

浏览器缓存

- 把已请求并返回的WEB资源（HTML页面、图片、Js文件、CSS文件、数据等）复制成一个副本存储在浏览器的缓存中。
- 缓存的好处：
 - 1、减少网络带宽的消耗
 - 2、降低服务器压力
 - 3、减少网络延迟

- Pragma: no-cache(响应头) – jsplusplus.com
- 指示浏览器忽略资源缓存副本，每次访问需要到服务器获取
- http1.0中字段，http1.1用Cache-Control替代（http协议向下兼容，所以很多网站依旧有这个字段）

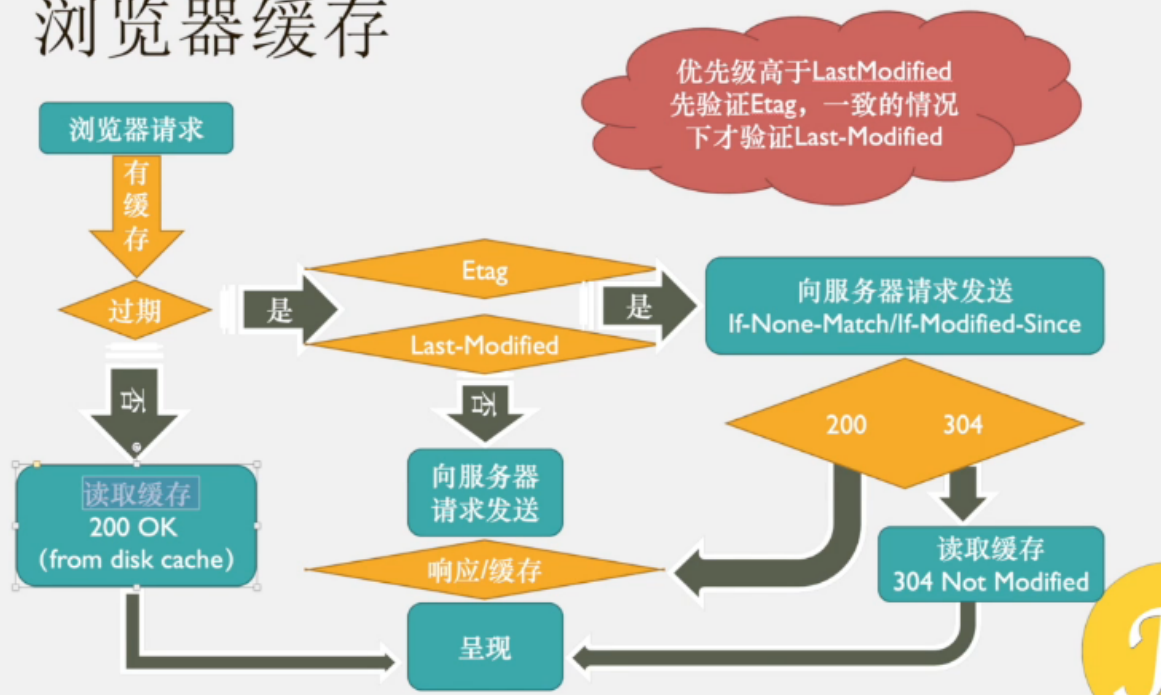
- 如果 Cache-Control 和 Pragma 同时存在，Cache-Control 优先级高于Pragma

- Cache-Control 缓存控制（响应头） jsplusplus.com
- no-cache: 指示浏览器忽略资源缓存副本，强制到服务器获取资源（浏览器依然缓存）
- no-store: 强制缓存在任何情况下都不要保留任何副本
- max-age=31536000: 指示缓存副本的有效时长，从请求时间开始到过期时间之间的秒数
- public: 表明响应可以被任何对象（包括：发送请求的客户端，代理服务器，等等）缓存。
- private: 表明响应只能被单个用户缓存，不能作为共享缓存（即代理服务器不能缓存它）。

- no-cache: 不是说不缓存，是指忽略缓存副本，强制到服务器获取资源
- no-store: 才是不保留缓存

- Expires: Mon, 15 Aug 2016 03:56:47 GMT(Cache-Control更高)
- 启用缓存和定义缓存时间。告诉浏览器资源缓存过期时间，如果还没过该时间点则不发请求
- 在HTTP 1.1版开始，使用Cache-Control: max-age=秒替代

