

2022《计算机网络》实验报告：Http 客户端和服务端的模拟

（Socket 编程）

- 小组成员 分工
- 张涵景 客户端
- 马培森 客户端
- 曾轲 客户端
- 李博轩 服务端
- 刘永全 服务端

1.运行说明

需要先启动类 HTTPSever 中的 main 方法，再启动 ClientMain 中的 main 方法。然后根据 ClientMain 的提示，控制台输入命令。选择长/短链接，然后会向服务器发送一次注册信息。如果是短链接，发送一次后就会自动释放，如果是长连接，则可以不断发送 post 或 get 请求，需要手动输入 stop 停止。

其中，post 方法客户端从 Client/Resources 发送，服务端接受在 Server/Resources 里面接受；get 方法服务端从 Server/Resources 发送，post 方法在 Client/Cache 里面接收。对于文本类型的内容，会打印在控制台中，例如 html 格式。对于非文本类型的内容，将其转化为 byte，再转化为相对应的文件保存。另外，由于报文最大大小的限制，传输文件不能超过 64kb。文件以外的报文信息，在控制台都可以看到。

2.界面展示

2.1 POST 展示

登录注册功能(通过 post 实现)

```

Do you want to keep alive?(Y/N)
Y
====>>> START LINE <<<====
You are trying to connect to a socket server.
You will be registered if you try to log in for the first time.

Your username:
1234
Your password:
1234

Date:Wed Dec 21 20:02:35 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:text/html
Content-Length:136

<!DOCTYPE html><html lang="en"><head>    <meta charset="UTF-8">    <title>Register and login success </title></head><body></body></html>
get, post or stop?
post
login or upload?
login
You are trying to change your account. Please input again.

Your username:
1234
Your password:
12345
Date:Wed Dec 21 20:02:52 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:text/html
Content-Length:152

<!DOCTYPE html><html lang="en"><head>    <meta charset="UTF-8">    <title>Fail to Login, invalid username or password</title></head><body></body></html>
get, post or stop?

```

Post200 状态码

```

get, post or stop?
post
login or upload?
upload
the url of the file you want to upload?(e.g. uploadSuccess.html)
uploadSuccess.html
target url?(e.g. ./Resources/uploadSuccess.html)
./Resources/uploadSuccess.html
Date:Wed Dec 28 22:53:42 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:text/html
Content-length:153

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>File successfully uploaded</title>
</head>
<body>

</body>
</html>

```

Post301 状态码 永久重定向

```
get, post or stop?
post
login or upload?
upload
the url of the file you want to upload?(e.g. uploadSuccess.html)
301origin.html
target url?(e.g. ./Resources/uploadSuccess.html)
./Resources/301origin.html
301 Permanent Redirecting To: ./Resources/301dest.html
```

```
get, post or stop?
post
login or upload?
upload
the url of the file you want to upload?(e.g. uploadSuccess.html)
301dest.html
target url?(e.g. ./Resources/uploadSuccess.html)
./Resources/301origin.html
Date:Wed Dec 28 23:08:55 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:application/octet-stream
Content-length:134
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>301dest</title>
</head>
<body>

</body>
</html>
```

Post302 状态码 临时重定向

```
get, post or stop?
post
login or upload?
upload
the url of the file you want to upload?(e.g. uploadSuccess.html)
302origin.html
target url?(e.g. ../Resources/uploadSuccess.html)
../Resources/302origin.html
302 Temporary Redirecting To: ../Resources/302dest.html
```

```
get, post or stop?
post
login or upload?
uupload
====>>> WARNING <<<<====
This is an invalid input.
Please input again.
upload
the url of the file you want to upload?(e.g. uploadSuccess.html)
302dest.html
target url?(e.g. ../Resources/uploadSuccess.html)
../Resources/302dest.html
Date:Wed Dec 28 23:10:54 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:text/html
Content-length:134
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>302dest</title>
</head>
<body>

</body>
</html>
```

Post 多种 MIME 支持

客户端上传照片

```
get, post or stop?
post
login or upload?
upload
the url of the file you want to upload?(e.g. uploadSuccess.html)
100.jpeg
target url?(e.g. ./Resources/uploadSuccess.html)
./Resources/newphoto.jpeg
Date:Wed Dec 28 23:24:16 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:image/jpeg
Content-length:5972
```

客户端上传 zip 文件

```
get, post or stop?
post
login or upload?
upload
the url of the file you want to upload?(e.g. uploadSuccess.html)
uploadSuccess.zip
target url?(e.g. ./Resources/uploadSuccess.html)
./Resources/uploadSuccess.zip
Date:Wed Dec 21 20:16:09 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:application/zip
Content-length:413
```

2.2GET 功能展示

Get 200 状态码

```
get, post or stop?
get
url?(e.g. ./Resources/index.html)
./Resources/index.html
====>>> RECEIVING MESSAGE <<<====
HTTP/1.1 200 OK
Date:Thu Dec 29 00:01:52 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:text/html
Content-length:164
```

