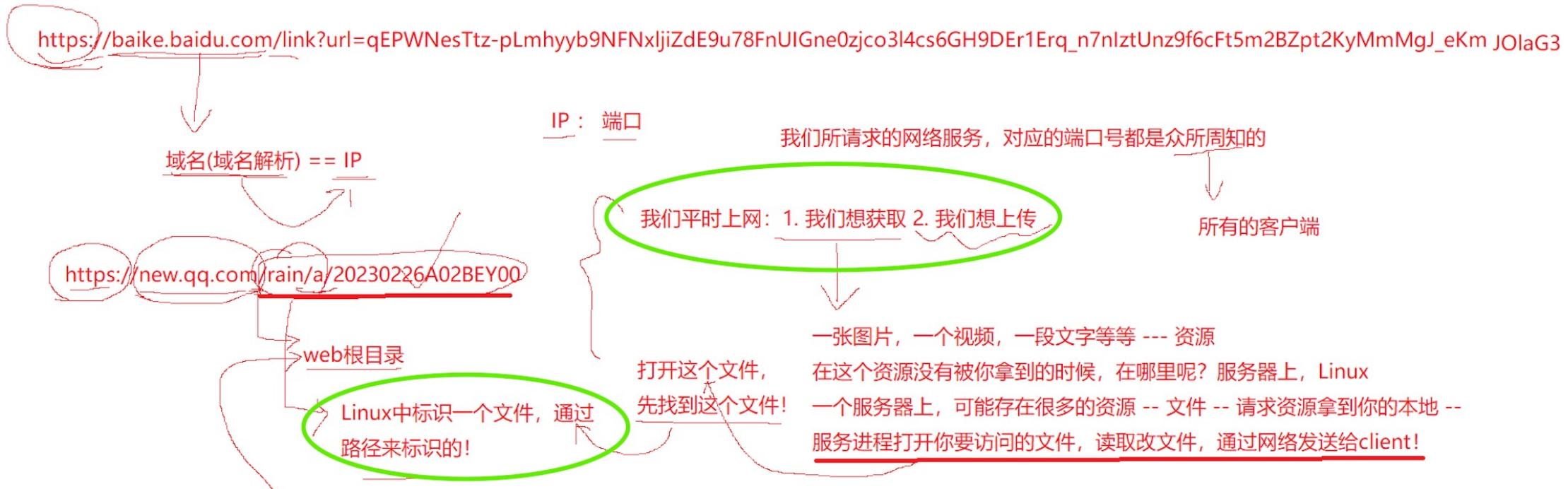
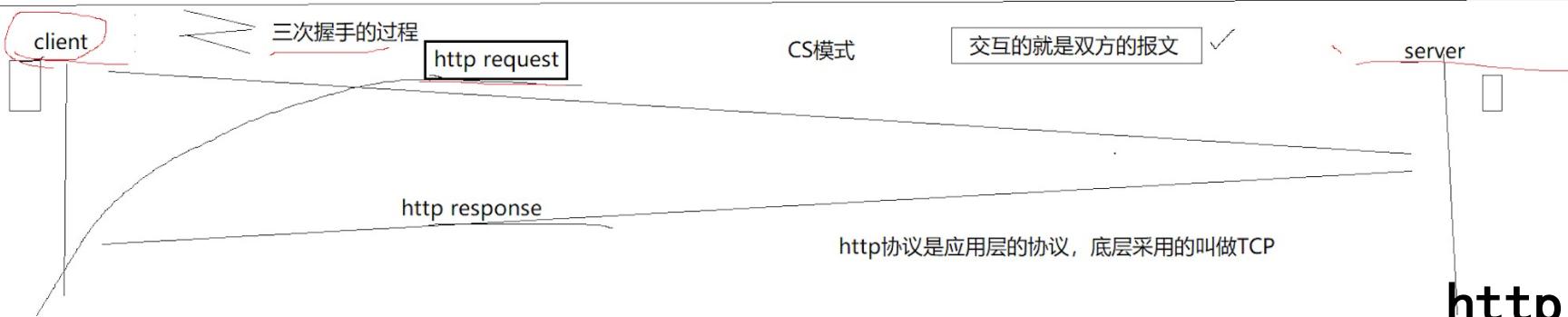


