

缓存 🔄

浏览器缓存

协商缓存(304)

Etag && If-None-Match(http1.1)

Last-Modified && if-modified-since (http1.0, 检查力度是秒级)

Expires(http1.0) (GMT的绝对时间)

⚠️ max-age(秒, 相对单位), 优先级高于expires

no-cache (为它在本地是有拷贝的, 但是在用之前都进行了确认, 这样http请求并未减少, 但可能会减少一个响应体)

no-store (诉浏览器在任何情况下都不要进行cache, 不在本地保留拷贝)

public (表示缓存的版本可以被代理服务器或者其他中间服务器识别)

private (意味着这个文件对不同的用户是不同的。只有用户自己的浏览器能够进行缓存, 公共的代理服务器不允许缓存)

Cache-Control (http1.1)

cache(使用本地缓存)

must-revalidate (强制浏览器严格遵守设置的缓存策略)

proxy-revalidate (强制proxy严格遵守缓存策略)

s-maxage (仅供共享缓存使用, 即proxy)

强制缓存【200 from memory/cache】

服务器缓存

CDN缓存

遵循http缓存策略

cache-control:max-age (关于与回源率关系, 太短回源率高, 负载压力大, 太长, 更新慢)

关注 “回源率” (max-age)

相比客户端ctrl+F5, CDN服务商提供了 “缓存刷新”, 清理CDN边缘缓存

combo服务

url由多个动态部分组成

线上资源合并服务

如果combo的url中任何一个文件发生改变, 都会导致整个url缓存失效, 从而导致浏览器缓存利用率降低

减少http请求

/??fle1,file2,file3 (多个资源合并成一个文件下载)

⚠️ 浏览器url有长度限制2048

HTML5 缓存

manifest

离线缓存资源, 离线应用

仅加载有更改的文件

提升了加载速度和访问体验

如果离线资源更新了, 必须更新manifest

默认更新的资源, 必须等待下次打开应用才能生效

window.applicationCache.swapCache()立即使更新生效

⚠️ 注意移动端的无痕模式, storage不起作用

如果需要兼容低版本IE, 注意使用cookie做兼容

size限制

pc端 (local,session) 各大浏览器一致: 5M

移动端localStorage大小2.5M,安卓(2-5M), ios可能有所不同 🔄

sessionStorage 高版本暂无限制

storage 🔄

超过存储大小, 都会一致抛出错误: QUOTA_EXCEEDED_ERR

firefox用户可以自主设置存储大小(about:config--->dom.storage.default_quota)

set注意事项

key不应过多, 且整站统一约束

页面path作为key, 该页面数据存储在value中

一条 cgi 一个 key 值, 不同参数在 value 中处理 (性能问题) 【动态请求和参数】

只存储提升体验和重要数据

value尽量使用string, 减少JSON.stringify()

不要频繁set/get,尽量在页面加载时, 一次写入或者读取

阻塞UI (事件处理中, 如果同步操作storage会阻塞接下来的UI渲染, 可以使用定时器解决)

indexedDB

web持久化存储

微信小程序缓存

storage

size限制

10M

文件缓存

size限制

小程序10M

小游戏50M