课程目标

HTTP协议内容和方法

HTTP请求常见请求头

- Method
- Path
- 协议版本
- Content-Length
- User-Agent
- Content-Type
- Accept
- Origin
- Referer
- Accept-Encoding
- Accept-Language
- Connection

HTTP常见返回头

- 协议版本
- 状态码
- X-Powered-By
- 日期
- Connection
- Content-Length

HTTP基本方法

- GET方法的目标(在浏览器上输入一个网址,是从服务器获取资源，一个网址代表的是一个资源，url是资源定位符)
- POST方法的目标是在服务器创建资源(比如在服务器创建一个订单)
- PUT方法的目标是在服务器修改资源(比如在服务器修改订单的状态)注意:幂等性就是调用POST10次会创建10个资源，最终在服务器产生的资源是不一样的，但是调用10次PUT在服务器产生的影响是一样的，只要你这10次请求是一样的
- DELETE(在服务器删除资源)

HTTP其他方法

- OPTION(跨域部分讲解)
- TRACE用于显示调试信息(多数网站不支持)
- CONNECT(代理部分讲解)
- PATCH(对资源进行部分更新，极少用)

常见HTTP状态码

状态码

- 1XX:提供信息(100 continue 早期，现在极少用,主要是客服端向服务端传送很大的数据，这个时候会询问一下服务端，如果服务端返回100的话，客户端就继续向服务端发送数据 101 协议切换 例如：HTTP/1.1 101 Switching Protocols Upgrade:websocket Connection 要求切换成webscoket协议)
- 2XX：成功
- 3XX：重定向(重定向到另外的当都资源)
- 4XX：客户端错误(例如: 404 客户端的访问资源不存在)
- 5XX: 服务端错误(服务端异常)

2XX状态码

- 200--OK(主要是GET 其他的也可以)
- 201-- Created 已创建(主要和POST一起)
- 202--Accept 已接收(请求已经收到，当你创建一个订单，这创建订单在服务单是有消耗的，所以一些异步创建的接口,先返回一个202，表示已经收到你的创建订单的请求，结果需要等待通知，或者轮询去获取)

- 203--非权威内容(原始服务器的内容，在中间或者其他地方被修改过，代理服务器中间修改了某些东西)
- 204-- NO Content 没有内容(成功了 但是无内容返回给你)
- 205--Use Proxy 需要代理
- 206--range header 服务器下发了部分内容

多数服务端开发已经不遵循状态码，我们在restful部分条论

3XX状态码

- 300--用户请求了多个选项的资源(返回选项列表，很少用)
- 301-- 永久转移
- 302--Found 资源被找到(以前是临时的，不推荐使用)
- 303--See Other 可以使用GET方法在另外一个URL找到资源(最终都是使用get方法)，替代原来302的作用
- 304-- Not Modified 没有修改(缓存部分特别说明，资源从缓存里面拿出来，没有再请求一遍,下发资源给浏览器的时候，会把下发给浏览器的内容，计算成一个一个唯一的hash值，hash值作作为Tag传下来，浏览器会对比这个Tag,如果Tag变化了，重新请求，没有变化就不会再请求)
- 305--Use Proxy需要代理
- 307--临时重定向(和303语义差不多，但是最终使用的和原来的一样，是post就是post，get就是get)
- 308--永久重定向(和301语义差不多)

面试疑惑 301VS308

共同点：资源被永久移动到新的地址

差异:客户端收到308请求后，之前是什么method，那么之后也会延用这个method(POST/GET/PUT)到新地址。客户端收到301请求后，通常用户会向新地址发送GET请求。这说一个历史问题，最原始的时候，不是这么多ajax请求，我们发json，收json，然后动态替换内容。现在浏览器用了vue react ng框架之后，就是一个一个html，然后没有内容，内容都是js文件渲染出来的。

面试疑惑 302/303/307

共同点:资源临时放到新地址(请不要缓存)

