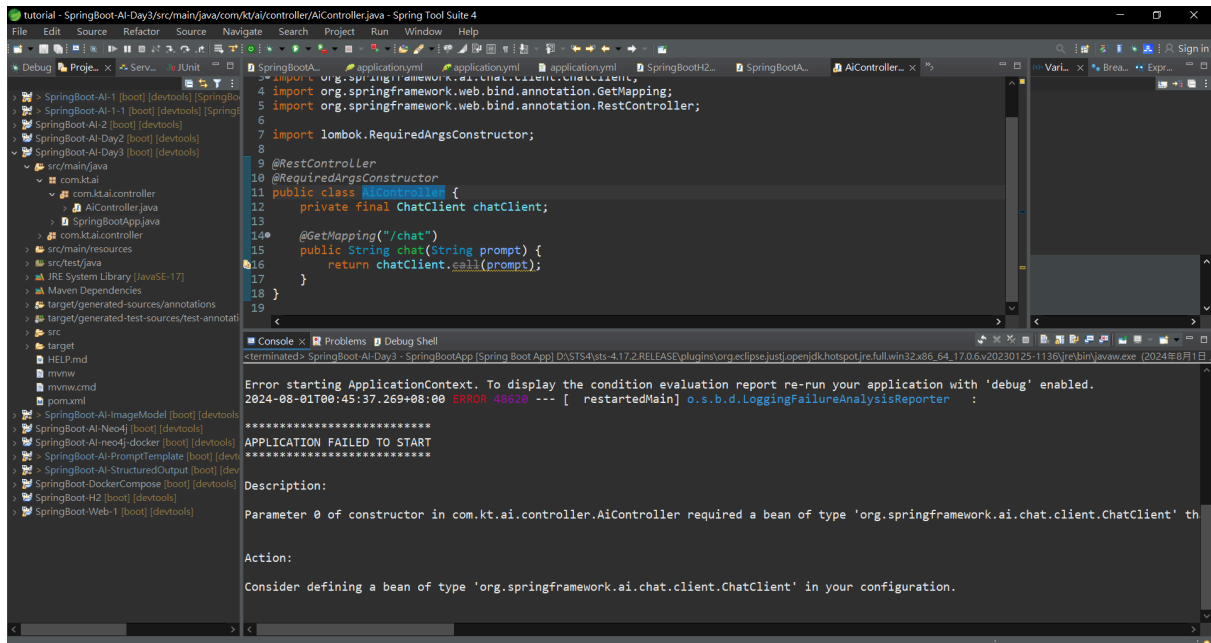
寫程式不免俗要來個 Hello World，依據開發 Spring Boot 程式的經驗，我們只需定義變數，並透過自動注入的機制就能開始使用，為了測試方便就直接寫個 Controller

```
@RestController
@RequiredArgsConstructor
public class AiController {
    private final ChatClient chatClient;

    @GetMapping("/chat")
    public String chat(String prompt) {
        return chatClient.call(prompt);
    }
}
```

| @RestController：專門用來開發 API 的標註

沒想到一執行就直接跳出錯誤



甚麼?! 系統直接找不到這個 Bean

required a bean of type
'org.springframework.ai.chat.client.ChatClient' that could not
be found.

查資料才發現 Spring AI 做了大幅度的異動（[官方完整異動說明](#)）

Upgrading to 1.0.0.M1

On our march to release 1.0.0 M1 we have made several
breaking changes. Apologies, it is for the best!

ChatClient changes

A major change was made that took the 'old'

`ChatClient` and moved the functionality into `ChatModel`. The
'new' `ChatClient` now takes an instance of `ChatModel`.

This was done to support a fluent API for creating and
executing

prompts in a style similar to other client classes in the Spring
ecosystem, such as

`RestClient`, `WebClient`, and `JdbcClient`. Refer to the [JavaDoc]
(docs.spring.io/spring-ai/docs/1.0.0-SNAPSHOT/api/) for
more information on the Fluent API, proper reference
documentation is coming shortly.

