



从前端小工到中级工程师的必备技能

基于缓存的前端优化

正课之前我们先说说：德雷福斯模型

新手(Novice)

新手缺乏经验，完成任务时需要操作指引；

高级新手(Advanced beginner)

能执行要做的任务并根据需求学习新任务，但没对任务环境获得更广泛的概念上的理解，缺乏全局思维；

胜任者(Competent)

胜任者能够独立解决问题。

精通者(Proficient practitioner)

精通者懂得反思，自我纠正，从自己和他人的经验中学习

专家(Expert)

专家凭直觉工作，基于他们在领域中的丰富经验，在恰当的情景中选取和应用这些经验。

在多数时候我们都在培养胜任者

完成指令

自我学习

尽在掌握

自我纠正

多问问自己

- 1 我是不是仅仅完成了任务？
- 2 我是不是觉得已经没什么好学的了？我知道下一阶段要预先准备好什么吗？
- 3 我是不是已经能够很好的把控工作中要做的事情？我能不能跳出舒适区？
- 4 有没有什么问题我忽略了或者遗漏了，未来可能会产生巨大的影响？

OK，回到正题

大部分性能优化都是基于缓存用空间换时间

缓存的作用

1 减少延迟

2 降低网络负载

3 保证稳定性

一、HTTP 缓存

Expires

设置对象的有效期

▼ Response Headers

access-control-allow-origin: *

cache-control: max-age=3600

content-encoding: gzip

content-length: 20924

content-type: text/css

date: Wed, 11 Oct 2017 11:48:58 GMT

expires: Wed, 11 Oct 2017 12:48:58 GMT

last-modified: Wed, 20 Sep 2017 17:17:22 GMT

server: NWSs

status: 200

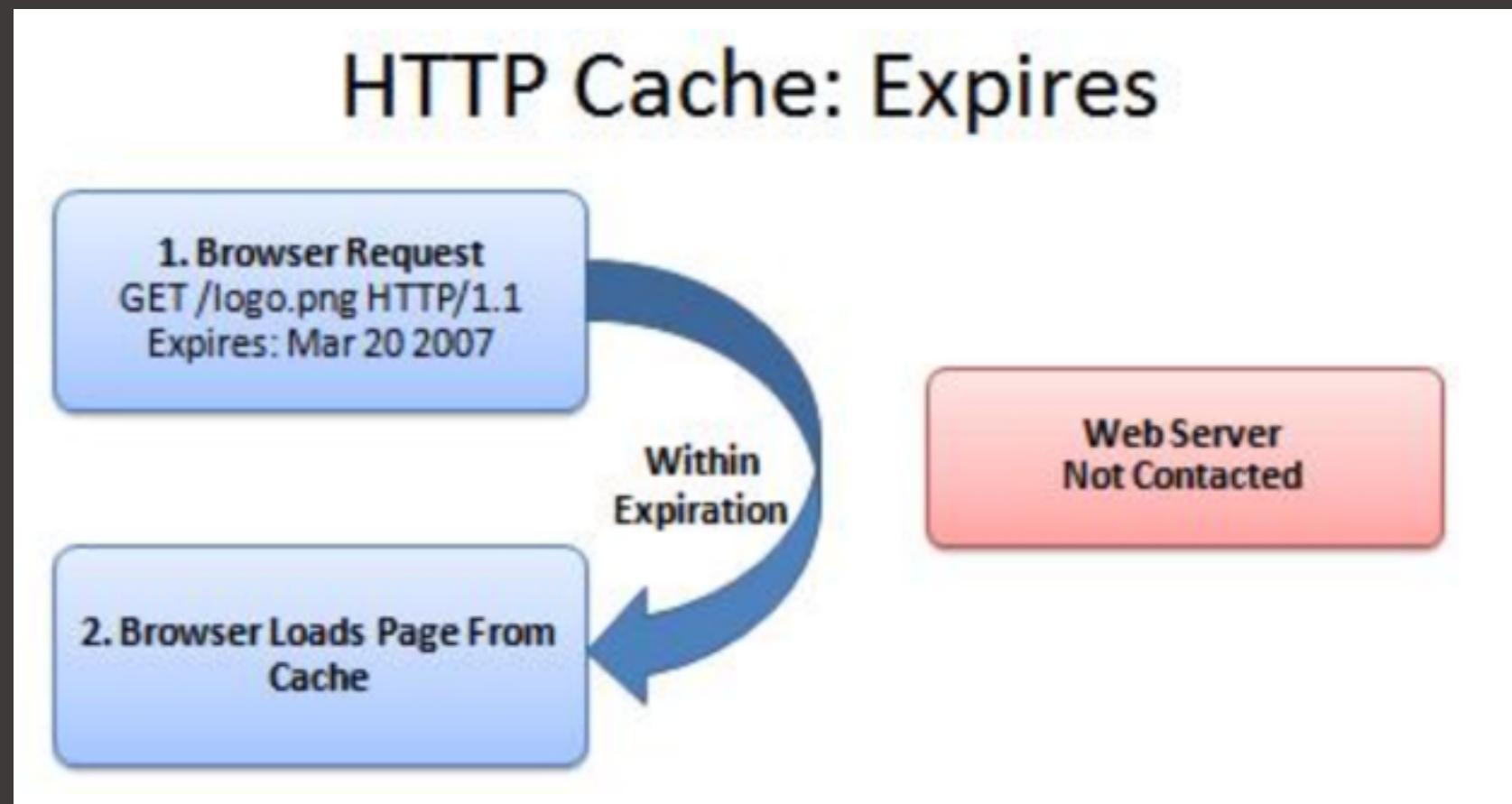
timing-allow-origin: *

x-cache-lookup: Hit From Disktank Gz

