

第18组大作业报告

1. 项目主题

基于Java Socket API搭建简单的HTTP客户端和服务器端程序

2. 项目简介

本项目基于 **Java Socket API (BIO)** 实现一个简易 HTTP 服务器和客户端，支持 GET、POST、302 重定向、304 缓存验证等。

3. 功能列表

- HTTP 服务器端 (BIO)
- 支持GET和POST请求
- 服务器、客户端均支持状态码：200、301、302、304、404、405、500，客户端可根据不同状态码做出响应
- 实现长连接 keep-alive
- MIME 类型支持text/plain、json、text/html、png、Location
- 实现注册，登录功能,数据存在内存中
- HTTP客户端可以发送请求报文、呈现响应报文,实现了GUI

4. 状态码测试说明

1. 200: 正常
2. 301: 永久重定向，如/moved 是一个示例接口，勾选“自动跟随重定向”，会展示login界面（根目录），不勾选就会出现301
3. 302: GET:/login 选择不自动重定向，可以返回302
4. 304: 使用缓存。示例：如果index.html内容不发生变化，就会不刷新index.html，出现304；如果index.html 内容发生变化，就会刷新显示。
5. 401: 登录错误/POST:/login 401
6. 404: 不合法请求，随便一个不正确的指令，示例：POST:register没有用户名和密码，会认为是未知的请求
7. 405: GET:/register 会报错405，因为register不允许GET，只支持POST请求
8. 500: 输入"/bug",弹出网络错误，因为在router里面写了这个命令，会故意抛出IOException，显示“Internal com Error”
9. 不开服务器，直接运行客户端：出现网络错误

5. 任务分配：

- A: 郭佳荣
- B: 崔可喻
- C: 程心妍
- D: 熊雅琪

角色	负责内容	产出文件
A: 服务器基础 & I/O框架	建立ServerSocket、接收连接、读取请求、实现长连接机制	com.server.HttpServer.java com.server.ConnectionHandler.java
B: HTTP协议解析 & 响应构造	解析请求行/头/体；构造响应行/头；处理MIME & 状态码 (200/301/302/304/404/405/500)	com.http.HttpRequestParser.java com.http.HttpResponse.java com.http.MimeType.java
C: 业务接口 (注册/登录) 与路由	HashMap存用户；实现/register /login POST处理；实现重定向；路由路径分发	com.http.Router.java com.model.UserService.java com.controller.UserController.java
D: 客户端 + 演示	用Socket手写GET/POST请求，打印响应，支持302自动重定向；编写最终报告 + 演示流程	Client.HttpClient.java

6. 运行说明

- 先运行 `com.Main` 的 `main()` 方法，启动服务器，监听8080端口
- 再运行 `client.Httpclient` 的 `main()` 方法，启动客户端GUI
- 在客户端页面可以实现UI和命令的自行输入设置，并查看状态码响应情况
- 同时支持在浏览器中进行注册和登录操作，在浏览器中输入 "<http://localhost:8080>" 进入网页
- 服务器端控制台会打印请求和响应日志，客户端GUI会显示响应