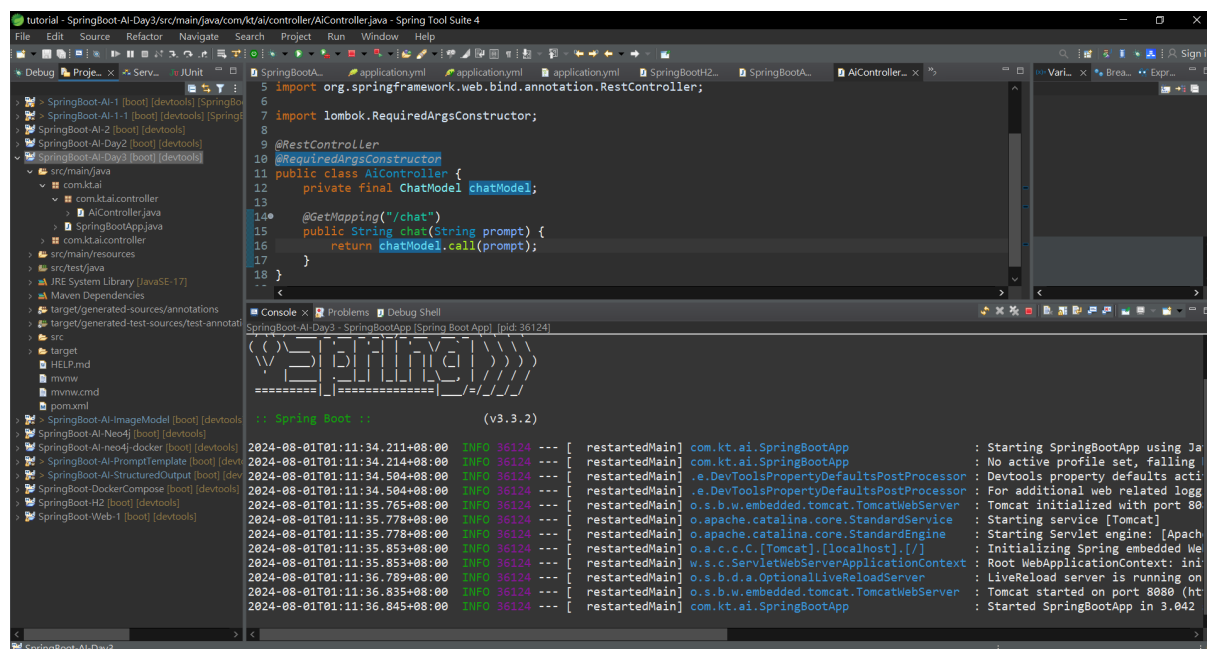
We renamed the 'old'

`ModelClient` to `Model` and renamed implementing classes, for
example `ImageClient` was renamed to `ImageModel`. The `Model`
implementation represents the portability layer that converts
between the Spring AI API and the underlying AI Model API.

簡單的說原本的 ChatClient 改成 ChatModel，而新的 ChatClient 則可以做更複雜的
應用

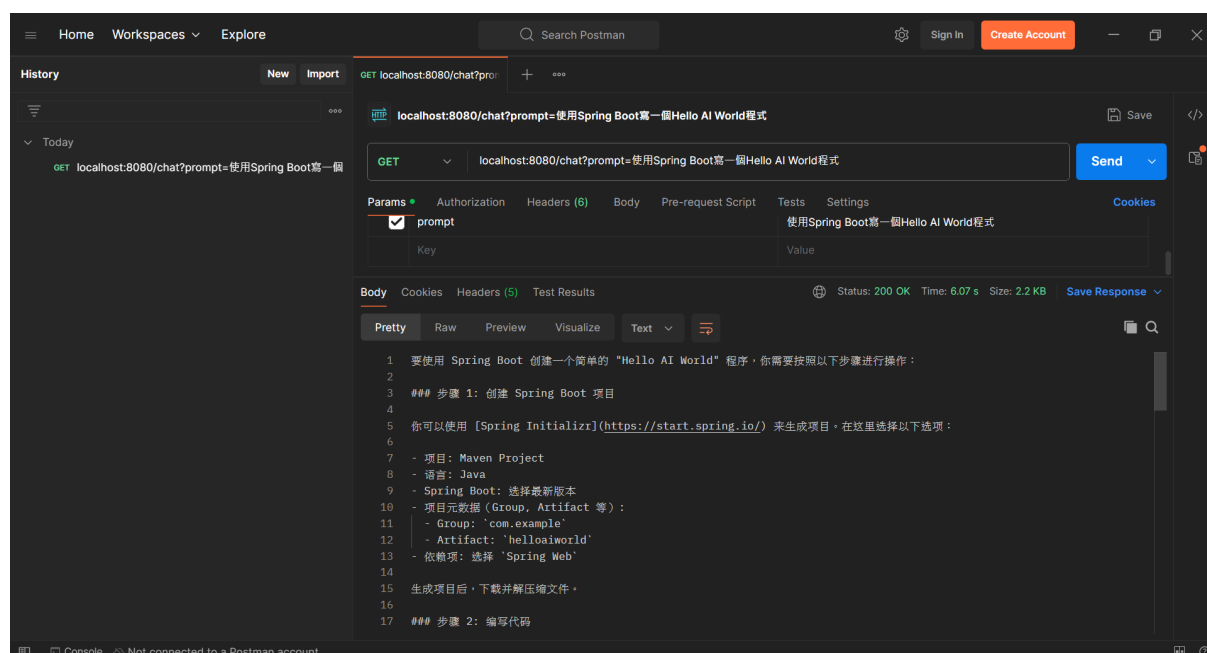
最簡單的修正就是將 ChatClient 改成 ChatModel，其實在昨天的程式也能看出一些
端倪，當還沒設定 API key 時，系統報的錯誤就是 openAiChatModel 找不到 API

key 設定



改成 ChatModel 後馬上就能正常啟動，網路上目前還是一堆舊寫法的教學，大家要特別注意

下面直接使用 Postman 進行測試，大家看看效果吧



總結今天學到的內容:

- 寫一個簡單的 Controller
- 了解 Spring AI 1.0.0-M1 重大改變
- 透過 ChatModel 與 AI 連線並透過 Postman 完成測試