0226http

应用层：就是程序员基于socket接口之上编写的具体逻辑，做的很多工作，都是和文本处理有关的！-- 协议分析与处理！

http协议，一定会具有大量的文本分析和协议处理！





http是基于行的文本协议！

读到报头的时候，怎么确定正文长度呢？
我们说了
http里面有很多属性
其中一个就叫做
Content-Length !!

1. 快速构建http请求和响应的报文格式



单纯在报文角度，http可以是基于行的文本协议！

client->server: client用的是哪一个版本

server->client: server
用的是哪一个http版本



如何看待：
看成线性结构



Content-Length: 123

我们如何得知正文的大小呢？

我们平时的上网行为其实就两种

1. 从服务器拿下来资源数据 --- GET
2. 把客户端的数据提交到服务器 --- POST, GET

```
Connection closed by foreign host.  
o (base) [yufc@VM-12-12-centos:~/Core/BitCodeField/0226/http_demo]$ telnet www.baidu.com 80  
Trying 14.119.104.254...  
Connected to www.baidu.com.  
Escape character is '^]'.  
^]
```

```
telnet>  
GET / HTTP/1.1
```

通过GET方法，抓到了www.baidu.com的/，即它的首页

```
HTTP/1.1 200 OK  
Accept-Ranges: bytes  
Cache-Control: no-cache  
Connection: keep-alive  
Content-Length: 9508  
Content-Security-Policy: frame-ancestors 'self' https://chat.baidu.com https://fj-chat.baidu.com https://hba-chat.baidu.com https://hbe-chat.baidu.com https://njjsc-chat.baidu.com https://nj-chat.baidu.com https://hna-chat.baidu.com https://hnb-chat.baidu.com;  
Content-Type: text/html  
Date: Tue, 25 Apr 2023 15:25:29 GMT  
P3p: CP=" OTI DSP COR IVA OUR IND COM "  
P3p: CP=" OTI DSP COR IVA OUR IND COM "  
Pragma: no-cache  
Server: BWS/1.1  
Set-Cookie: BAIDUID=73880A52E64777AE8DD1AF9B1DF5AA69:FG=1; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com  
Set-Cookie: BDUID=73880A52E64777AE8DD1AF9B1DF5AA69; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/.
```