x-nws-log-uuid: 854493f1-7b6d-4009-8d1c-cd328db2fbfb

Expires

设置对象的有效期



Cache-Control

用户本地时间可能跟服务器时间并不一致

1 max-age=num(s), 设置最大缓存时间

2 public, 缓存能被多用户共享

3 private, 缓存不能在用户间共享

4 no-cache, 不会被缓存

5 no-store, 不能被存储

last-modified

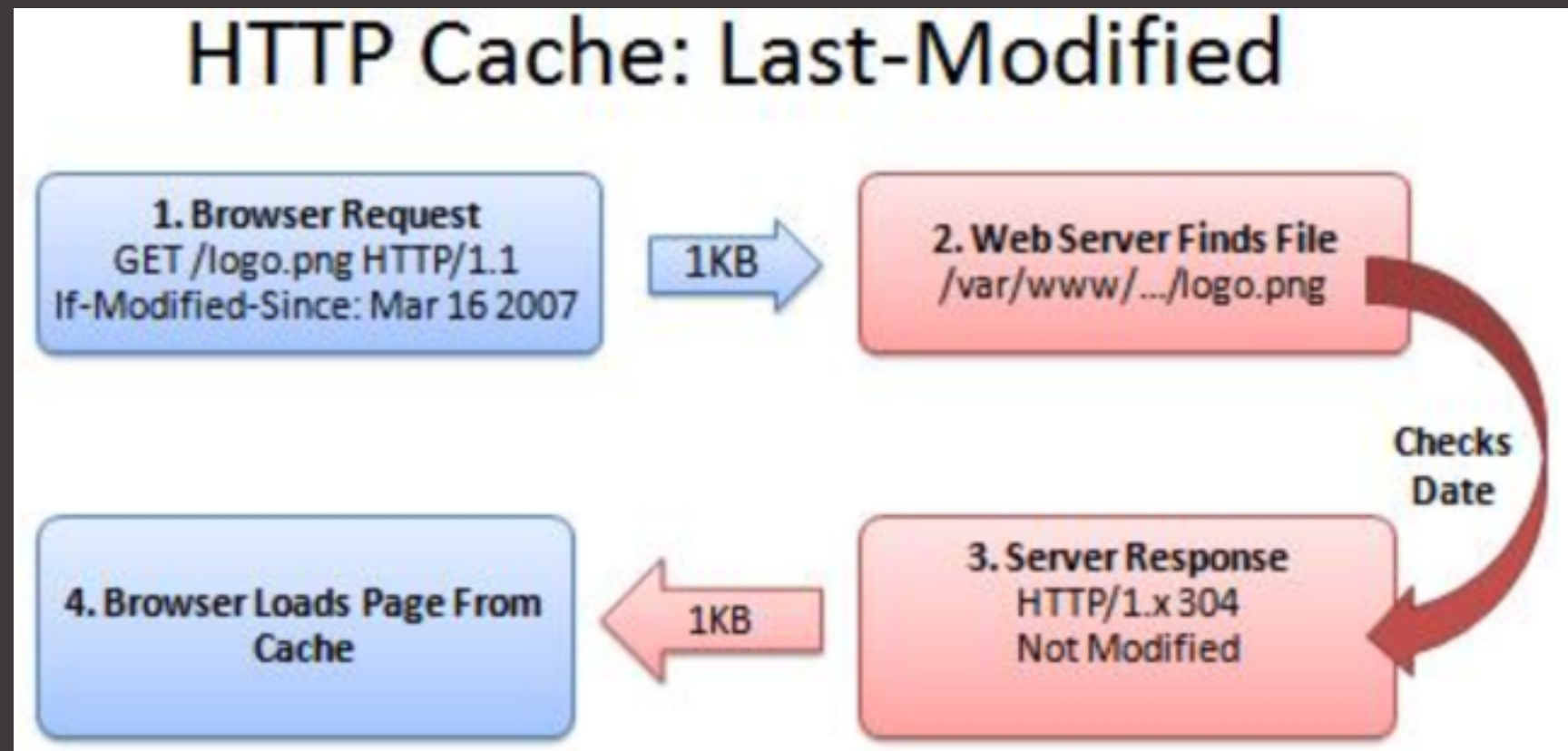
设置对象的最后修改时间

▼ Response Headers

access-control-allow-origin: *
cache-control: max-age=3600
content-encoding: gzip
content-length: 20924
content-type: text/css
date: Wed, 11 Oct 2017 11:48:58 GMT
expires: Wed, 11 Oct 2017 12:48:58 GMT
last-modified: Wed, 20 Sep 2017 17:17:22 GMT
server: NWSs
status: 200
timing-allow-origin: *
x-cache-lookup: Hit From Disktank Gz
x-nws-log-uuid: 854493f1-7b6d-4009-8d1c-cd328db2fbfb

last-modified

设置对象的最后修改时间



E-tag

可能在同一秒保存两次，也可能服务器上文件修改时间并不一致

▼ Response Headers

```
age: 39
cache-control: max-age=0, s-maxage=60
content-encoding: gzip
content-type: text/html; charset=utf-8
date: Wed, 11 Oct 2017 12:06:19 GMT