差异:302是http1.0提出的，最早叫做MovedTemporarily；很多浏览器实现的时候没有遵照标准，把所有请求都重定向为GET 1999年标准委员会增加了302和307，并将302重定义

为Found 303告诉客户端使用GET方法重定向资源 307告诉客户端使用原请求的method重定向资源

4XX状态码

- 400-Bad Request 请求格式错误(入参的数据格式)
- 401-Unauthorized 没有授权
- 402-Payment Required 请先付费
- 403-Forbidden 禁止访问(401去授权登录，可以解决 403是解决不了，禁止访问，找相关管理员)
- 404-Not Found 没有找到资源，这个都定义到是客户端的错误
- 405-Method Not Allowed 方法不被允许(这个资源只接收post，却给了一个put，就可能出现405，也有可能出现404)
- 406-Not Acceptable 服务端可以提供的内容和客户端期待的不一样(例如：服务端期待的是utf-8的数据,却给了国标码的数据，这个时候可能会返回406，如果服务端做的好，还可以，有的做的乱的好，连404都木有，直接就抱500了)

5XX状态码

- 500-Internal Server Error(内部服务器错误)
- 501-Not Implemented(没有实现，这个接口已经存在，但是我还木有写)
- 502-Bad Gateway(网关错误) 现在互联网体系，你请求百度，不是直接就到百度的服务器上的，是通过防火墙，负载均衡才能请求到服务器上，所以负载均衡其实就是一个网关，请求到达网关，再通过网关请求服务器内容的时候，出现错误的话，网关直接就抛出502 网关错误
- 503-Server Unavailable 服务不可用(有很多原因比如线程池溢出 服务正在启动 服务的内存用光了.....)
- 504-Gateway Timeout(网关超时），通过网关去请求服务的时候,服务一直没响应。挂了，服务器不会通知的，但是网关知道自己超时了，所以504的话，直接看网关，服务挂了，没有响应
- 505-HTTP Version Not Supported(版本不支持)例如:浏览器http2.0的请求http1.0或者1.1的服务的时候会出现

常见HTTP请求头

content-length

- 发送给接收者的Body内容长度(字节,一个byte是8bit utf-8编码的字符是1-4个字节,在request的请求头和response的header里面都可能看到content-length)

user-Agent

- 帮助我们区分客户端特性的字符串(操作系统 浏览器 制造商 内核类型 版本号...)

content-Type

- 帮助区分资源的媒体类型(text/html text/css application/json image/jpeg ....)

Origin

- 描述请求来源地址(scheme://host:port 不含路径 可以是null)

Accept

- Accept:建议服务端返回何种媒类型(只是建议，服务端会不会接纳，客户端无法控制 */*代表返回所有类型(默认) 多个类型用逗号隔开例如:text/html applicetion/json)
- Accept-Encoding：建议服务端发送那种编码(压缩算法 defate,gzip,*;q=0.5)
- Accept-language:建议服务端传递哪种语言 fr-Ch,fr；q=0.9.....

Referer

- 告诉服务端打开当前页面的上一张页面的URL 如果ajax请求那么就告诉服务端请求的URL是什么(非浏览器环境有时候不发送Referer(或者虚拟的Referer通常是爬虫)，常常用户行为分析)

connection

- 决定连接是否在当前事务完成后关闭(Http1.0 默认就是close(最早的http请求是无状态的，request过去，response回来，请求就关了) Http1.1后默认是keep-live(考虑到性能的优化，TCP连接还是开的，因为后续还有很多请求，TCP连接开着，就少了三次握手的过程))

课程小结

思考平时工作中前端和服务端有没有遵循HTTP协议规范？(OSI TCP/IP 协议群)

HTTP协议相关代理/安全/缓存/跨域等重点内容见后续内容

http长连接什么时候断开？http1.1后出现的keep-alive 多个http复用一个tcp连接，但是这有时间限制的，服务端必须接收到多个连续的请求连接才不会断开，否则会断开。