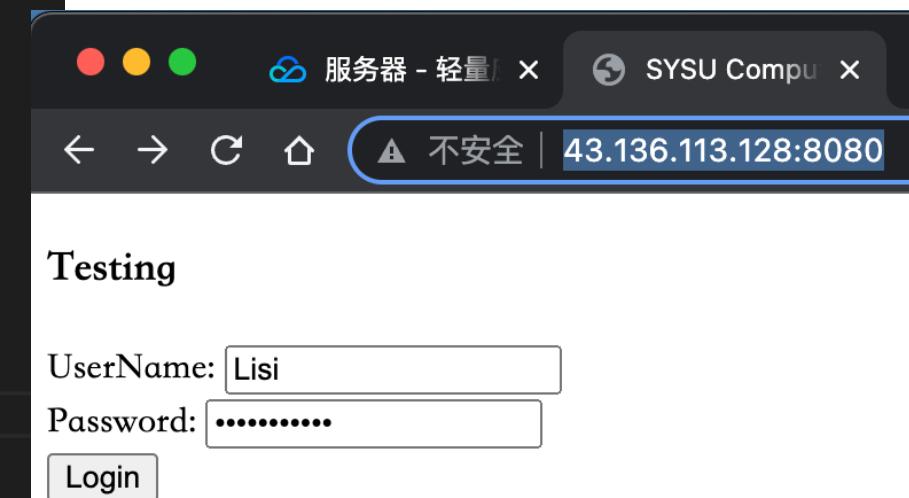
下面这一大坨就是 html

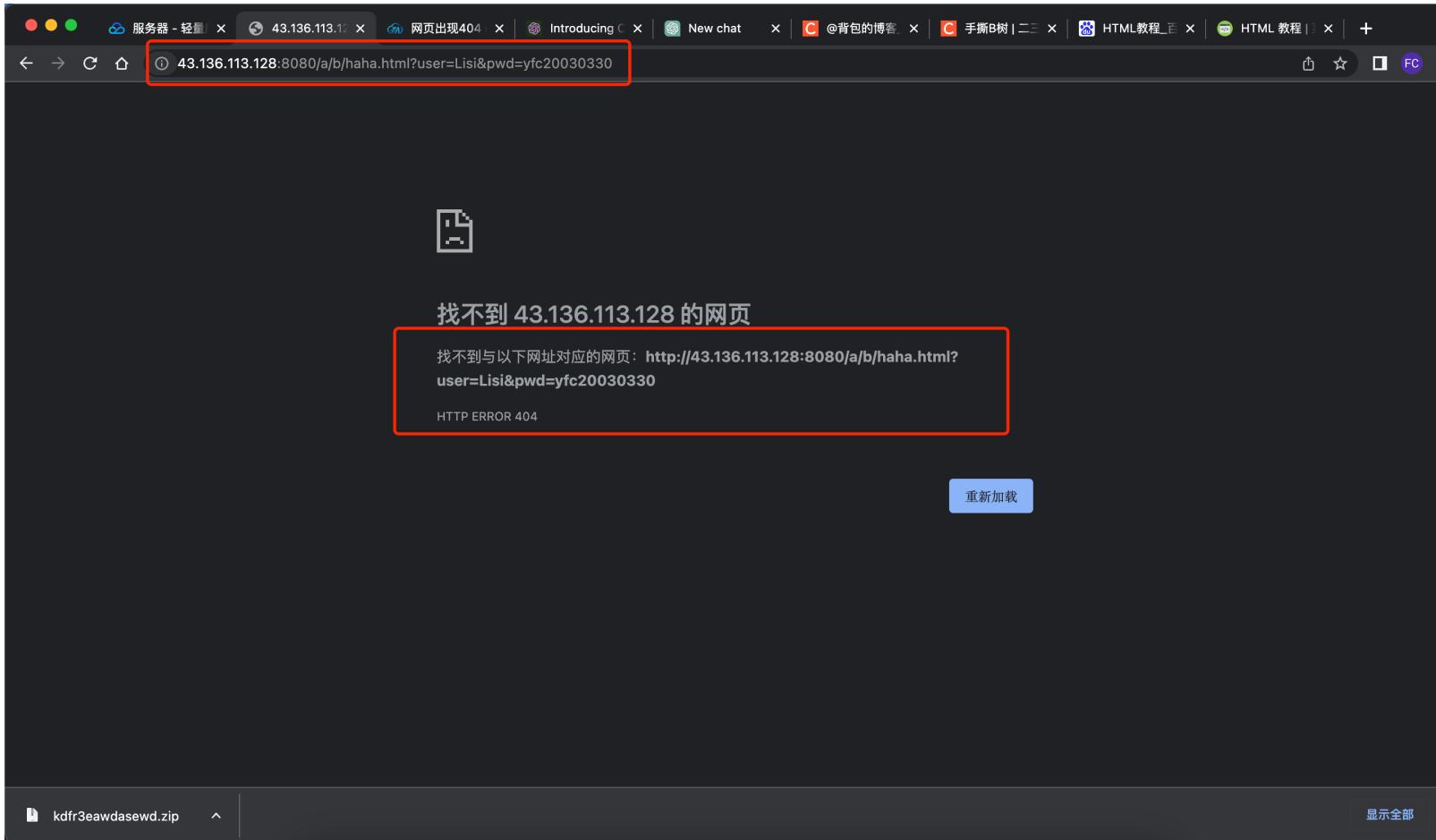
此时就能引出一个概念：表单

它可以收集用户的数据，并把用户数据推送给服务器

表单中的数据，会被转化成http request中的一部分

```
h++ HttpServer.hpp    h++ Sock.hpp    h++ Log.hpp    C++ HttpServer.cc    index.html X    h++ utility.hpp    h++  
Core > BitCodeField > 0226 > http_demo > wwwroot > index.html > html > body > form > br  
1   <!DOCTYPE html>  
2   <html lang="en">  
3   <head>  
4       <meta charset="UTF-8">  
5       <meta http-equiv="X-UA-Compatible" content="IE=edge">  
6       <meta name="viewport" content="width=device-width, initial-scale=1.0">  
7       <title>SYSU ComputerNetWork Course</title>  
8   </head>  
9   <body>  
10     <h3>Testing</h3>  
11     <form name="input" action="/a/b/haha.html" method="get">  
12         UserName: <input type="text" name="user"><br/>  