eagleeye-traceid: 0a67920815077235792645080e758e
etag: W/"380f8-CYSgfb0i7zAQT6V8SYCwr/r6GRs"
server: Tengine/Aserver
status: 200
strict-transport-security: max-age=0
timing-Allow-Origin: *
vary: Origin, Ali-Detector-Type
vary: Accept-Encoding
via: db568081[0,200-0,H]
x-cache: HIT TCP_MEM_HIT dirn:-2:-2
x-readtime: 118
x-response-time: 116
x-server-id: wormholesource010178132229.test.et2
```

E-tag

可能在同一秒保存两次，也可能服务器上文件修改时间并不一致

▼ Request Headers

:authority: www.tmall.com

:method: GET

:path: /

:scheme: https

accept: text/html,application/xhtml+xml,application/xml;q=0.9,
apng,*/*;q=0.8

accept-encoding: gzip, deflate, br

accept-language: zh-CN,zh;q=0.8,en;q=0.6

cookie: _med=dw:1440&dh:900&pw:2880&ph:1800&ist:0; l=AoCAdpcFN
VSemTq; cq=ccp%3D0; hng=CN%7Czh-CN%7CCNY; cookie2=3616408b80c
8b2c; t=b1dfc9e60cc2a84f5f1289b53b6d20f2; skt=9de6b53131b52e3
533863a357; sm4=440300; cna=XJb3EXHVKzECAXkAHcg4jG7S; isg=Avn
W0mCCGPNAfHg8fdMDbsuxiDqohg0Y17UiaiyUlpCy

