Day4 - 如何像 ChatGPT 產生流式輸出

由於 AI 產生內容得靠伺服器運算後產生結果，資料多的話得等不少時間，若能產生資料後馬上分段送出，可大大提升使用者感受，要達到這種效果主要靠的是 SSE 伺服器主動推播協定

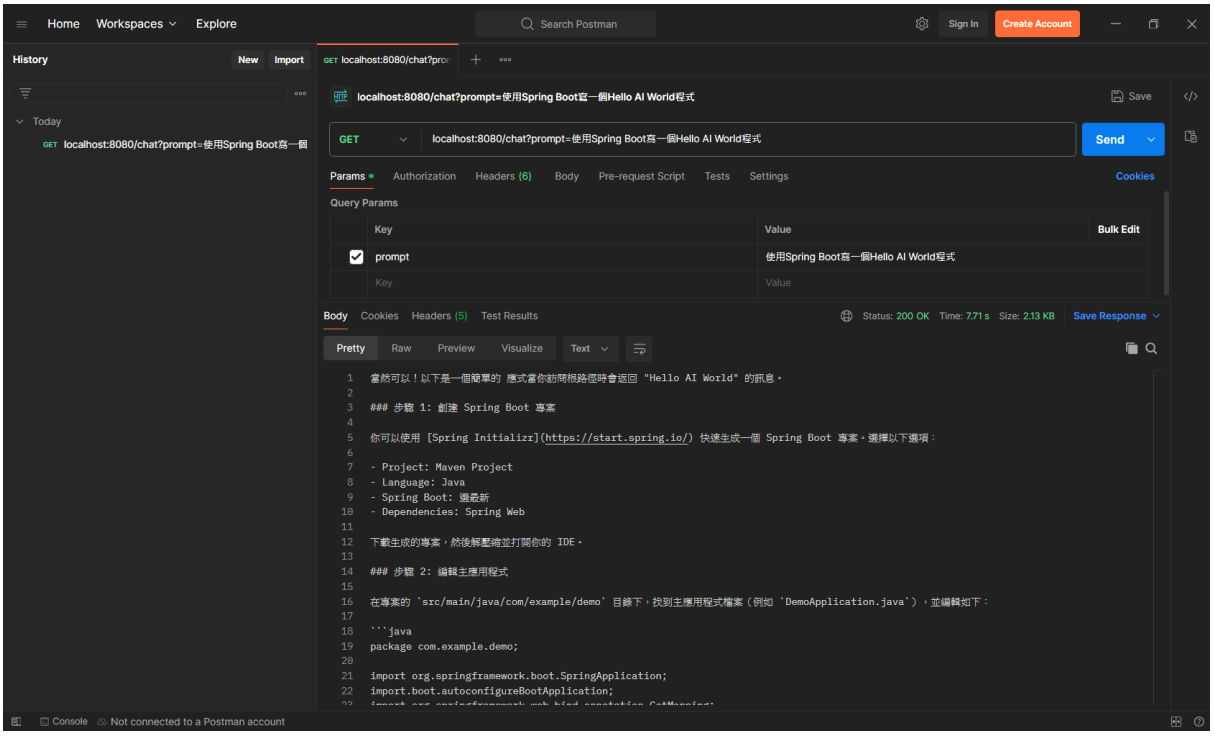
前端只需寫個事件監聽函式，收到資料就立刻加在聊天訊息上，不需等全部內容產出就能開始觀看，而且資料產生的速度遠比人觀看的速度快，使用者可以不用對著螢幕發呆

Spring AI 的 ChatModel 也實作了這種方法，我們只需改兩個小地方即可達成流式輸出效果

1. 將原本的 chatModel.call(prompt) 改為 chatModel.stream(prompt)
2. 將回傳值由 String 改為 Flux<String>

