

2020/5/12

# Websocket

---

## 方法1:

- 放弃开源库。重写handler、response等。
- 握手
  1. 设置入口 `127.0.0.1 6666`
  2. tcp socket响应收到的 **GET** 回复地址 `127.0.0.1`
    1. 浏览器最好使用80或443端口，否则可能会有防火墙block
    2. 地址可以设置uri，引导至不同的服务
    3. 握手前可以使用http的状态码，握手后需要定义一套不同的。
  3. 读取获得的header，去掉最后的 `/r/n`；把key-value一一对应存起来；读取key，解析header，发送回websocket的返回包
  4. [TODO]
- 回复
  1. 每个header后都有 `\r\n`
  2. [TODO]

## 方法2:

- 利用开源库websocketpp写一个单独用来处理websocket信息的server
  - 注：实际上websocketpp中还可以判断普通http请求，并进行简单的处理。
- 代码文件说明：
  - tcpserver.cpp：tcpserver函数实现
  - tcpserver.h：tcpserver头文件
  - runtcpserver.cpp：主文件（尝试在同一个main函数里判断连接类型）
- 测试编译make

基础逻辑:

- on\_open
- on\_message
- on\_error
- on\_close

1. 将之前登录记录的表全部清除。（frame payload作用？这两行没懂。）

- [TODO]<https://www.zaphoyd.com/websocketpp/manual/reference/logging>
- 2. asio初始化（如果使用asio的话，否则可以无视）
- 3. set\_balabala\_handler(...) balabala填入操作，比如open，fail，message，然后对handler做出相应操作。  
函数中参数设为bind()
- 4. listen (**port**)
  - websocketpp中默认端口为9002
  - See the asio transport component documentation for information on customizing this behavior.
- 5. start\_accept() 开始循环accept client并立即将其排列到队列中
- 6. run() asio内部开始创建循环，直到server被停止。

```
编译命令行： g++ -std=c++11 -g -o test websocketserver.cpp -I./ -I
/Users/mac/Documents/websocketpp-master/ -
I/usr/local/Cellar/boost/1.72.0_2/include/ -
L/usr/local/Cellar/boost/1.72.0_2/lib/
```

## 已完成的任务

1. 完成server对websocket的连接及发送接受信息。
2. 通过对server参数的输入判断接收tcp连接还是websocket连接

## 遇到的困难

1. 连接websocket客户端成功 但是只能接受一次消息】
  - on\_message中逻辑写错了，停留在send中，无法接受on\_message。
2. 如何在同一个server中完成同时对两种连接的判断？
  - 暂时用了参数判定，运行后只能接收一种请求。

## 补充：OS系统中的gdb调试

1. g++ -std=c++11 -g -o test runtcpserver.cpp -I./ -I /Users/mac/Documents/websocketpp-master/ -I/usr/local/Cellar/boost/1.72.0\_2/include/ -L/usr/local/Cellar/boost/1.72.0\_2/lib/
2. sudo gdb test
3. break linenumber
4. start
5. c（跳转到断点）
6. n（下一步）