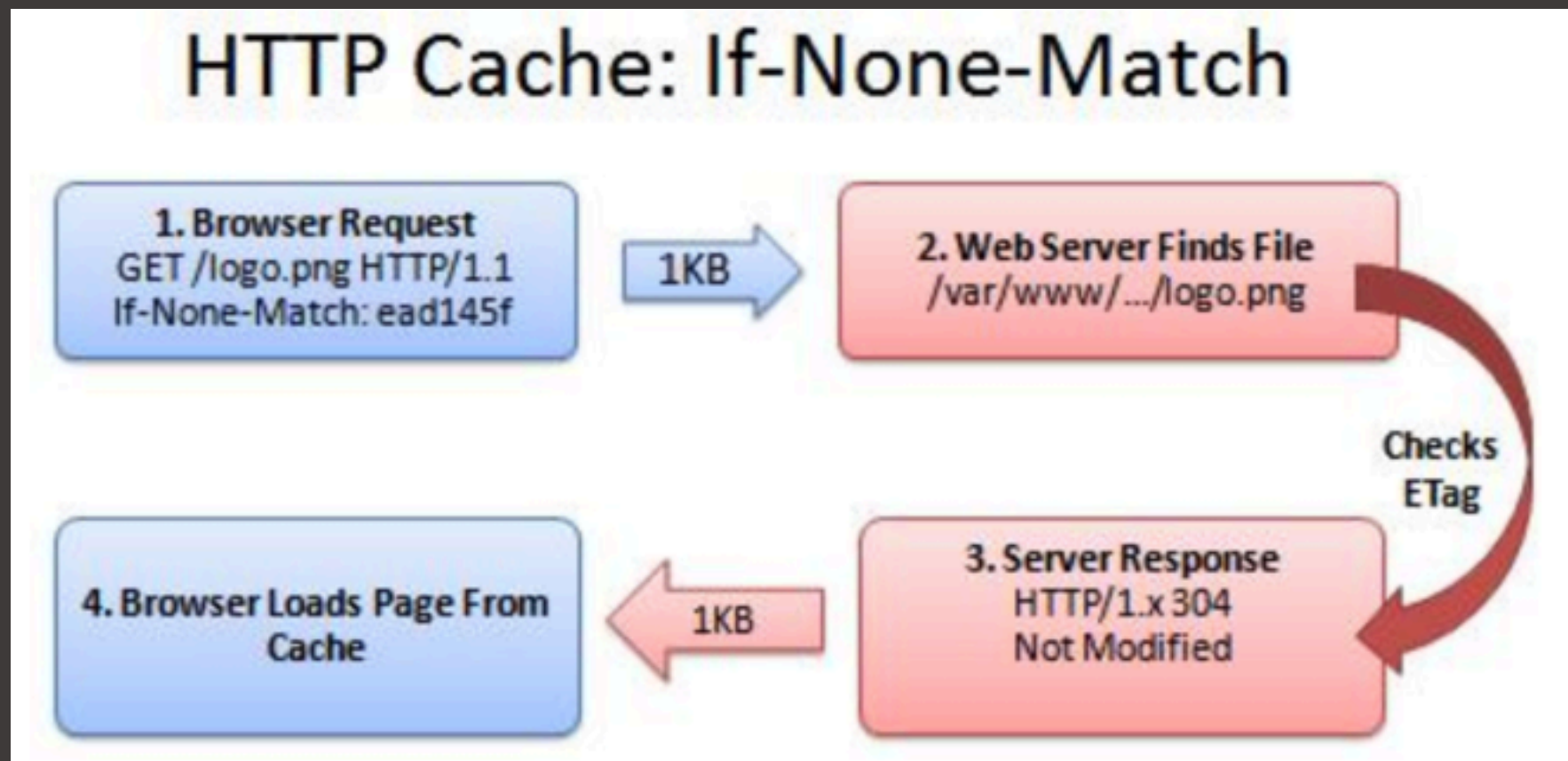
if-none-match: W/"380f8-xXk3rqkZzARYTEcBGxAp1WnbxP8"

upgrade-insecure-requests: 1

user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) Ap
(KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36