13         Password: <input type="password" name="pwd"><br/>  
14         <input type="submit" value="Login">  
15     </form>  
16   </body>  
17 </html>
```





我们发现，提交之后，http收到了这么一个请求

<http://43.136.113.128:8080/a/b/haha.html?user=Lisi&pwd=yfc20030330>

所以我们可以发现，get方法可以帮我们把东西的参数
通过url来传参！

Ok！现象我们看到了，get方法通过
url传参

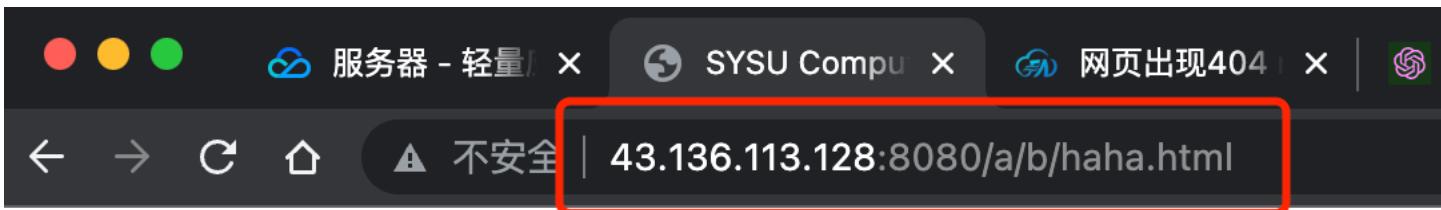
那么post方法呢？

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>SYSU ComputerNetwork Course</title>
8  </head>
9  <body>
10     <h3>Testing</h3>
11     <form name="input" action="/a/b/haha.html" method="post">
12         UserName: <input type="text" name="user"><br/>
13         Password: <input type="password" name="pwd"><br/>
14         <input type="submit" value="Login">
15     </form>
16  </body>
17  </html>
```

我们看到的现象：

用post方法的时候

我们并没有看到参数在url上回显

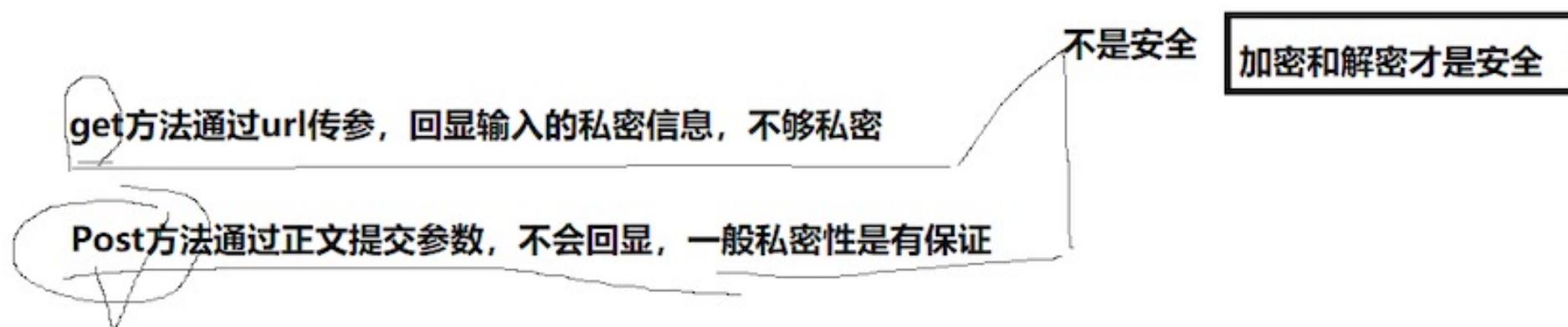


Testing

