HTTP协议

- 测试高阶课程内部讲义——**讲师：潘sir**

HTTP简介

HTTP

名称：超文本传输协议

超文本：网页源代码(HTML)，我们看到的网站就是浏览器对html渲染后的结果

特点：

1. 基于请求和响应模式：一个请求对应一个响应
2. 无状态：无记忆，每个状态都是独立的，不关联的
3. 应用层协议：基于TCP/IP

URL详解

URL：Universal Resource Lɵcator，统一资源定位符

请求示例：访问百度

- url输入：https://www.badiu.com/s?wd=深圳
- 说明：通过url访问，url指定百度的地址，百度收到请求，给你一个响应

URL拆解：

百度地址：https://www.baidu.com/s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=2&tn=baiduhome_pg&wd=%E6%B7%B1%E5%9C%B3&rsv_spt=1&oq=%25E6%25B7%25B1%25E5%259C%25B3&rsv_pq=fcd4422c0005494d&rsv_t=b202vuEHPjv4OYj1rhfPpq9fMRqta5MYI3Tr9ytRQ0%2FVfHREWPYRd5PJeBGXyAvFU3M6&rqlang=cn&rsv_enter=0&rsv_dl=tb&rsv_sug3=1&rsv_sug1=1&rsv_sug7=100&rsv_sug4=1858

1. 协议类型：http/https，https多了证书，测试环境一般是http，正式环境是https
    - https的安全基础是ssl，所以传输内容是经过ssl加密的，主要作用如下：
        1. 建立一个信息安全通道来保证数据传输安全
        2. 确认网站的真实性，通过点击"锁头"标志来查看证书
2. 主机地址或域名：www.baidu.com
    - ip地址+端口：192.168.xx.xx:8080
        - 说明：默认端☑号是80的时候，可以省略
    - 本机调试：localhost:8080
    - 域名：www.xxx.com，一般域名都会通过nginx映射到ip+端口
3. 服务器路径地址(接口名称)：/s

4. 分隔符号：?
5. 请求参数：wd=%E6%B7%B1%E5%9C%B3 / wd=深圳
    - %E6%B7%B1%E5%9C%B3 是 “深圳” 的urlencode编码，目的是为了将中文转换成该编码格式，让不同的服务器都能够识别
    - 在线转码：http://tool.chinaz.com/tools/urlencode.aspx
6. 多个参数分割：ie=utf-8&f=8&rsv_bp=1&rsv_idx=2&tn=baiduhome_pg&wd=深圳

抓包

抓包工具

- 浏览器F12(network)，或者charles，fiddler，wireshark
- 浏览器经常用于测试：权限控制的问题，比如：你要跳转到一个被控制的页面

浏览器抓包

例如请求：www.baidu.com

network纵览：

1. Name：请求名称
2. Status：响应状态码，200代表响应正常，后端返回
3. Type：请求的文档类型，document代表请求的是html文档
4. Initiator：请求源，标记请求是哪个对象或进程发起的
5. Size：从服务器下载的文件和请求的资源大小
    - from memory cache：从内存取的缓存内容
    - from disk cache：从磁盘取的缓存内存
6. Time：发起请求，到获取响应的耗时
7. Waterfall：可视化瀑布流

选择一个条目查看：

- General：
    1. Request URL：请求URL地址
    2. Request Method：请求方法
    3. Status Code：响应状态码
    4. Remote Address：远程服务器的地址和端口，一般是外网ip和端口
    5. Referrer Policy：Referrer判别策略
- Request Headers
- Response Heraders
- Form Data

抓包重点

- 区分get/post请求：
    - 抓包，找到该请求的请求头，找到Request Method：GET/POST，即可判断

    - 抓包，get没有body部分，post带body，但body可为空
- 分析请求头：重点分析 Content-Type
- 分析请求内容

HTTP请求

请求，由客户端向服务端发出，可以分为4个部分：

1. Request Method 请求方法
2. Request URL 请求地址
3. Request Heders 请求头
4. Request Body 请求体(非必须)

请求方法

http1.0/http1.1区别

- http1.0包含3种最基本的方法：get/post/head
- http1.1新增了5种方法：optinos/put/delete/trace/connect

请求方法说明

主要是用的是get/post/put/delete

请求方法	说明
get	请求指定的页面信息，并返回实体主体
post	多了一个body，提交大量数据
put	从客户端向服务器传输数据，并取代指定文档的内容，修改
delete	请求服务器删除指定的页面
head	类似get，不过返回响应中没有具体内容，用于获取请求头
connect	把服务器当作跳板，让服务器代替客户端访问其他页面，很少用
options	允许客户端查看服务器端性能
trace	回显服务器接收到的请求，主要用于测试或诊断

- 注意：
    1. Get和Post都可以提交数据，从功能上讲，是没有区别的
    2. Get请求的参数包含在URL里，数据可以在URL中看到
    3. Post请求的URL是不包含参数，数据放在请求体里，可以提交大量数据，例如：图片，文件
- 场景说明：
    1. 登录提交用户名/密码，包含了敏感信息，使用Get请求会暴露在URL，所以选择用Post请求
    2. 上传文件/图片时，内容较大，也会选用Post请求

请求地址

- 即URL，通过它**唯一确定**我们想要请求的资源

请求头

4个请求对比分析：

CSDN博客请求(get)：https://blog.csdn.net/qq_32618327/article/details/93973190

CSDN评论请求(post)：https://blog.csdn.net/qq_32618327/phoenix/comment/list/93973190?page=1&size=15&tree_type=1

