

# 中间件技术 实验3

22920212204392 黄勛、22920212204385 胡翼翔、22920212204492 张靖源

## 1 实验目的

- 了解WEB Service基本概念
- 了解典型的Web Service服务
- 探讨大模型用于类似即时通讯任务的方式和方法

## 2 实验环境

系统：Windows 10

软件：

- VSCode
- Winsock
- Java with Springboot
- Activemq

## 3 实验步骤

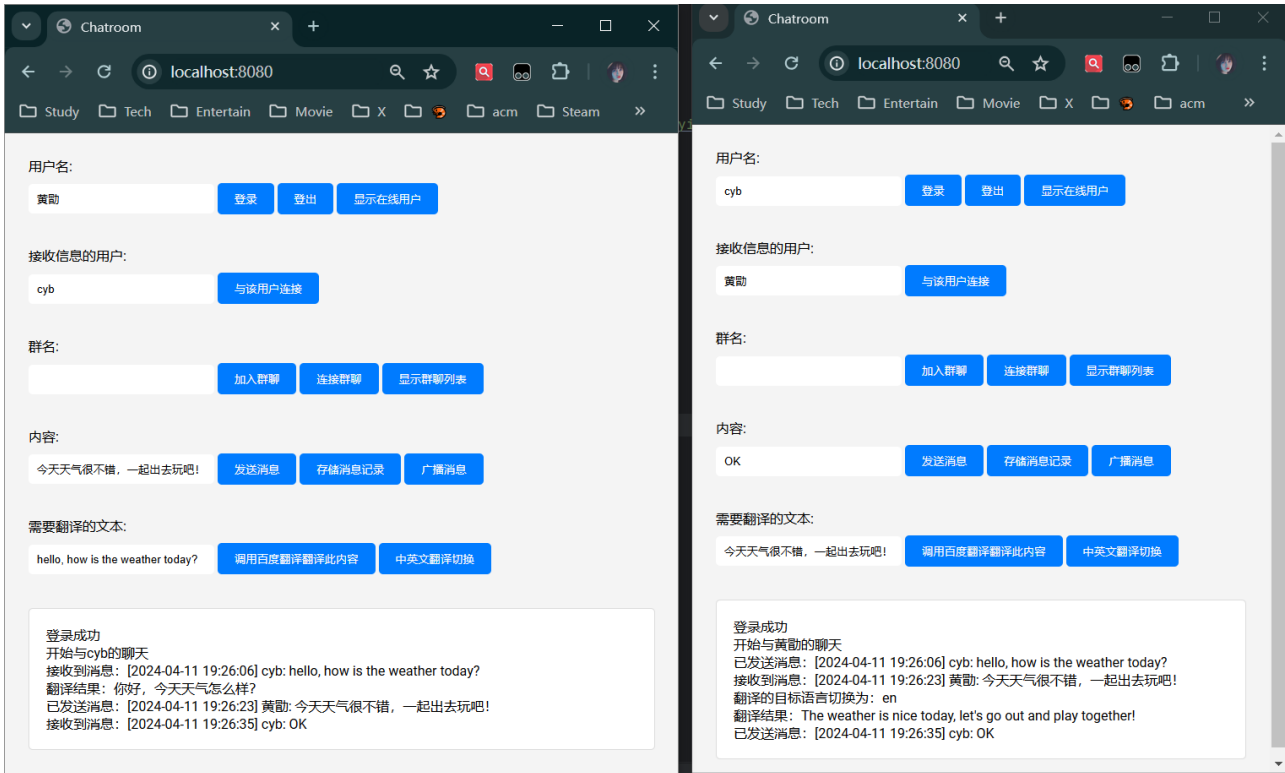
实现了**选择 1：利用谷歌翻译（百度翻译）实现聊天软件的边聊天边翻译功能。**