```
[NORMAL] [1682437642] link success, serviceSock: 4 | 120.236.174.197 : -1658836626
POST /a/b/haha.html HTTP/1.1
Host: 43.136.113.128:8080
Connection: keep-alive
Content-Length: 27
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://43.136.113.128:8080
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://43.136.113.128:8080/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
user=Wangwu&pwd=Wangwu01234-----
```

我们发现，数据放在这里了

所以，post方法是通过在http报文的正文中添加东西来传参的！



注意，上面说的私密不是安全
对数据加密和解密才是安全

状态码

HTTP的状态码

类别	原因短语
1XX Informational (信息性状态码)	接收的请求正在处理
2XX Success (成功状态码)	请求正常处理完毕
3XX Redirection (重定向状态码)	需要进行附加操作以完成请求
4XX Client Error (客户端错误状态码)	服务器无法处理请求
5XX Server Error (服务器错误状态码)	服务器处理请求出错

4开头叫什么呢？这个是客户端错误码
客户去访问一个服务端不存在的资源，是客户端的问题。

5开头的叫服务端错误码

最常见的状态码，比如 200(OK), 404(Not Found), 403(Forbidden), 302(Redirect, 重定向), 504(Bad Gateway)

什么叫重定向状态码？

其实我们平时浏览网页，可能会跳转到另一个地址
这个是通过重定向完成的

状态码

永久重定向 301 vs 临时重定向 302/307

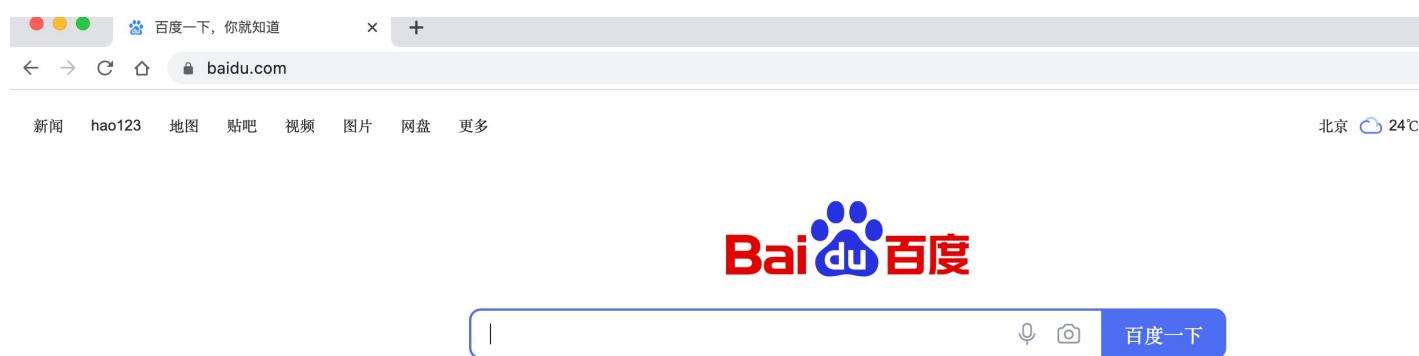
临时重定向，不影响用户后续的请求策略

永久重定向，影响用户后续的请求策略

添加 Location 字段