CSDN关注列表(post-json)：https://msg.csdn.net/v1/web/message/view/unread

本地ip端口请求(登录-post-formdata)：http://0.0.0.0:5001/login

- General部分：

```
Request URL:
https://blog.csdn.net/qq_32618327/phoenix/comment/list/93973190?
page=1&size=15&tree_type=1
Request Method: POST
Status Code: 200 OK
Remote Address: 39.96.126.153:443
Referrer Policy: unsafe-url
```

字段	说明
Request URL	请求地址
Request Method	请求分方法
Status Code	响应状态：200/ok
Remote Address	远程地址，一般不关注
Referrer Policy	一般不关注

- Request Headers部分：

```
Accept: */*
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9
Connection: keep-alive
Content-Length: 0
Cookie: xxx
Host: blog.csdn.net
Origin: https://blog.csdn.net
Referer: https://blog.csdn.net/qq_32618327/article/details/93973190
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_2)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.67 Safari/537.36
X-Requested-With: XMLHttpRequest
```

字段	说明
Accept	指定**客户端**可接收的数据类型
Accept-Encoding	可接收的压缩格式(内容编码)，当服务端返回的数据量比较大(3M图片)，加快速度，节省带宽和流量
Accept-Language	指定**客户端**可接收的语言类型
Connection	keep-alive 长连接(持久连接)；http1.1默认启用Keep-Alive，如果加入"Connection: close "，才关闭
Content-Length	HTTP消息长度，若压缩，则为压缩后的长度
Content-Type	指定body类型，post请求一般都有，否则服务端无法解析
Cookie	缓存信息(登录缓存)，维持当前访问对话
Host	请求资源的域名或ip端口
Referer	标识这个请求从哪个页面发过来的，可用作：来源统计，防盗链处理
User-Agent	标识从哪个客户端(浏览器/操作系统/语言)发出去的，后端分析是否是脚本(拒绝请求)，伪装请求

请求体

请求体一般存放的内容是：Post请求中的表单数据。对于get请求，请求体则为空

解释：

1. get请求没有请求体
    - 例子：csdn关注列表 => https://www.csdn.net/api/articles?type=more&category=watchers&shown_offset=1557641870000000
2. post请求一般都有请求体，也可以没有请求体

- 例子1：https://blog.csdn.net/qq_32618327/article/details/93973190 => https://blog.csdn.net/qq_32618327/phoenix/comment/list/93973190?page=1&size=15&tree_type=1 注意：Content-Length为0
- 例子2(from-data表单)：http://0.0.0.0:5001/index 登录
- 例子3(json数据)：csdn关注列表 => https://msg.csdn.net/v1/web/message/view/unread

请求参数分析

- get参数：一般是qury string，使用 & 符号隔开
- post参数(body)：Content-Type来声明支持哪种格式，json格式最常用，具体对应详情可以参考：http://tool.oschina.net/commons

Content-Type支持格式	参数示例	说明
application/x-www-form-urlencoded	key1=xx&key2=yy&key3=zz	表单提交
application/json	{"key1":"xx","key2":"yy","key3":"zz"}	json数据
multipart/form-data	富文本提交	上传文件/图片
text/xml	xml格式，少见	XML数据
ortests/stream	文件下载，二进制流	下载图片/视频

请求分析

请求分析步骤：

1. 分析url地址：域名地址/接口名称/参数
2. 看请求类型：get/post
3. 分析请求头：重点分析 Content-Type
4. 分析请求参数：param/body

Response响应

Response组成

响应，由服务端返回给客户端，分为三部分：

1. Response Status Code 响应状态码
2. Response Headers 响应头
3. Response Body 响应体

状态码

状态码	说明
1xx	请求已经接受，会继续处理
2xx	成功，请求被服务端理解了，请求是正确的，服务端也是正常的返回，没有报错
200	ok/客户端请求成功
3xx	重定向
301	永久重定向，请求重定向到另外一个接口
302	临时重定向，临时设定呼叫转移，过两天又正常
304	用到缓存，请求服务器资源未改变，用本地未过期的缓存
4xx	客户端错误，请求有语法错误或请求无法实现
400	客户端请求语法有错误，不能被服务端理解
401	请求未经授权
403	服务端收到请求，但拒绝提供服务，一般会带有错误描述。例如：缺少cookie
404	请求地址错误：https://tieba.baidu.com/xxxxx/index.html?traceid=，出现404
5xx	服务端错误，服务器未能实现合法的请求
500	服务器发送不可预期的错误，后端bug没跑了
503	服务器当前不能处理客户端请求，一段时间后可能恢复正常。被攻击，或者大并发导致机器压挂

补充：响应码是一种规范，不是固定模版，后端可以自定义code返回

响应头

响应头包含服务器的应答信息

1. Data：响应时间，注意是中国是东八区
2. Content-Encoding：指定响应内容编码/压缩方式
3. Server：包含服务器信息，如名称/版本号
4. Content-Type：返回文档类型
5. **Set-Cookie**：设置cookies，告诉浏览器下次请求带上什么cookie信息
6. Location：重定向指向的地址
7.

响应体

响应体，即：响应内容，一般为json格式，返回的内容一般由前端进行处理并展示，或者直接返回给用户，让用户处理

- 请求网页，响应体就是网页的html代码
- 请求图片，响应体就是图片的二进制数据
- 请求json接口，响应体就是后端返回的json数据