

为了确保顺利完成项目，你可以按以下顺序逐步实现各项功能。这样可以帮助你从简单到复杂，逐步增加功能，并确保每个功能块在实现后可以进行独立测试。

建议实现顺序

基础通信搭建



- **实现服务端和客户端的 Socket 连接：**在服务端创建 `ServerSocket`，客户端连接到服务端，确保基本的连接是可行的。
- **简单的消息传递：**在客户端发送一条消息，服务端接收并返回确认消息。这样可以确认基本的 Socket 通信正常工作。

请求报文和响应报文的格式



- **构建基本的 HTTP 请求报文：**在客户端实现构建简单的 HTTP 请求报文（如 `GET` 请求），并通过 Socket 发送到服务端。
- **服务端解析请求报文：**在服务端实现解析客户端发来的请求，提取出请求方法、路径等信息。

响应报文的构建



- **构建基本的 HTTP 响应报文：**在服务端根据解析的请求信息构建响应报文，并返回给客户端。可以先实现 `200 OK` 的简单响应。
- **客户端解析响应报文：**客户端接收并解析服务端返回的响应，显示响应的状态码和内容。

添加常见状态码支持

- **实现 404 状态码：**当客户端请求的路径不存在时，服务端返回 `404 Not Found` 响应。
- **实现 500 状态码：**当服务端发生错误时返回 `500 Internal Server Error`。
- **添加其他状态码：**根据项目需求逐步添加 `301`、`302`、`304` 等状态码，处理重定向和缓存逻辑。

支持 GET 和 POST 方法

- **实现 GET 请求处理：**在服务端实现对 GET 请求的响应，提供静态资源（如 HTML 文件）的返回。
- **实现 POST 请求处理：**在客户端构造 POST 请求，服务端解析并处理 POST 数据（如表单提交的数据）。

连接管理和并发处理

- **支持多个客户端连接：**使用多线程处理服务端，允许同时处理多个客户端连接，确保服务端能够处理并发请求。

- **连接保持与关闭**：实现连接的保持（`Connection: keep-alive`）和关闭，确保资源合理使用。

MIME 类型支持

- **添加 MIME 类型响应头**：在服务端根据请求的文件类型设置正确的 `Content-Type`（如 `text/html`、`application/json`），确保客户端能够正确解析返回的内容。

用户注册与登录功能

- **实现内存中的用户数据管理**：在服务端实现简单的用户注册和登录逻辑，将用户信息存储在内存中。
- **实现注册和登录接口**：客户端可以通过 POST 请求注册新用户或登录已有用户，服务端处理并返回响应。

测试和优化

- **测试不同场景**：使用命令行、Postman 或其他工具测试各种请求和状态码，确保功能稳定。
- **错误处理和日志**：添加适当的错误处理和日志输出，以便调试和跟踪问题。

总结

通过以上顺序，你可以先确保基本功能（如 Socket 通信和报文解析）正常，再逐步扩展到更复杂的功能（如多线程和用户管理）。每一步完成后，可以进行测试，以确保新功能不会影响已有功能的稳定性。