```
in.close();
}
// 2. 构建一个Http构建一个响应
std::string HttpResponse;
if (content.empty())
{
    HttpResponse = "HTTP/1.1 302 Found\r\n";
    HttpResponse += "Location: https://www.baidu.com/\r\n"; // 如果找不到 直接重定向到百度
}
else
{
    HttpResponse = "HTTP/1.1 200 OK\r\n";
    HttpResponse += "\r\n";
}
```

如果找不到，我们可以让我们的网页直接重定向到百度



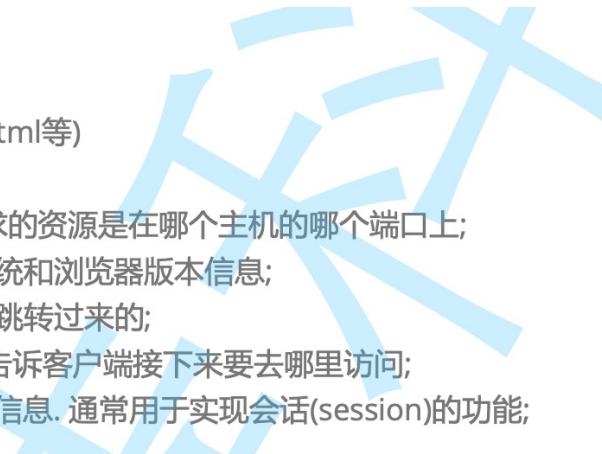
当我们给一个不存在的路径时

浏览器就自动跳转到了百度了！

现在我们可以回过头来再谈一下http的报头

HTTP常见Header

- Content-Type: 数据类型(text/html等)
- Content-Length: Body的长度
- Host: 客户端告知服务器, 所请求的资源是在哪个主机的那个端口上;
- User-Agent: 声明用户的操作系统和浏览器版本信息;
- referer: 当前页面是从哪个页面跳转过来的;
- location: 搭配3xx状态码使用, 告诉客户端接下来要去哪里访问;
- Cookie: 用于在客户端存储少量信息. 通常用于实现会话(session)的功能;



我们刚才写的http服务器版本是
没有加啥报头的

就刚才加了一个Location

我们还可以加上一些其他的

Content-type也是有对照表的一般的服务器，会根据请求的报文里面，获取客户端要抓取资源的类型

然后content-type这里每种资源的类型有不同的写法
网上可以查对照表

```
}  
else  
{  
    HttpResponse = "HTTP/1.1 200 OK\r\n";  
    HttpResponse += "Content-Length: " + std::to_string(content.size()) + "\r\n";  
    HttpResponse += ("Content-Type: text/html\r\n");  
}  
  