Get301 状态码

```
get, post or stop?
get
url?(e.g. ./Resources/index.html)
./Resources/301origin.html
====>>> RECEIVING MESSAGE <<<====
301 Redirecting to: ./Resources/301dest.html
Created new file " ./Client/Cache/301dest.html".
```

Get302 状态码

```
get, post or stop?
get
url?(e.g. ./Resources/index.html)
./Resources/302origin.html
====>>> RECEIVING MESSAGE <<<====
ffResources/302dest.html
302 Redirecting to: Resources/302dest.html
Created new file " ./Client/Cache/302dest.html".
```

Get304 状态码

```
get, post or stop?
get
url?(e.g. ./Resources/index.html)
./Resources/index.html
====>>> RECEIVING MESSAGE <<<====
304 Redirecting to: ./Resources/index.html
```

2.3 服务端

登陆注册功能的响应报文（状态码 200，具体成功与否体现在报文的实体主体）

```
HTTP/1.1 200 OK
Date:Thu Dec 29 14:18:31 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:text/html
Content-length:154
```

Post200 状态码的响应报文

```
HTTP/1.1 200 OK
Date:Thu Dec 29 14:28:30 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:text/html
Content-length:166
```

Post301 状态码的响应报文

```
HTTP/1.1 301 Moved Permanently
Location: Resources/301dest.html
Date:Thu Dec 29 14:31:33 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:text/html
Content-length:134
```

Post302 状态码的响应报文


```
HTTP/1.1 302 Found
Location: Resources/302dest.html
Date:Thu Dec 29 14:32:48 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:text/html
Content-length:134
```

post 多种 MIME 支持服务端响应报文
照片

```
HTTP/1.1 200 OK
Date:Thu Dec 29 14:34:06 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:image/jpeg
Content-length:134
```

Zip 文件

```
HTTP/1.1 200 OK
Date:Thu Dec 29 14:37:26 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:application/zip
Content-length:194
```

Get200 状态码响应报文


```
HTTP/1.1 200 OK
Date:Thu Dec 29 14:38:42 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:text/html
Content-length:136
```

Get301 状态码响应报文

```
HTTP/1.1 301 Moved Permanently
Location: Resources/301dest.html
Date:Thu Dec 29 14:39:54 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:text/html
Content-length:134
```

Get302 状态码响应报文

```
HTTP/1.1 302 Found
Location: Resources/302dest.html
Date:Thu Dec 29 14:40:59 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:text/html
Content-length:134
```

Get304 状态码响应报文

```
HTTP/1.1 304 Not Modified
Date:Thu Dec 29 14:42:19 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:text/html
Content-length:130
```

404 状态码（未找到文件）响应报文

```
HTTP/1.1 404 NOT FOUND
Date:Thu Dec 29 15:02:41 CST 2022
Server:HOST Sever/0.0.0;charset=GBK
Content-type:text/html
Content-length:130
```

3.代码解析

3.1 代码说明

客户端代码说明：Client/Cache 存放 get 获得的文件；Client/Methods 利用策略模式，实现 get 和 post 方法；Client/Requestmessage 提供报文类，包括 line、head、body 三个部分；Client/Resources 存放可以 post 的资源；ClientMain 为 Main 方法；Connent 为连接的抽象类，Newclient 会调用 Connect 建立链接，Redirectlist 是重定向表格。

服务器端代码说明：RedirectList 是重定向表格，用于判断是否重定；UserService/RegisterAndLogin 处理注册和登；Request 为请求报文的数据；Handle 用于处理请求报文，生成状态码；Response 生成响应报文；Server/Resources 存放可以 get 的资源；HTTPServer 为 main 方法；ServerConnect 建立/中断连接；MIMETypes 判断 MIME 类型。

工具类代码说明：FileMaker 用于创建新的文件并写入；FileTable 保存文件的修改时间，判定 304 状态码；GetFile 读取文件；StreamReader 用于 stream 和 byte 之

间的转换。

测试类代码说明：FileMakerTest 用于测试 FileMaker 是否可以正常创建、修改各类文件，以支持 MIME；LoginTest 用于测试登录功能能否正常使用；VersatileTest 用于检查 Get、Post 方法及长短连接能否正常运行并可检查控制台输出是否正常。

3.2 个别功能简述

当用户首次使用客户端时，会强制要求其注册

NewClient 类中登录注册方法的实现

```
public NewClient(int port, String host, boolean isKeepAlive) throws IOException {
    // login in(temporarily true)
    connection = new Connect(host, port, isKeepAlive: true);
    connection.creat();
    // login message
    Login login = new Login();
    String infor = login.set();
    // register after login
    switchMode( method: "POST");
    requestMethod.sendRequest( url: "./Resources/index.html", new RequestBody(infor));
    connection.setKeepAlive(isKeepAlive);
}
```

