1. http缓存优点:

1 减缓服务器压力

2 让客户端浏览器进行缓存

1. http缓存实现方法:

1 让浏览器缓存

2 让代理服务器缓存

1. http的cache-control:

HTTP/1.1可以通过Cache-Control首部字段来控制缓存

Cache-Control: public

no-store (禁止进行缓存): 指定不能对请求或响应任意一部分缓存

no-cache (强制确认缓存): 缓存服务器要先向服务端验证资源的有效性, 只有当缓存资源有效才能使用缓存对请求进行响应

private (私有缓存): 指定资源作为私有缓存, 只能单用户使用, 一般是存储在用户浏览器中

public (公共缓存): 指定资源作为公共缓存, 可以被多个用户使用, 一般是存储在代理服务器中

1. http缓存过期机制:

max-age 在请求头中表示: 如果缓存时间小于max-age时间, 那么接受该缓存

max-age 在响应头中表示: 表示缓存在缓存服务器存储的时间

Cache-Control: max-age=31536000

Expires用于告知缓存服务器缓存什么时候过期

Expires: Wed, 04 Jul 2012 08:26:05 GMT

在 HTTP/1.1 中, 会优先处理 max-age

在 HTTP/1.0 中, max-age 指令会被忽略

1. http缓存验证:

ETag 字段: 它是资源的唯一标识, 如: [www.github.com有两个资源(中文和英文),](http://www.github.com有两个资源(中文和英文),)

只有 ETag 才能对这两个资源进行唯一标识

ETag: "3a0772443a0739141292a5429b952fe6"

将缓存资源的 ETag 值放入 If-None-Match 首部, 服务器收到该请求后, 判断缓存资源的 ETag 值和资源的最新 ETag 值是否一致, 如果一致则表示缓存资源有效, 返回 304 Not Modified

If-None-Match: "3a0772443a0739141292a5429b952fe6"

Last-Modified 字段也可以用于缓存验证, 它包含在代理服务器发送的响应报文中, 指示源服务 器对资源的最后修改时间, 但是它是一种弱校验器, 因为只能精确到一秒, 所以它通常作为 ETag的备用方案, 如果响应首部字段里含有这个信息, 客户端可以在后续的请求中带上 If-Modified-Since 来验证缓存, 服务器只在所请求的资源在给定的日期时间之后对内容进行过 修改的情况下才会将资源返回, 状态码为 200 OK, 如果请求的资源从那时起未经修改, 那么返

回 304 Not Modified 响应

Last-Modified: Wed, 12 Oct 2019 22:16:25 GMT

If-Modified-Since: Wed, 12 Oct 2019 22:16:25 GMT

1. http内容协商类型:

1 服务端驱动:

客户端设置头信息(Accept, Accept-Language等), 服务端根据头信息返回内容

2 代理驱动:

服务器返回 300 Multiple Choices 或者 406 Not Acceptable,

客户端从中选出最合适的那个资源

1. http内容协商的vary

Vary: Accept-Language

在使用内容协商的情况下, 只有当缓存服务器中的缓存满足内容协商条件时, 才能使用该缓存,

否则应该向源服务器请求该资源。

例如: 客户端发送了包含 Accept-Language 首部字段的请求之后, 服务器返回的响应包含 Vary: Accept-Language 内容, 缓存服务器对这个响应进行缓存之后, 在客户端下一次访问同一个 URL 资源, 并且 Accept-Language 与缓存中的对应的值相同时才会返回该缓存

1. http响应缓存的条件:

1 请求方法必须是GET,HEAD

POST 大多情况下不可缓存

PUT, DELETE不可缓存

2 响应码必须是 200, 203, 204, 206, 300, 301, 404, 405, 410, 414, 501

3 响应报文一定包括Cache-Control, 如果不指定则不缓存

1. 1
2. 1