成果：采用springboot框架，java语言，中间件activemq完成即时通信软件，实现了群聊、私聊、消息存储，登陆登出等功能。

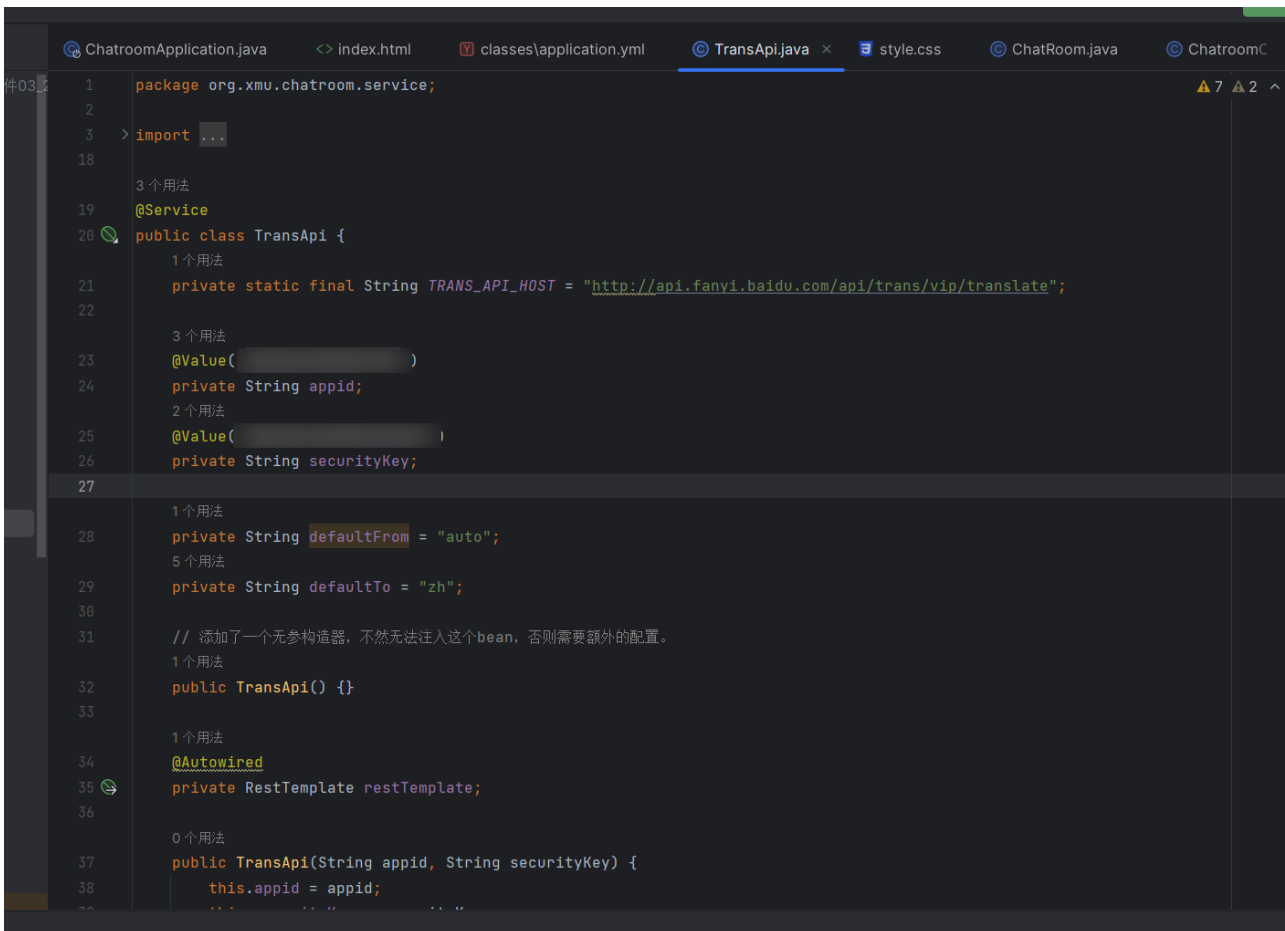
在先前聊天软件的基础上实现了翻译功能。

如下图，黄勛 和 cyb 为中国用户和英国用户，实现了调用翻译软件实现的边聊天边翻译。

界面展示：



不仅实现了翻译，也支持了多语种翻译和翻译文本的自动识别。代码大致如下：



```

51 > public String getTransResult(String query) { return getTransResult(query, defaultFrom, defaultTo); }
54
1个用法
55 public String getTransResult(String query, String from, String to) {
56     Map<String, String> params = buildParams(query, from, to);
57     MultiValueMap<String, String> requestParams = new LinkedMultiValueMap<>();
58     requestParams.setAll(params);
59     UriComponentsBuilder builder = UriComponentsBuilder.fromHttpUrl(TRANS_API_HOST);
60     URI uri = builder.queryParams(requestParams).build().encode().toUri(); // 这里不能进行 encode 了, 编码就错误了
61     System.out.println("uri: " + uri);
62     Object returnObj = restTemplate.getForObject(uri, Object.class);
63     Map entity1 = (Map)returnObj;
64     String transResult = entity1.get("trans_result").toString();
65     transResult = transResult.substring(2, transResult.length() - 2);
66     String result = "";
67     for(int i = 0, p = 0; i < transResult.length(); ++i){
68         if(transResult.charAt(i) == '='){
69             p++;
70         }
71         if(p == 2){
72             result = transResult.substring(beginIndex: i + 1);
73             break;
74         }
75     }
76     return result;
77 }
78