7. 注意事项

- 服务器端和客户端需要分别运行在不同的进程中
- 确保8080端口未被占用
- 如果需要测试不同的状态码，可以通过修改请求路径或参数来触发相应的处理逻辑

8. 项目结构

```

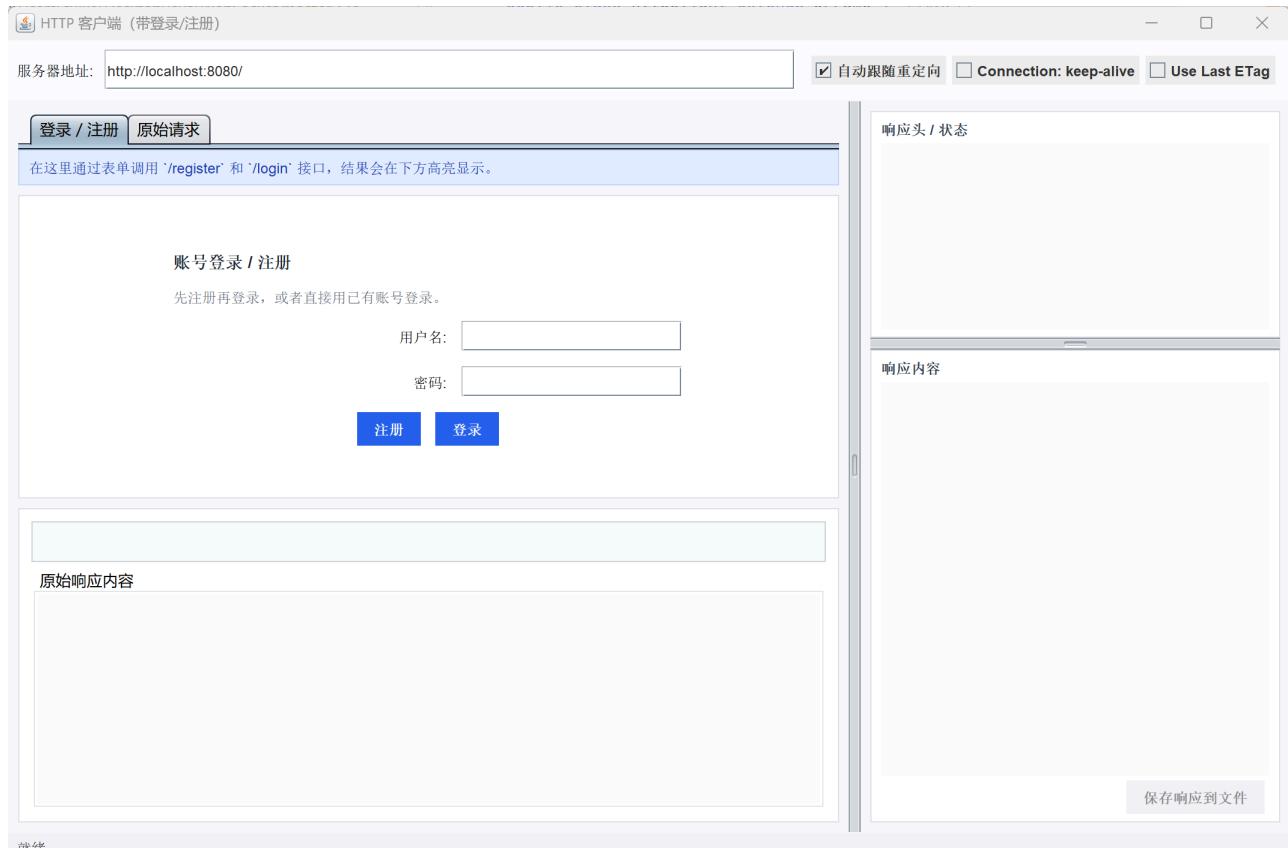
1  src
2   └── client
3     └── HttpClient.java          # 简单的 HTTP 客户端，用于测试 GET/POST、重定向等功能
4
5   └── com
6     ├── controller
7     │   └── UserController.java    # 处理 /register 和 /login 的业务逻辑
8
9     └── http
10    ├── HttpRequestParser.java    # 解析原始 HTTP 报文 → 得到 HttpRequest 对象
11    ├── HttpResponse.java         # 构造 HTTP 响应报文（状态行、头部、响应体）
12    ├── Router.java               # 根据路径选择交给哪个 Controller 或静态文件处理
13    └── MimeType.java             # 根据文件扩展名返回 MIME 类型（text/html,image/png等）
14
15   └── model