HttpResponse += "\r\n";  
HttpResponse += content;  
send(sockfd, HttpResponse.c_str(), HttpResponse.size(), 0);
```

当然，为什么我们之前不加上Content-Type也能浏览呢？

因为现在的浏览器都很聪明，即便没有这个报头他也能自己识别类型，至少Chrome是可以的

当然，如果你把html文件的Content-Type写成纯文本，那么我们的浏览器也不会去解释html文件了，直接当文本输出

会话管理

HTTP常见Header

- Content-Type: 数据类型(text/html等)
- Content-Length: Body的长度
- Host: 客户端告知服务器, 所请求的资源是在哪个主机的那个端口上;
- User-Agent: 声明用户的操作系统和浏览器版本信息;
- referer: 当前页面是从哪个页面跳转过来的;
- location: 搭配3xx状态码使用, 告诉客户端接下来要去哪里访问;
- Cookie: 用于在客户端存储少量信息. 通常用于实现会话(session)的功能;

http的特征:

1. 简单快速
2. 无连接
3. 无状态

无状态是什么意思呢?

即http是不会有客户端以前的动作有所记录的。你客户端之前是否访问过我这个页面, http是不知道的。

那么就会有一种情况, 如果我们平时登陆一个网页, 登录之后是会跳转到另一个页面的, 如果跳转之后这个页面之后, http无状态, 那跳转之后的页面怎么知道我是登陆之前的还是登陆之后的呢?

但是实际上, 我们在使用的时候, 一般网站是会记录下我的状态的



会话管理

http的特征:

1. 简单快速
2. 无连接
3. 无状态

实际上我们在进行使用的时候，一般网站是会记录下我的状态的

网络功能



cookie文件

文件or内存文件

用户名

密码等私密信息



Cookie:

Set-Cookie:

个人信息就被严重泄漏了



注入木马

www.bite.com

用户名:

密码:

认证



Cookie文件存在内存级别的也存在文件级别的
如果删除整个浏览器，发现再登陆还在的话，就是文件级别的

哔哩哔哩 (° - °) つ口 干杯~-bilibili

bilibili.com

在这里可以查看Cookie文件

首页 番剧 直播 游戏中心 会员购 漫画 赛事 放放假 纪录片 下载客户端

搜索框: kcar

动态 大会员 消息 99+ 动态 收藏 历史 创作中心 投稿

动态 热门 频道

番剧 国创 综艺 动画 鬼畜 舞蹈 娱乐 科技 美食 汽车 运动

电影 电视剧 纪录片 游戏 音乐 影视 知识 资讯 生活 时尚 更多

专栏 活动 社区中心

直播 课堂 新歌热榜

换一换

致不火的你 Season 2

完结！我们终将在未来重逢

深圳农村惊人的物价水平

调剂复试被刷，我在回程的绿皮火车上找到了未来的答案

自己开飞机游曼哈顿 自由女神到华盛顿大桥 被纽瓦克塔台教训