Post 中 **sendRequest** 方法的实现

```
public void sendRequest(String url, RequestBody body) throws IOException {
    // 发送报文前先查找重定向表
    for (Map.Entry<String, String> entry : RedirectList.getRedirectList().entrySet()) {
        if (entry.getKey().equals(url)) {
            url = entry.getValue();
            break;
        }
    }

    // 报文通过byte数组，再转化为stream发送，以实现MIME中的多种文件格式的传输
    HTTPRequest request = assembleRequest(url, body);
    byte[] bytes = new byte[request.toString().getBytes().length + request.requestBody.getLength()];
    // head
    System.arraycopy(request.toString().getBytes(), srcPos: 0, bytes, destPos: 0,
        request.toString().getBytes().length);
    // body
    System.arraycopy(request.bodytoByte(), srcPos: 0, bytes, request.toString().getBytes().length,
        request.bodytoByte().length);
    // send
    connection.getSendStream().write(bytes);
    conductResponse(url);
}
}
```

重定向 200 301 302 的实现

在本次项目中，服务端维护了 RedicectList 来记录会发生重定向的资源以及其对应的重定向目的地，通过其 search 方法查询是否需要重定向

```
public class RedirectList {
    //只能有一个
    3 个用法
    private static RedirectList redirectList = null;

    1 个用法  211250125liu +1
    public static RedirectList getRedirectList(){
        if(RedirectList.redirectList == null){
            RedirectList.redirectList = new RedirectList();
        }
        return redirectList;
    }

    5 个用法
    public static HashMap<String,String> redirectLists = new HashMap<>();
    3 个用法
    public static HashMap<String,Integer> redirectType = new HashMap<>();
    1 个用法  211250125liu +1
    private RedirectList(){
        redirectLists.put("Resources/301origin.html", "Resources/301dest.html");
        redirectType.put("Resources/301origin.html", 301);
        redirectLists.put("Resources/302origin.html", "Resources/302dest.html");
        redirectType.put("Resources/302origin.html", 302);
    }

    2 个用法  micer +1
    public String search(String originURL){
        if(!redirectLists.containsKey(originURL)){
            return "";
        }
        return redirectLists.get(originURL);
    }
}
```



```

/**
 * 重新读取配置文件
 * @param path
 */
private void parseConfig(String path) {
    String []line;
    File file = new File(path);
    try (BufferedReader reader = new BufferedReader(new FileReader(file))) {
        String tempString;
        while ((tempString = reader.readLine()) != null) {
            line = tempString.split(regex: "\\s+");
            redirectLists.put(line[0], line[1]);
            redirectType.put(line[0], Integer.parseInt(line[2]));
        }
    } catch (IOException e) {
        System.out.println("配置文件读取失败");
    }
}
}

```

服务端 304 的实现

在本次实验中，当用户两次 get 一份相同的且未修改的资源的时候，服务器返回 304。为了满足这个条件，我们在服务器端引入了资源管理机制，具体来说为工具类 FileTable，定义如下：

```

public class FileTable {
    7 个用法
    HashMap<String, Long> files;
    7 个用法
    Lock lock;

    2 个用法 211250125liu
    public FileTable() {...}

    /** 从一个文件夹中读取全部文件，修改时间记为读取时间 */
    1 个用法 211250125liu
    public void initInAFolder(String path) {...}

    /** 修改一个文件 */
    3 个用法 211250125liu
    public void modify(String file) {...}

    /** 获取一个文件的修改时间，没出现过则返回 -1 */
    2 个用法 211250125liu
    public Long getModifiedTime(String file) {...}
}

```

使用一个 HashMap 来记录所有文件被修改或被获取的时间戳。在服务器初始化

时，使用一个 `FileTable` 维护所有资源被修改的时间戳；同时在每一次连接使用一个单独的 `FileTable` 维护这次连接中获取的文件以及获取的时间戳。比较两个 `FileTable` 即可判断是否满足 304 的返回条件。

```
statusCode = 200;
location = url;
// 304
// 文件修改时间
Long getTime = getFile.getModifiedTime(location);
Long modifyTime = modifiedFileTable.getModifiedTime(location);
//System.out.println(getTime + " " + modifyTime);
if (getTime > modifyTime) {
    statusCode = 304;
    location = NOT_MODIFIED_RES;
    request.setUrl(location);
} else {
    // 修改文件
    getFile.modify(location);
}
```

服务端 405 的实现

Handle 类中非 `get`、`post` 指令，那么状态码为 405

```
//不支持的指令
else {
    // 405
    statusCode = 405;
    location = METHOD_NOT_ALLOWED_RES;
    request.setUrl(location);
}
```

服务端 500 状态码的实现

使用 `isDown` 模拟服务器挂掉，默认设为 `false`


```
public void setIsDown(){
    isDown = true;
}
```

3 个用法 👤 Zhang

```
public void handle() {
    //模拟服务器挂掉
    if(isDown){
        statusCode = 500;
        location = SERVER_ERROR_RES;
        request.setUrl(location);
    }
}
```

4.备注

Git 仓库地址: <https://github.com/ZhangHJ11/Network2022Socket.git>

组长 qq: 1369061973