這個 Flux 背後正是使用 SSE 技術，不過 Spring 將其封裝起來，後端實際上並不用花太多功夫處理

我們先看看 Postman 測試結果



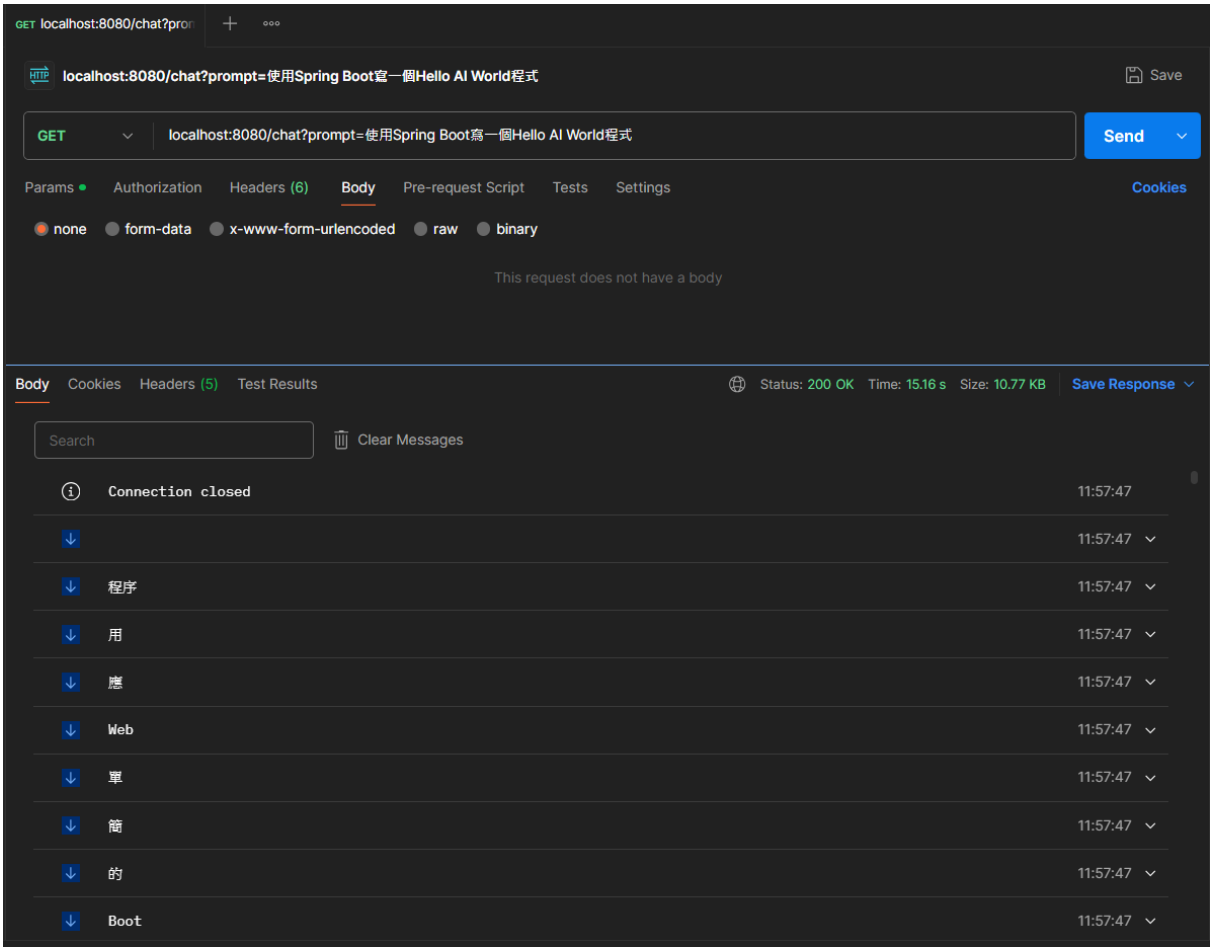
結果跟昨天一樣，看起來能正常運行，不過似乎少了甚麼？

由於沒指定輸出格式 Postman 還是將結果收集完才送出

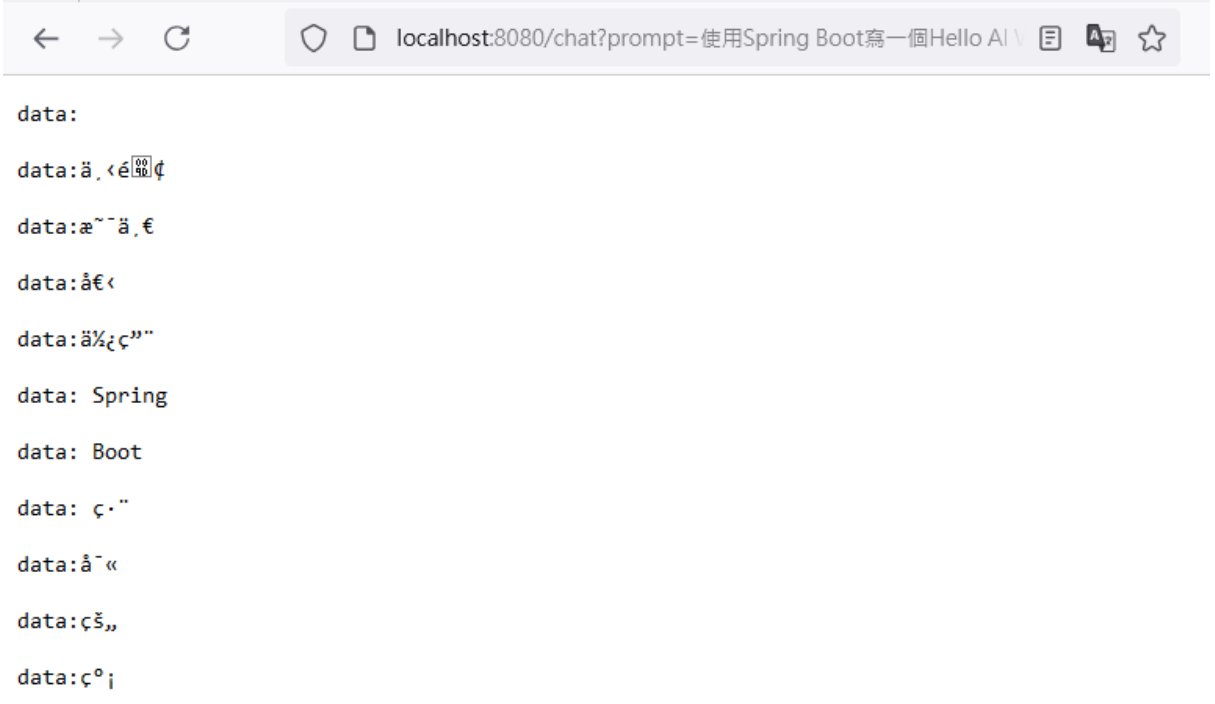
我們只需在 API 接口加上 `produces = MediaType.TEXT_EVENT_STREAM_VALUE`

```java
@GetMapping(value = "/chat", produces = MediaType.TEXT_EVENT_STREAM_VALUE)
public Flux<String> chat(String prompt) {
    return chatModel.stream(prompt);
}
```