如何从零准备一次留学申请-MIT哈佛CMU录取经验分享

每天建模一小时，挑战外包赚钱！

美国最贵超市

哔哩哔哩 (° - °) つ口 干杯~-bilibili

如果我们在**bilibili**的全
都移除掉

我们再打开试试

发现已经退出登陆了

Cookie 和网站数据

来自您正在查看的网站

包括同一个网域中的各个网站，例如 google.com 和 mail.google.com。您正在查看的网站可将数据保存在您的设备上。

来自其他网站

您正在查看的网站可以嵌入其他网站的内容，例如图片、广告和文字。这些嵌入式网站可将数据保存在您的设备上。

完成

如何从零准备一次留学申请-MIT哈佛CMU录取经验分享

每天建模一小时，

首页 番剧 直播 游戏中心 会员购 漫画 赛事 放放假 记录片 下载客户端 新鬼灭之刃锻刀村篇 登录 大会员 消息 动态 收藏 历史 创作中心

动态 热门 频道

番剧 国创 综艺 动画 鬼畜 舞蹈 娱乐 科技 美食 汽车 运动

电影 电视剧 纪录片 游戏 音乐 影视 知识 资讯 生活 时尚 更多

直播 课堂 新歌热榜

是时候展现真正的技术了

社牛选手采访秀英文！

我求婚啦

二喜的奶茶店

超人强自己看了都没绷住

正经配音 谷歌机翻

所有人给我站一边

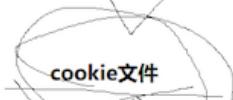
会话管理

http的特征:

1. 简单快速
2. 无连接
3. 无状态

实际上我们在进行使用的时候，一般网站是会记录下我的状态的

网络功能



文件or内存文件

用户名
密码等私密信息



Cookie:

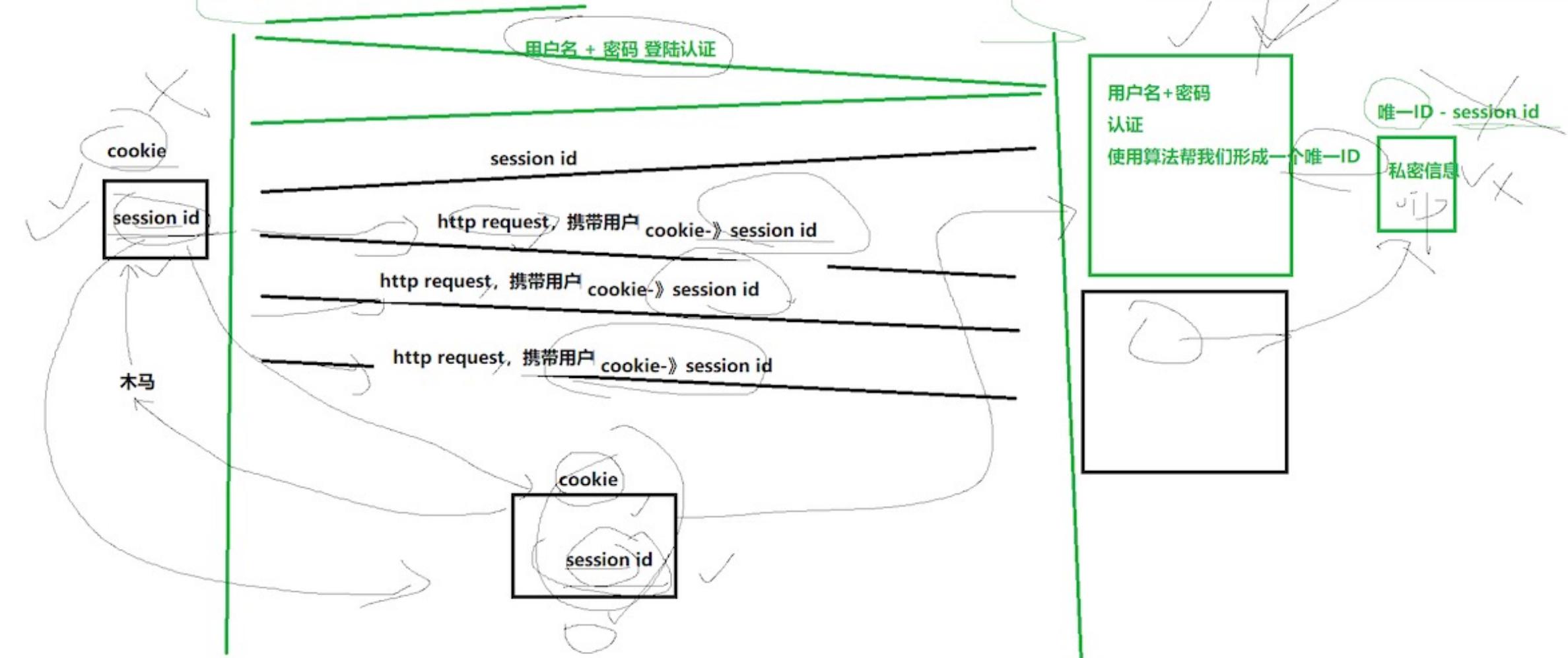
Set-Cookie:

黑客

个人信息就被严重泄漏了

注入木马

但是这种方案，十分的不安全！



```
-----  
[NORMAL] [1683534093] link success, serviceSock: 4 | 120.236.174.205 : -1264659026  
GET /error/404.html HTTP/1.1  
Host: 47.120.41.234:8080  
Connection: keep-alive  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0  
.0 Safari/537.36  
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8  
Referer: http://47.120.41.234:8080/  
Accept-Encoding: gzip, deflate  
Accept-Language: zh-CN,zh;q=0.9  
Cookie: this is a cookie; this is a cookie
```

Connection选项

如果是keep-alive叫做长连接

close表示短连接

一张完整的网页，是由很多的资源构成的