```

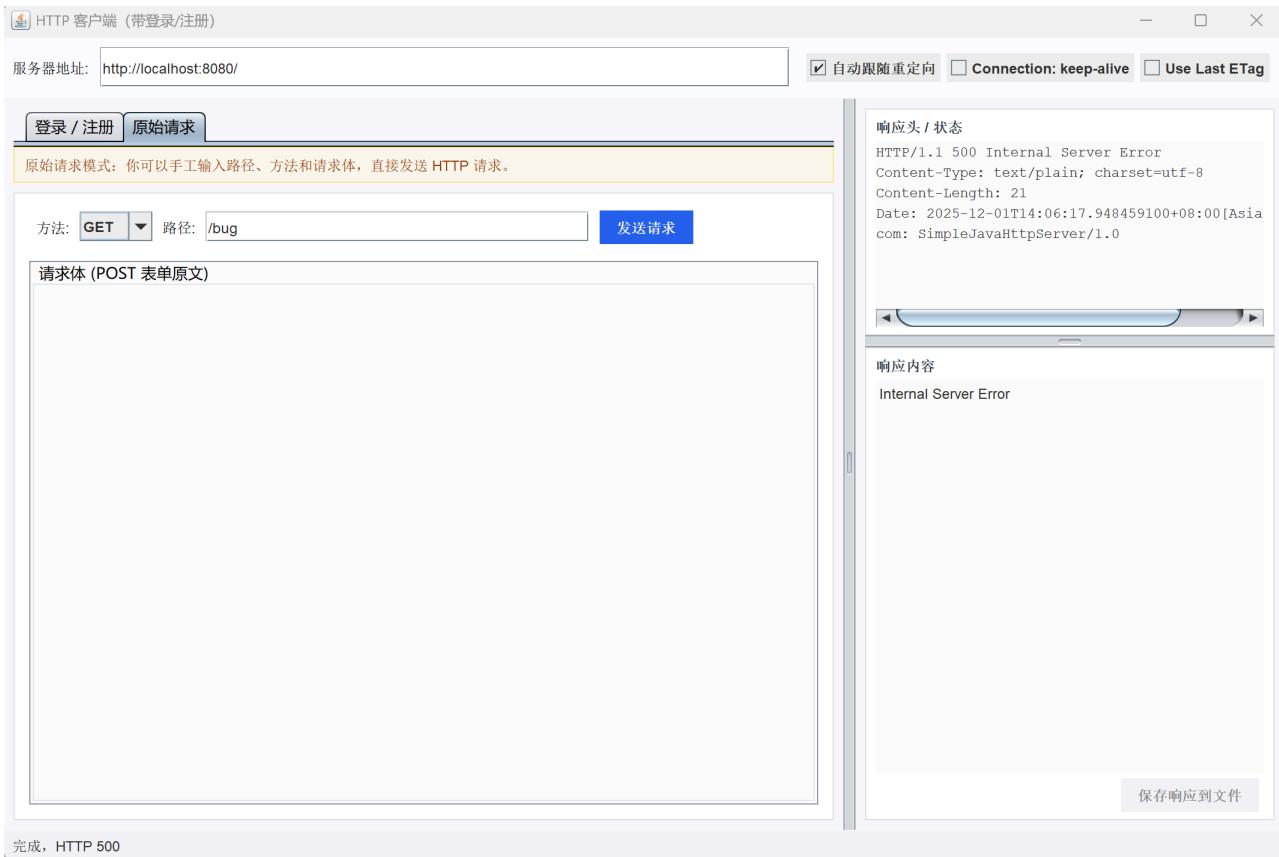
```
16 |     └── UserService.java          # 处理用户数据（注册/登录），数据存储在内存中
17 |
18 |     └── server
19 |         ├── ConnectionHandler.java # 一个线程处理一个 Socket 连接（BIO），实现 Keep-Alive
20 |         ├── HttpServer.java       # 启动服务器、监听端口、等待客户端连接
21 |         └── StaticFileHandler.java # 处理静态文件请求（如 HTML/CSS/JS）
22 |
23 |     └── Main.java                # 程序入口，启动 HttpServer
```

