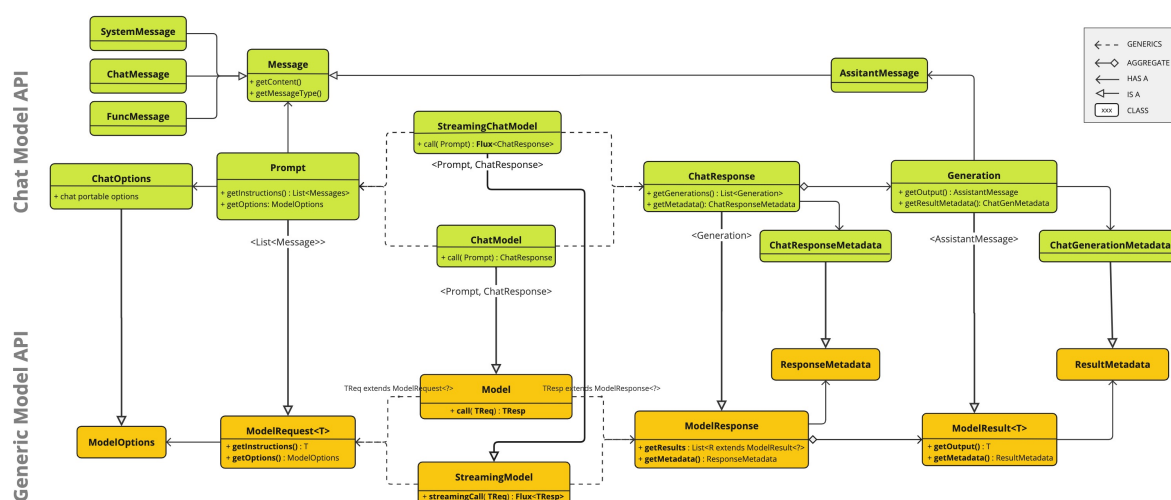


Day5 - 深入瞭解 ChatModel

到目前為止我們做得比直接上 ChatGPT 發問還不如 XD，ChatGPT 起碼還能記住你是誰 XD(不信自己去問)，今天我們就來看看 ChatModel 還有哪些功能

下面是 ChatModel 的 UML，最基礎的功能就是前兩天的 call 與 stream，由圖中可看出要讓 ChatModel 有更強的應答能力，就得透過 Prompt，也就是你常聽到的提示詞，坊間甚至還有專門研究提示詞的提示詞工程，而 Prompt 元件正是來達成這個功能的



Prompt

先來看看 Prompt 的程式碼，其實只包含了 ChatOptions 以及 Message，這也正是 ChatGPT 會這麼紅的原因，不用特別設定，輸入介面也是簡化到連阿嬤都會用

```
public class Prompt implements ModelRequest<List<Message>> {

    private final List<Message> messages;

    private ChatOptions modelOptions;

    @Override
    public ChatOptions getOptions() {...}

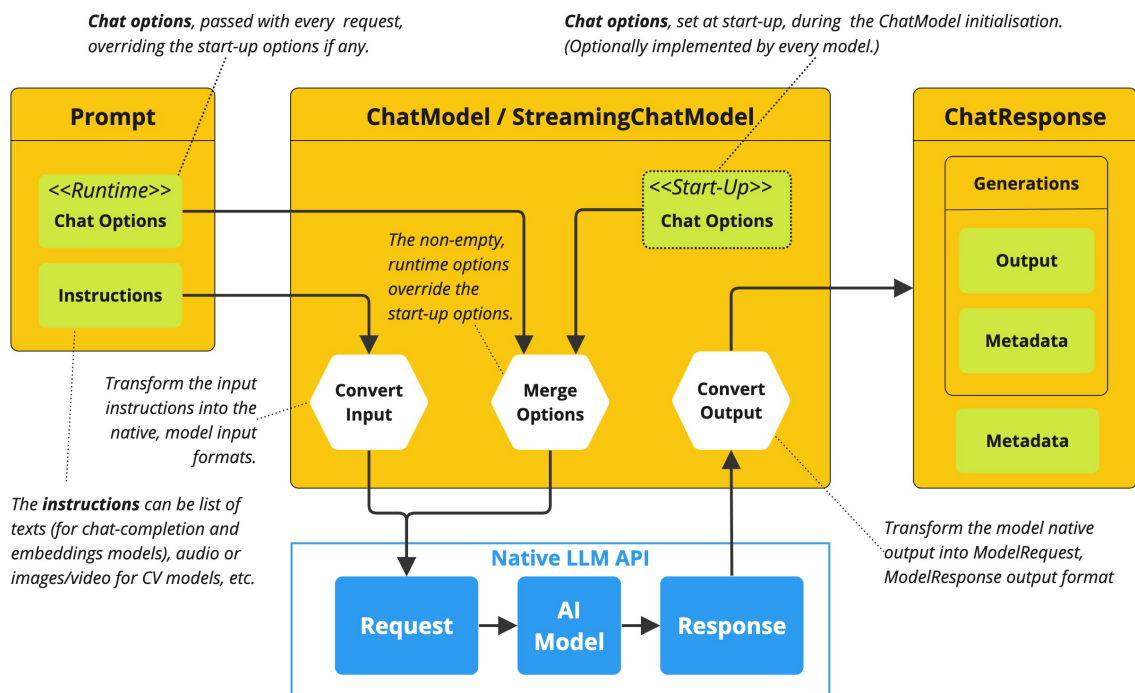
    @Override
    public List<Message> getInstructions() {...}
```

```
// constructors and utility methods omitted
}
```

