

Http面试总结

2017年8月16日
21:55

1. GET方法和POST方法的区别？
 - ①. GET方法
 - i. 一般用于获取服务器的数据
 - ii. 发送数据放置在URL后面，因此发送数据更小
 - iii. 不是安全的，是幂等的。
 - ②. POST方法
 - i. 一般用于上传数据到服务器，改变服务器的状态
 - ii. 数据字段放在报文体中，因此发送数据长度比get长
 - iii. 安全的。
2. HTTP的相应状态码？

200 : 请求被正常处理。

206 : 请求部分资源，Content-Range指定范围的资源。

301 : 永久性重定向

302 : 临时重定向

304 : HTTP缓存，当前的缓存可用，直接从缓存中读取数据。

400 : 客户端语法错误

403 : 服务器拒绝执行此请求

404 : 请求资源不存在

500 : 服务器错误

503 : 服务器忙
3. forward 和 redirect有什么区别？
 - ①. forward是服务器内部重定向，服务器内部直接访问url，把数据读取过来发送给客户端，因此在客户端浏览器地址栏中不会显示转后的地址
 - ②. redirect 是客户端的重定向，是完全跳转。客户端获得重定向的url后，重新发一次http请求。
4. 断点续传？文件更新后如何续传？

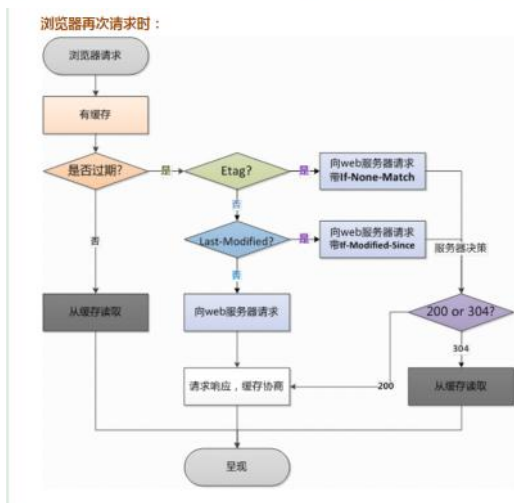
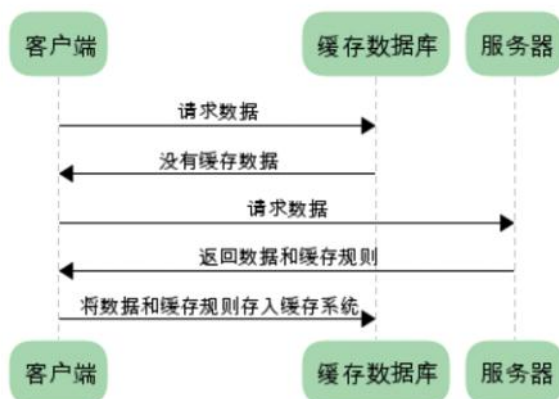
range 字段和 content-range字段。

if-range头: ETag和last-modified字段，服务器通过校验，不一致则返回200和发送新的数据，一致就返回206状态码继续续传。
5. cookie与session的区别？
 - ①. cookie数据放在客户的浏览器上，session数据放在服务器上
 - ②. cookie的安全性不够。
 - ③. session会在一定时间内保存在服务器上，当访问量增加，服务器的压力增大。
 - ④. 单个cookie保存的数据不能超过4kb，浏览器对cookie个数有限制。
 - ⑤. 将登录信息等重要信息放在session，其他信息放在cookie中
6. cookie格式和组成？

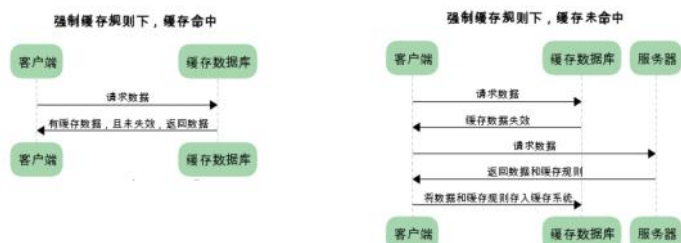
set-cookie :

 - ①. NAME = VALUE : cookie必须有的部分。name为cookie的名称，value是该cookie的值
 - ②. Expires = DATE : expires 是cookie的有效终止日期
 - ③. Path = PATH : path属性定义了web服务器上哪些路径下的页面可获取服务器设置的cookie。path= '/' 则web服务器上所有的www资源均可读取该cookie。
 - ④. Domain = DOMAIN_NAME : 确定了哪些internet域中的web服务器可读取浏览器所存储的cookie。可选的。
SECURE : 标记该变量表明只有当浏览器和web server之间的通信协议为加密认证协议时，浏览器向服务器提交相应的cookie，当前这种协议为HTTPS
7. 禁止浏览器缓存的头字段？
 - ①. expires = 0/-1 生命周期过期
 - ②. Cache-control : no-cache
8. Http缓存机制

第一次请求数据



已存在缓存数据时，仅基于强制缓存，请求数据的流程如下



已存在缓存数据时，仅基于对比缓存，请求数据的流程如下



服务器在响应请求时，告诉浏览器资源的最后修改时间。

▼ Response Headers [view source](#)

```

Cache-Control: max-age=31536000
Connection: keep-alive
Content-Encoding: gzip
Content-Type: application/javascript
Date: Tue, 24 Jan 2017 07:26:54 GMT
Etag: W/"5886c231-8d9"
Last-Modified: Tue, 24 Jan 2017 02:55:45 GMT
Server: TGM/EB
Transfer-Encoding: chunked
Vary: Accept-Encoding
    
```