9. 运行截图

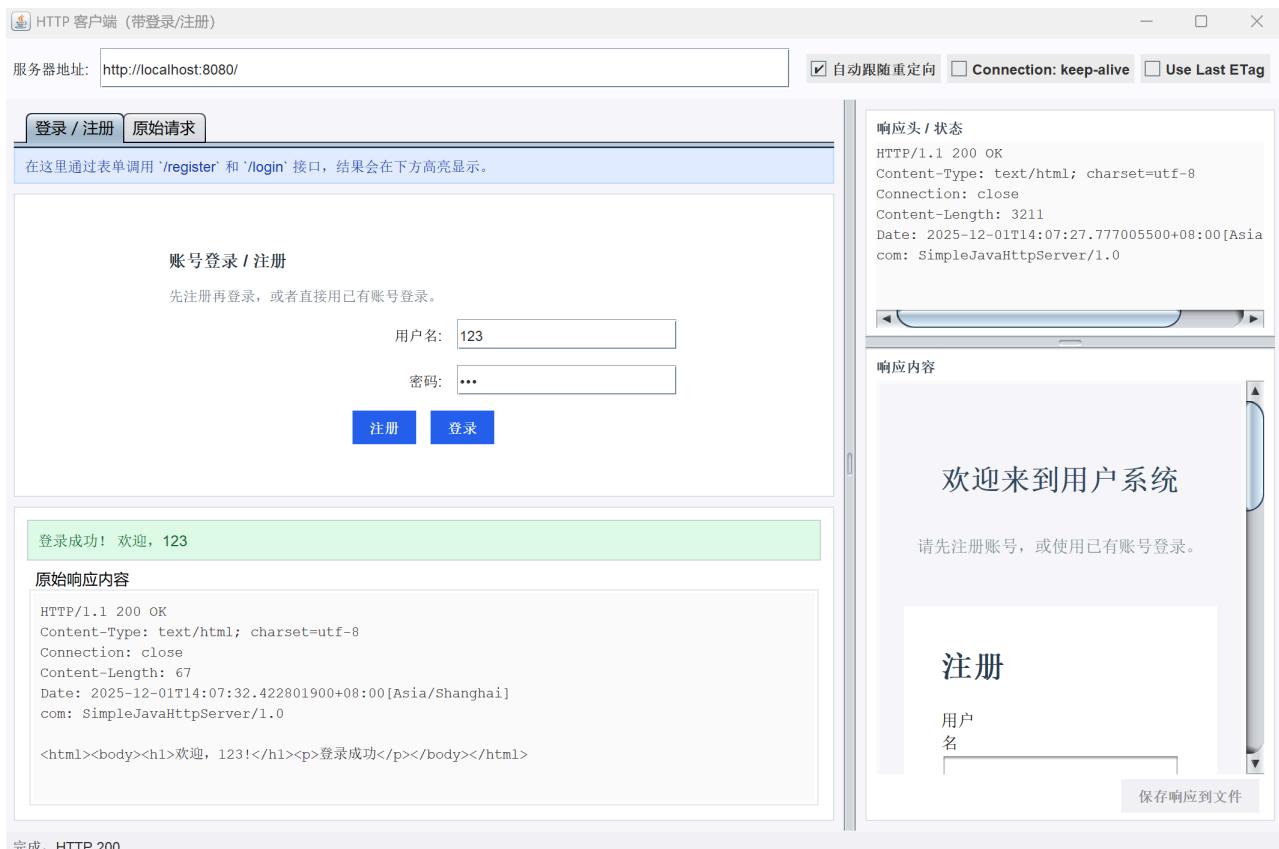
UI交互注册登录页面：



UI路径与参数输入页面：



示例：登录成功页面



10. 运行环境

- java: JDK21