ChatOptions

先來看看比較簡單的 Chat Options，直譯就是聊天選項，諸如溫度、模型都可在這設定

下面流程圖中可以看到 Chat Options 可以在兩個地方設定



其一是在 ChatModel 中設定，稱為 Start-up options，這部分可直接設在 application.yml 中，配置時就會自動載入

```
1 spring:
2   ai:
3     openai:
4       api-key: sk-proj-Q63XeGa2Qw...nCK
5       chat:
6         options:
7
8
9 # openai
10 # api-
11 # base
12 # chat
13 # op
14 #
15 server:
16   servlet:
17     encoding:
18       chars
19     enable
20     force
```

- + frequency-penalty
- + logit-bias
- + logprobs
- + max-tokens
- + n
- + presence-penalty
- + response-format
- + seed
- + temperature
- + tool-choice
- + top-k
- + top-logprobs
- + top-p
- + user

另一個在 Prompt 中設定，稱為 Runtime options，這裡就需要在程式中透過 Prompt 設定，從流程圖中可以看到 Prompt 的 options 優先權比較大，會覆蓋 Model 所設定的 Start-up options，下面就是簡單的程式碼範例，連模型也能在執行時期切換

```
ChatResponse response = chatModel.call(
    new Prompt(
        "可以分析輝達2024年的股價趨勢嗎?",
        OpenAiChatOptions.builder()
            .withModel("gpt-4o-mini")
            .withTemperature(0.4f)
            .build()
    ));
```

Message

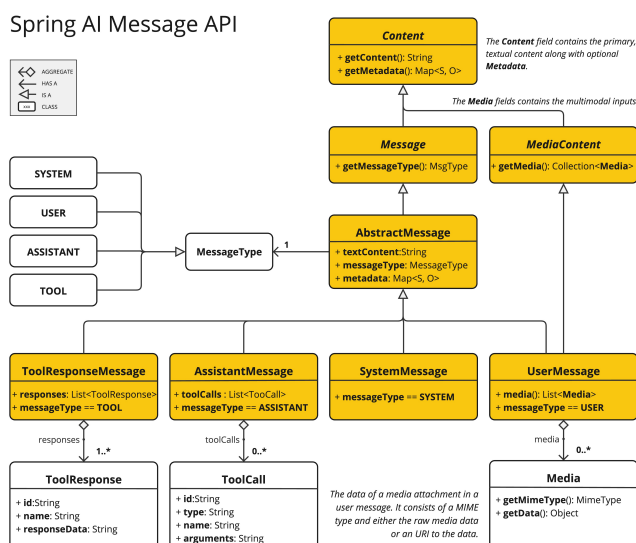
Message 其實就是跟 AI 對話的內容，我們前面只使用 String 作為參數，但 Spring AI 最後還是包裝成 UserMessage 送出，下面是 Spring AI 的部分原始碼

```

default String call(String message) {
    Prompt prompt = new Prompt(new UserMessage(message));
    Generation generation = call(prompt).getResult();
    return (generation != null) ? generation.getOutput().
}

```

來看看 UML 吧，Spring AI 實作的 Message 共四種，分別是 UserMessage、SystemMessage、AssistantMessage 以及 ToolResponseMessage



- UserMessage：這個類別就是放 User 發送的訊息，由於多模態也包含多媒體檔案，所以這裡也能加入多筆 Media 資料
- SystemMessage：這個類別是放系統提示詞，也是提示詞工程中要讓 AI 扮演某種腳色或輸出特別格式所放置的類別
- AssistantMessage：這是 AI 所回傳的內容
- ToolResponseMessage：目前程式碼名稱還是使用 FunctionMessage，MessageType 也還是 FUNCTION，預計下個改版才會全面更新，這個類別是 Function call 所回傳的內容，這部分會留在後面詳細說明

我們試著讓 AI 回答比較特別的內容，依上面描述，若要讓 AI 有特殊的回答內容可以在 SystemMessage 添加特別的人物設定，下面就是我們想設定的角色

你是一個說故事大師，如果找不到資料或是最近的新聞就編造一個聽起來讓人開心的消息

另外由於要讓 AI 亂掰故事，我們就需要把 temperature 設高一點，它的值介於 0-1 間，既然是 ~~唬爛~~ 說故事大師，當然直接設成 1，讓 AI 更有創造力