```

至此实验要求完成，成功实现功能。

## 4 考虑 1

调研如何应用大模型工具比如 chatGPT, new bing 等，如何用于改造你的聊天软件（受限于客观条件，可以探讨思路，现阶段不必实现）。

要将大型语言模型如 ChatGPT 整合到聊天软件中，需要遵循几个具体的步骤：Java 调用 OpenAI 接口，Java 调用实现 ChatGPT 聊天，可用于通过 Java 代码调用 GPT3.5 模型进行自然语言交互，实现智能聊天等功能。以下是如何进行这个过程的简要思路：

### 4.1 需求分析和规划

- **识别目标：**明确想通过整合ChatGPT实现什么目标。可能的目标包括提升用户体验、增加自动化程度、提供即时信息查询等。
- **用户研究：**了解用户的需求和期望，这将帮助确定模型的使用方式。
- **技术评估：**确定聊天软件的技术栈是否支持与ChatGPT等大型模型的集成。

### 4.2 设计与开发

- **接口设计：**设计ChatGPT与聊天软件之间的交互界面，确保用户能够流畅地与模型交互。
- **集成方案制定：**选择适合的API或SDK来集成ChatGPT。例如，使用OpenAI提供的API进行接口调用。
- **开发与测试：**开发必要的集成代码，并在沙盒环境中进行测试，确保系统稳定且响应迅速。

使用OpenAI API进行调用主要包括以下步骤：

要使用Java调用OpenAI API，你需要遵循以下步骤：

### 4.2.1 获取API密钥

- 在OpenAI官网注册并获取API密钥。

### 4.2.2 添加依赖

- 项目使用Maven，添加如下依赖到 `pom.xml` 文件中来进行HTTP请求（使用OkHttp客户端）：

```
<dependency>
  <groupId>com.squareup.okhttp3</groupId>
  <artifactId>okhttp</artifactId>
  <version>4.9.0</version>
</dependency>
```

### 4.2.3 3. 设置认证和执行请求

- 使用OkHttp创建请求，并在请求中加入你的API密钥来进行认证。然后，调用相应的OpenAI API端点。
- 下面是一个使用Java和OkHttp调用OpenAI API的例子：

```
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.MediaType;
import okhttp3.Response;

public class OpenAiExample {
    private static final String API_KEY = "your-api-key";
    private static final String API_URL =
        "https://api.openai.com/v1/engines/gpt-4/completions";

    public static void main(String[] args) {
        OkHttpClient client = new OkHttpClient();
        String jsonBody = "{\"prompt\": \"Translate the following English text to French: 'Hello, how are you?'\", \"max_tokens\": 60}";

        RequestBody body = RequestBody.create(
            jsonBody,
            MediaType.get("application/json; charset=utf-8")
        );

        Request request = new Request.Builder()
            .url(API_URL)
            .post(body)
            .addHeader("Authorization", "Bearer " + API_KEY)
            .build();

        try (Response response = client.newCall(request).execute()) {
```

```
        if (!response.isSuccessful()) throw new
RuntimeException("Unexpected code " + response);

        System.out.println(response.body().string());
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
```

#### 4.2.4 解析响应

- 处理并解析API返回的响应。将响应的JSON字符串转换为Java对象以便更方便地处理。

#### 4.2.5 异常处理和日志

- 确保代码中包含异常处理逻辑来捕获并处理可能发生的错误，如网络问题或API限制问题。
- 记录适当的日志以监控API调用情况。

### 4.3 安全和合规性

- **数据隐私**：确保处理数据时遵守GDPR、CCPA等数据保护法规。
- **安全措施**：实施加密、访问控制和审核日志等安全措施来保护用户数据。

### 4.4 部署和测试

- **逐步部署**：通过在小范围内部署开始，逐步扩大到所有用户，以监控性能和收集反馈。
- **性能监控**：实时监控系统性能，确保服务稳定且响应时间短。

### 4.5 用户培训和支持

- **用户指南**：提供用户指南和FAQ，帮助用户了解如何与ChatGPT交互。
- **客户支持**：设立客户支持渠道，解决用户在使用过程中遇到的问题。

### 4.6 持续评估和优化

- **收集反馈**：定期收集用户反馈，了解他们的体验和需求。
- **优化模型**：根据反馈调整和优化ChatGPT的使用，包括训练数据、交互方式和功能集成。

### 4.7 实施细节

- **API调用限制**：注意OpenAI的API可能有调用次数和频率限制，确保聊天软件的使用模式符合这些限制。
- **自定义训练**：如果需要，可以考虑对ChatGPT进行进一步的定制或微调，以更好地满足特定的业务需求。

通过遵循这些步骤，你可以有效地将ChatGPT集成到聊天软件中，提升用户体验，同时保证系统的稳定性和安全性。

## 5 实验总结

在本次实验中，我们小组通过编写相关代码，成功地利用中间件相关技术实现了实验的要求，体会到了中间件技术的方便。

## 6 实验分工

排序如下：

- 1 黄勛：activemq、后端翻译功能
- 2 胡翼翔、张靖源：前端代码、功能测试