浏览器缓存

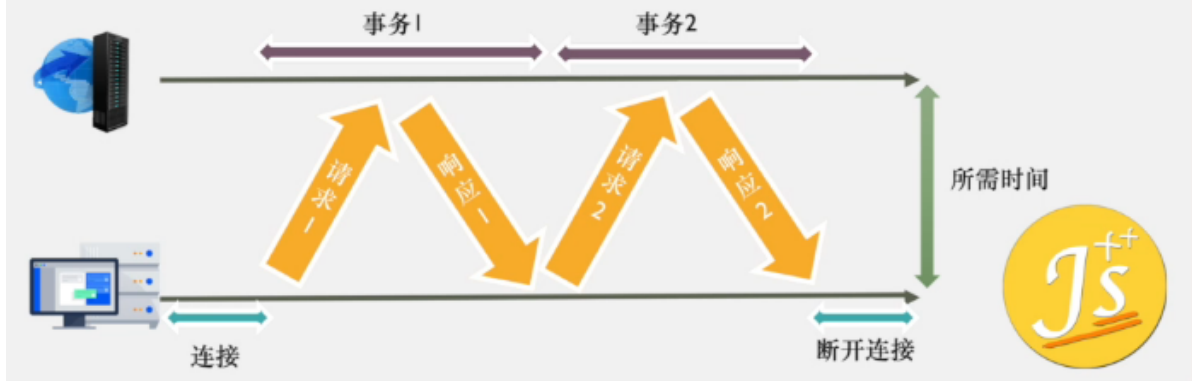


长短连接

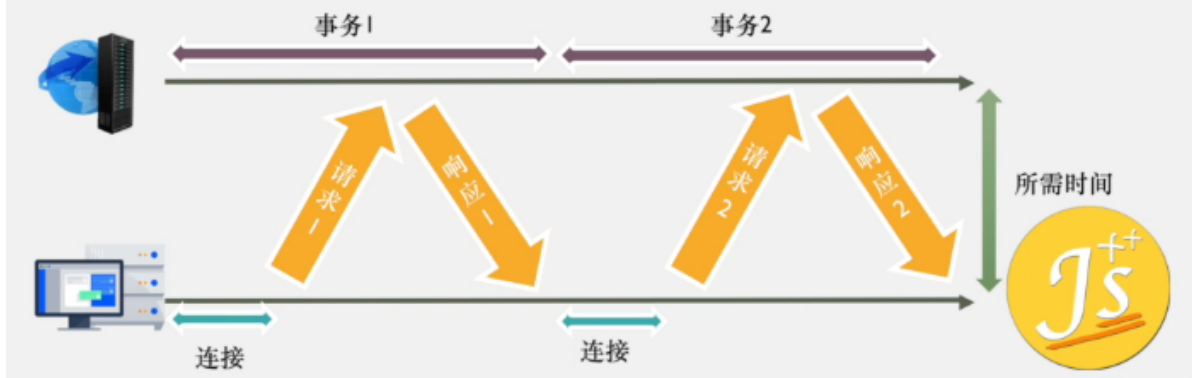
- HTTP短连接与长连接
- 短连接
- 所谓短连接，就是每次请求一个资源就建立连接，请求完成后连接立马关闭。



- HTTP短链接与长连接
- 长连接
- 只建立一次连接，多次资源请求都复用该连接，完成后关闭。



- 早期HTTP1.0: 每个HTTP请求都要创建一个TCP/IP连接
- 串行连接（短连接）



- 后期的HTTP/1.0
- 在请求头增加: Connection: keep-alive
- HTTP/1.1
- 默认开启: Connection: keep-alive
- 如需关闭: Connection: close
- 注意: Connection: keep-alive在大多数浏览器中都默认使用HTTP/1.1
- 服务器也须由相应的支持才能达成C/S协议进行长连接

144115231488510110正在观看视频

CONTENT -LENGTH

- Content-Length用于描述HTTP消息实体的传输长度
- 案例：表单提交 – view source -> request headers -> content-length
- 服务器返回123456 -> response headers -> content-length
- GET请求：请求头没有content-length，响应头带content-length
- POST请求：请求头与响应头都带content-length

- length 单位是字符，length长度是按底层传递的格式来计算的也就是按 a=1&b=2 这种格式来计算的

REFERER

- 案例：google搜索腾讯课堂 -> 打开查看请求头中的Referer -> <https://www.google.com>
- Referer (referrer:来源域名)：Referer是request header的一部分，当浏览器向web服务器发送请求的时候，一般会带上Referer，告诉服务器我是从哪个页面链接过来的
- 好处：可通过Referer分析不同渠道的流量分布、用户搜索的关键词

- 案例：
- 不发送Referer信息：<meta name="referrer" content="no-referrer">
- Referrer Policy: no-referrer
- 无referer字段
- 案例：
- 只发送Referer: origin信息：<meta name="referrer" content="origin">
- Referrer Policy: origin
- Referer: <http://localhost/>

- 案例：
- 发送Referer信息：
- Referrer Policy: no-referrer-when-downgrade
- 仅当协议降级（如HTTPS页面引入HTTP资源）时不发送Referrer信息。是大部分浏览器的默认策略。
- Referer: <http://localhost/network/class5/>

- 第三张图是在meta 中不设定referrer

- no-referrer-when-downgrade: 仅当协议降级 -> HTTPS 跳转 HTTP 时不发送Referrer信息



- 简述 referrer 的防盗链机制
- 简述 七牛的时间戳防盗链机制：通过时间和签名生成一个密钥，这个密钥写在链接后面。当发起请求时，服务器识别密钥，获得时间和签名，比对时间和签名，判断是否正确响应请求