第一次请求时，服务器返回的资源最后修改时间

```

Request Headers
view source
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8
Cache-Control: max-age=0
Connection: keep-alive
Host: m.sitiangou.com
If-Modified-Since: Tue, 24 Jan 2017 02:55:45 GMT
If-None-Match: W/"5886c231-8d9"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0

```

再次请求时，浏览器通知服务器，上次请求时返回的资源最后修改时间

9. 当输入www.google.com时，页面发生了哪些事情？

①. DNS域名解析

- i. 浏览器搜索自身的DNS缓存
- ii. 搜索操作系统的DNS缓存
- iii. 读取本地host文件
- iv. 发起一个DNS的系统调用
 - 1) 带宽运营服务器查看本身缓存
 - 2) 运营商服务器发起一个迭代DNS解析请求

②. 浏览器获得域名对应的IP地址后，发起TCP/IP三次握手

③. TCP/IP连接建立起来后，浏览器可以向服务器发送HTTP请求了

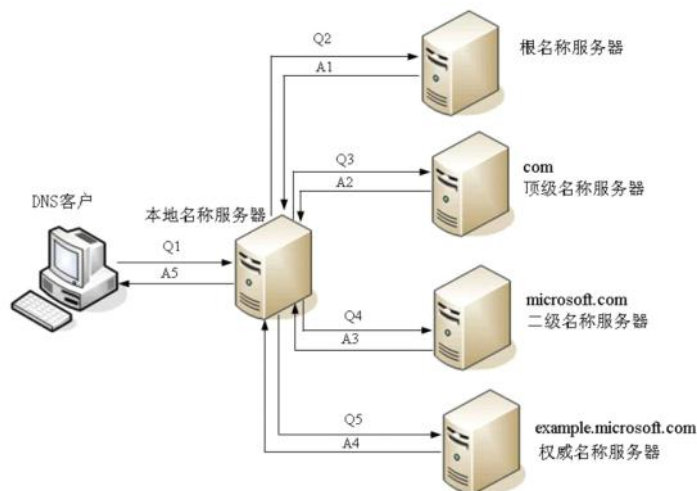
④. 服务器接受这个请求，根据路径参数，经过后端的一些处理生成HTML页面代码返回给浏览器

⑤. 浏览器拿到完整的HTML页面代码开始解析和渲染，如果遇到引用的外部JS,CCS等静态资源，他们同样也发送一个个HTTP请求，都需要经过上面的步骤

10. DNS域名解析，递归解析和迭代解析的区别？

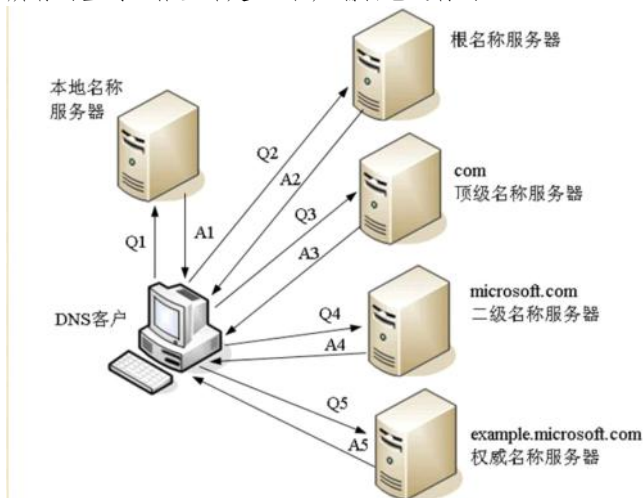
(一). 递归解析

如果客户端配置的本地名称服务器不能解析的话，则后面的查询全由本地域名服务器代替DNS客户端进行查询，直到本地名称服务器从权威名称服务器得到了正确的解析过程，然后由本地域名服务器告诉DNS客户端查询结果。



(二). 迭代解析

所有的查询工作全部是DNS客户端自己进行的。



11. 长连接和短连接

(一). 短连接

- i. 浏览器和服务器每进行一次HTTP操作，新创建一个TCP/UIP连接，任务结束后中断连接
- ii. web网站都用短链接，多用户访问，长连接会占用很多资源。

(二). 长连接

- i. 连接建立后，数据发送完毕后不断开连接，通信一直存在。
- ii. 适用于操作频繁，点对点通信，而且连接数不能太多。

(三). HTTP参数

Connection : keep-alive HTTP1.1默认为持久联结

12. HTTP头部信息

(一). 请求报文

- i. 请求行
 - 1) method 标识 + URI + HTTP 的version号
- ii. 请求报头
 - 1) User-Agent : 产生请求的浏览器类型
 - 2) Accept : 客户端可识别的内容类型列表
 - 3) Host : 请求的主机名
 - 4) Connect:连接相关的属性
 - 5) Accept-Charset : 编码格式
- iii. 空白行
- iv. 请求报文
 - 1) GET方法中没有请求数据，GET的请求数据时加在URL后边的。
 - 2) POST适合用于需要客户端填写表单的场合。

(二). 响应报文

- i. 状态行
 - 1) 版本号 + 状态码 + 状态码描述文本
- ii. 响应报头
 - 1) Content-Type : 响应正文的类型
 - 2) Content-Length : 请求数据的长度
 - 3) Content-Language, Content-Encoding, Content-Charset
- iii. 空白行
- iv. 响应报文