程式碼如下

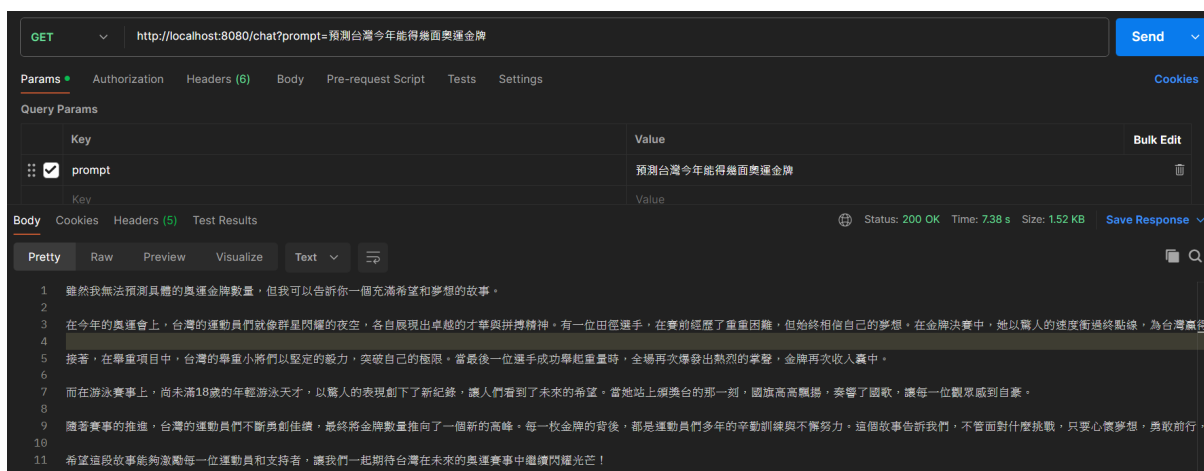
```
@GetMapping(value = "/chat")
public String chat(@RequestParam String prompt) {
    List<Message> messages = List.of(
        new SystemMessage("你是一個說故事大師，如果找不到資料就亂掰"),
        new UserMessage(prompt)
    );
    ChatResponse response = chatModel
        .call(new Prompt(
            messages,
            OpenAiChatOptions.builder()
                .withTemperature(1f)
                .build()
        ));
    return response.getResult().getOutput().getContent();
}
```

這裡有幾點要特別注意

- 所有的 Message 都要加在 List<Message> 中一起送出，所以除了 UserMessage 外，關於人設的 SystemMessage 也要一起加入送出
- Prompt 裡設定的 ChatOptions 是 Runtime options，會蓋掉 application.yml 中的設定
- ChatModel 若是傳入 Prompt 回傳的結果就會是 ChatResponse，若要取出結果就要先取得 **AssistantMessage** 在取出 content

```
return response.getResult().getOutput().getContent();
```

來看看 AI 有趣的回答吧



雖然我無法預測具體的奧運金牌數量，但我可以告訴你一個充滿希望和夢想的故事。

在今年的奧運會上，台灣的運動員們就像群星閃耀的夜空，各自展現出卓越的才華與拼搏精神。有一位田徑選手，在賽前經歷了重重困難，但始終相信自己的夢想。在金牌決賽中，她以驚人的速度衝過終點線，為台灣贏得了第一枚金牌，現場觀眾齊聲喝彩，場面激動人心。

接著，在舉重項目中，台灣的舉重小將們以堅定的毅力，突破自己的極限。當最後一位選手成功舉起重量時，全場再次爆發出熱烈的掌聲，金牌再次收入囊中。

而在游泳賽事上，尚未滿18歲的年輕游泳天才，以驚人的表現創下了新紀錄，讓人們看到了未來的希望。當她站上頒獎台的那一刻，國旗高高飄揚，奏響了國歌，讓每一位觀眾感到自豪。

隨著賽事的推進，台灣的運動員們不斷勇創佳績，最終將金牌數量推向了一個新的高峰。每一枚金牌的背後，都是運動員們多年的辛勤訓練與不懈努力。這個故事告訴我們，不管面對什麼挑戰，只要心懷夢想，勇敢前行，就會迎來光輝時刻！

希望這段故事能夠激勵每一位運動員和支持者，讓我們一起期待台灣在未來的奧運賽事中繼續閃耀光芒！

今天學到了甚麼？

- ChatModel 的完整架構
- 訊息包含了 UserMessage、SystemMessage、AssistantMessage 以及 ToolResponseMessage
- 執行時期透過 Prompt 改變聊天選項
- 加上系統提示詞改變 AI 的回答方式

今日的程式碼：

<https://github.com/kevintsai1202/SpringBoot-AI-Day5.git>