這樣 Postman 就會知道 API 回傳內容要採用流式輸出



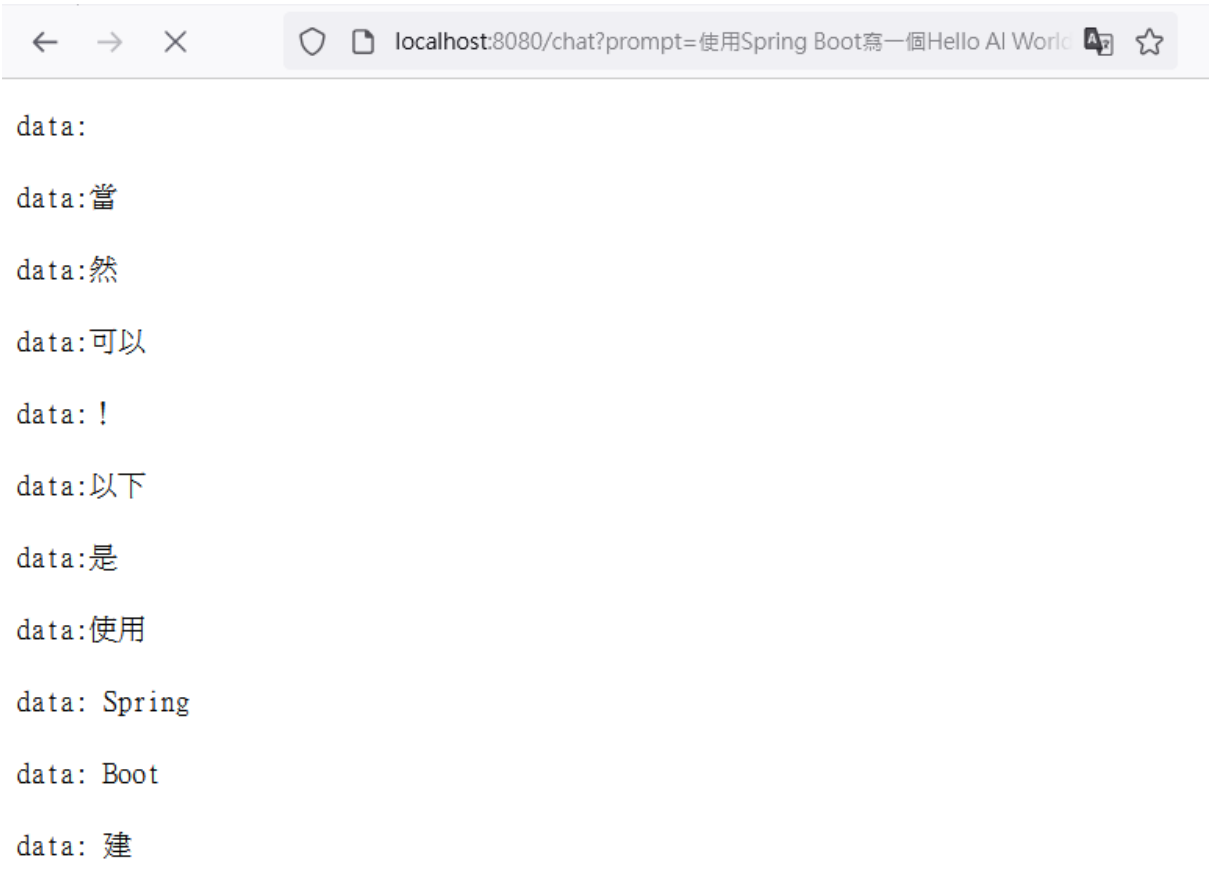
另外若直接呈現在網頁上會有中文亂碼問題



可以在 application.yml 增加字元格式，將其設為 UTF-8

```
server:
  servlet:
    encoding:
      charset: UTF-8
      enabled: true
      force: true
```

如此就能正確顯示中文了



回顧一下今天學到甚麼

- 將 AI 回傳結果改為流式輸出
- 解決網頁執行時中文亂碼問題