Integrating JSP/Servlets with ChatGPT API

Step-by-Step Guide

Presented by: Fakhrul Adli bin Mohd Zaki

FSKM, UMT

29/4/2025

Introduction

- What is JSP/Servlet?
- What is ChatGPT API?
- Why and how to integrate them?

What is ChatGPT API?

- The ChatGPT API allows developers to integrate OpenAI's ChatGPT models into their own applications, websites, or services.
- https://platform.openai.com/
- It enables natural language understanding and generation capabilities through programmatic access.
- Communication is done through HTTP requests (typically via JSON) to OpenAI's hosted servers.

Key Features of ChatGPT API

- Access to powerful AI models (e.g., GPT-3.5, GPT-4)
- Highly flexible — use for chatbots, summarization, translation, coding help, etc.
- Pay-as-you-go pricing based on usage
- Supports fine-tuning (for some models) and parameter control (e.g., temperature, max tokens)

How ChatGPT API Works

- Step 1: Send an HTTP POST request to the API endpoint with your input prompt.
- Step 2: The model processes the prompt on OpenAI’s server.
- Step 3: Receive the model's response as a JSON object.Requirements:API Key for authenticationEndpoint: https://api.openai.com/v1/chat/completions

Tools and Requirements

- Java JDK (8 or later)
- Apache Tomcat Server
- Gson Library for JSON
- OpenAI API Key
- Basic HTML/CSS (Bootstrap Optional)

Project Structure

```
YourProject/
 |- WebContent/
 |    |- index.jsp
 |    |- result.jsp
 |- src/
 |    |- com/yourapp/ChatServlet.java
 |- WEB-INF/
 |    |- web.xml
```

Setup index.jsp

- Create a simple form:
- - Input for ingredients
- - Dropdown for dietary preferences
- - Dropdown for meal types
- Submit to /chat endpoint

Create ChatServlet.java

- Extend HttpServlet
- Override doPost()
- Build prompt based on user input
- Call getChatGPTReply() method

Calling ChatGPT API

- Use HttpURLConnection
- Set headers:
  - Authorization: Bearer {API_KEY}
  - Content-Type: application/json
- POST JSON payload with model and messages

Parsing the Response

- Read API response
- Use Gson to parse JSON
- Extract the message content

Forwarding to result.jsp

- Store the ChatGPT reply in request attribute
- Forward using RequestDispatcher
- Display the suggestion nicely in Bootstrap card

web.xml Configuration

- <servlet>
- <servlet-name>ChatServlet</servlet-name>
- <servlet-class>com.yourapp.ChatServlet</servlet-class>
- </servlet>
- <servlet-mapping>
- <servlet-name>ChatServlet</servlet-name>
- <url-pattern>/chat</url-pattern>
- </servlet-mapping>

Key Libraries and Imports

- Java Networking
- Java Servlet API
- Gson for JSON

Important Tips

- Always protect your OpenAI API Key
- Validate user input
- Handle HTTP errors gracefully
- Keep UI simple and clean

Demo Flow

- 1. User fills form on index.jsp
- 2. Servlet processes input and calls ChatGPT
- 3. Response shown on result.jsp

Conclusion

- You have learned:
  - Building a JSP/Servlet project
  - Integrating with external APIs
  - Handling JSON in Java
- Start building smarter apps!

Q&A

- Any Questions?