

C++网络通信利器：使用cpp-http库实现轻松HTTP请求，GET、POST应有尽有！

作者：微信公众号：【架构师老卢】

1-9 8:52

~ 141



概述：在C++中发起HTTP请求通常借助网络库或HTTP客户端库，例如cpp-httplib。通过选择适用的库、构建请求、发送请求、处理响应，实现与服务器的通信。

在C++中，使用HTTP请求需要通过网络库或HTTP客户端库，如Curl或cpp-httplib。这些库提供了API来构建和发送HTTP请求，并处理响应。

方法

- 选择合适的库：**
 - 选择网络库或HTTP客户端库，如Curl或cpp-httplib。
- 构建HTTP请求：**
 - 使用库提供的API构建HTTP请求，包括URL、方法（GET、POST等）、头部、请求体等。
- 发送请求：**
 - 调用库的函数发送HTTP请求。
- 处理响应：**
 - 处理库返回的HTTP响应，包括状态码、头部、响应体等。

步骤

- 引入库：**
 - 引入网络库或HTTP客户端库，如Curl或cpp-httplib。
- 构建HTTP请求：**
 - 使用库提供的函数构建HTTP请求。
- 发送请求：**
 - 调用库的函数发送HTTP请求。
- 处理响应：**
 - 处理库返回的HTTP响应。

实例源代码

以下是使用cpp-httplib库的简单示例，执行GET和POST请求：

```
1 #include <iostream>
2 #include "http.h"
3
4 int main() {
5     // 创建http客户端
6     http::Client client("http://jsonplaceholder.typicode.com");
7
8     // 发送GET请求
9     auto get_res = client.Get("/posts/1");
10    if (get_res && get_res->status == 200) {
11        std::cout << "GET请求成功: " << get_res->body << std::endl;
12    } else {
13        std::cout << "GET请求失败" << std::endl;
14    }
15
16    // 发送POST请求
17    auto post_res = client.Post("/posts", R({"title": "foo", "body": "bar", "userId": 1}), "application/json");
18    if (post_res && post_res->status == 201) {
19        std::cout << "POST请求成功: " << post_res->body << std::endl;
20    } else {
21        std::cout << "POST请求失败" << std::endl;
22    }
23
24    return 0;
25 }
```

注意事项

- 选择合适的库：**根据需求选择适用于项目的网络库或HTTP客户端库。
- 异常处理：**处理可能的异常情况，如网络连接失败或服务器错误。

3. **注意安全性**: 避免在请求体中硬编码敏感信息, 使用安全的方式传递参数。

使用C++进行HTTP请求通常需要引入专用的网络库或HTTP客户端库。在选择合适的库后, 通过构建HTTP请求和处理响应, 可以轻松地与服务端进行通信。务必注意异常处理和安全性, 确保代码的健壮性和安全性。