

# Lecture 15: Implementing Prompt Template

## Why We Need this?

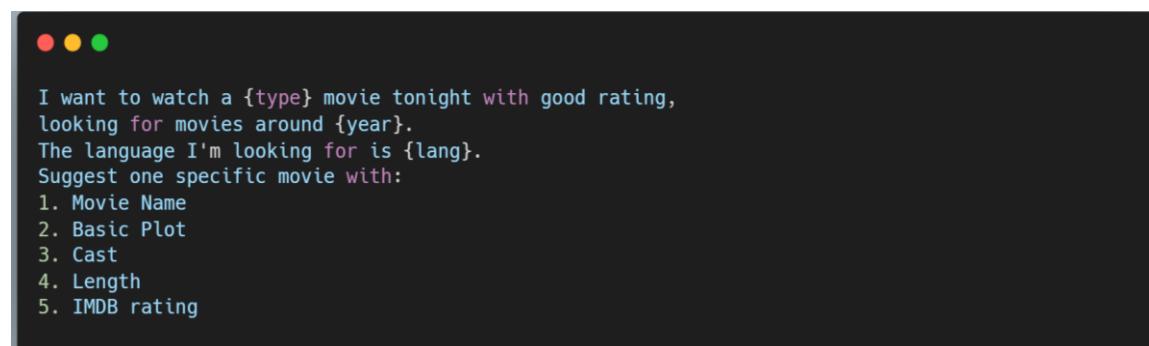
- Earlier, we created a **prompt template** but didn't actually implement it.
- Now, we want to build a **real endpoint** that takes user inputs and dynamically generates prompts.
- Goal: Instead of writing the full prompt every time, pass only values (like type, year, and language) → the template replaces placeholders → AI gives a response.

## API Setup

- Create a new **POST API** → /api/recommend.
- Accepts three inputs via `@RequestParam`:
  - `type` → movie type (Comedy, Horror, etc.)
  - `year` → preferred decade/year
  - `lang` → language preference

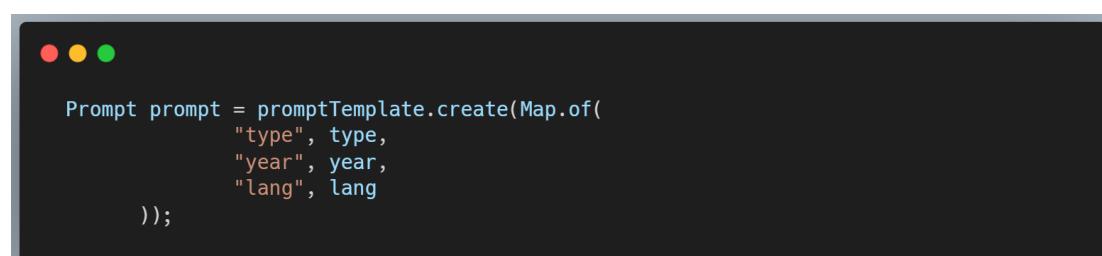
## Building the Prompt

- Use **PromptTemplate** with placeholders `{type}`, `{year}`, and `{lang}`.
- Example template:



```
I want to watch a {type} movie tonight with good rating,
looking for movies around {year}.
The language I'm looking for is {lang}.
Suggest one specific movie with:
1. Movie Name
2. Basic Plot
3. Cast
4. Length
5. IMDB rating
```

- When API is called, placeholders are replaced with actual values by using the below



```
Prompt prompt = promptTemplate.create(Map.of(
    "type", type,
    "year", year,
    "lang", lang
));
```

## Key Flow

- User sends request: POST /api/recommend?type=comedy&year=2010&lang=Telugu

- Controller picks values and creates a **PromptTemplate**.
- Replaces placeholders with real inputs.
- Sends final prompt to AI model using `chatClient.prompt(prompt).call().content()`.
- Returns AI's response in the required format.

## Code Implementation:

```

● ● ●

@RestController
public class OpenAIController {
    private ChatClient chatClient;

    public OpenAIController(OpenAiChatModel chatModel) {
        this.chatClient = ChatClient.create(chatModel);
    }

    @PostMapping("/api/recommend")
    public String recommend(@RequestParam String type, @RequestParam String year, @RequestParam String lang) {
        String temp = """
            I want to watch a {type} movie tonight with good rating,
            looking for movies around this year {year}.
            The language im looking for is {lang}.
            Suggest one specific movie and tell me the cast and length of the movie.

            response format should be:
            1. Movie Name
            2. basic plot
            3. cast
            4. length
            5. IMDB rating
            """;
        PromptTemplate promptTemplate = new PromptTemplate(temp);

        Prompt prompt = promptTemplate.create(Map.of(
            "type", type,
            "year", year,
            "lang", lang
        ));

        String response = chatClient
            .prompt(prompt)
            .call()
            .content();
        return response;
    }
}

```

## Key Takeaways:

- **PromptTemplate** makes prompts reusable and structured.
- Always map placeholders with actual values (`Map.of`).
- We can further improve the format by editing the template.