E-tag

可能在同一秒保存两次，也可能服务器上文件修改时间并不一致



分析下

max-age + Last Modified 会怎么样

分析下

max-age + Last Modified + Etag 会怎么样

那么

如果让我们去设计一个资源缓存方案，要怎么做呢？

资源缓存方案

1 HTML

2 CSS、Javascript

3 Image

资源缓存方案

- 1 HTML (no-cache)
- 2 CSS、Javascript (MD5/timestmap/version + 长缓存)
- 3 Image (随机名字 + 长缓存)

二、数据缓存

本地数据缓存

1 Cookie

2 localStorage

3 SeesionStorage

4 IndexedDB

Cookie

存储数据量有限，并且每次都会带到服务器，占用宽带

```
accept-encoding: gzip, deflate, br
accept-language: zh-CN,zh;q=0.8,en;q=0.6
cache-control: max-age=0
cookie: pgv_pvi=251840512; tvfe_boss_uuid=57bf934d0521fd9e; RK=jS/SftE/YX; pac_
uid=1_734205675; pgv_si=s7860182016; LW_uid=u1i5J0e3C2h3g8j5J0V601L9F2; eas_si
d=a115k0a3g2i3Q8T540s6L1k9C8; p_uin=o0734205675; p_skey=*SJK*GeS80n-0ByxIrGGjY
*5iL9GiTJ4NPQJja7Mmm4_; pt4_token=Fzvc-gorkgW8EHnhPZIASGlvcQbYQGgUZoa0J1ICKoQ
_; rv2=80F26E9BAA13A780259E322EB5E1033AB0A30B510A007D79FE; property20=E74A988E
425AC91058E1252BABADD6BBAD7DEDB3A57F5D3994A215FAA167F715C2700E199938D129; qqmu
sic_uin=; qqmusic_key=; qqmusic_fromtag=; LW_sid=31P500Z441Q7X0M7S5b8J5w4X7; i
swebp=1; qv_als=g8mYxN0uSB5tyUWVA11505999777AVuZrw==; o_cookie=734205675; qm_a
uthimgs_id=1; qm_verifyimagesession=h01f483194679207034df0ee9d2832f7306a2870d5
c2380bd3990ea9ae685ee04309bc5a6ce1b7f714a; ptisp=cm; ptcz=2a1f195ae54693b61e4b
72e8760527e92c335a411bb72bd87a403c81f98823fe; pt2gguin=o0734205675; uin=o07342
05675; skey=@WzCo5RVb4; enc_uin=M19HgYvTs_9d2X0Mlki7gg; course_origin=[{"cid":
163603,"ext":{"pagelocation":"newagencyhome,1.10"}}]; pgv_info=ssid=s557227165
1&pgvReferrer=http://tiem-cdn.qq.com/html/mms/nativeads/game_rank.html; ts_ref
er=imweb.ke.qq.com/; pgv_pvid=6707401337; ts_uid=9213423499; webwx_data_ticket
=gSdp0BacJCa+NjvLWdjGmz6u; _pathcode=0.9115907389153888; tdw_data={"ver4":"4
1","ver5":"","ver6":"","refer":"www.baidu.com","from_channel":"41","path":"a-
0.9115907389153888","uin":734205675}; ke_login_type=1; tdw_first_visited=1
```

localStorage

有大小限制，数据不安全，不应当把重要数据存在这

localStorage

其实如果真使用起来你会发现更多有趣的东西

- 1 是把多个数据存在一个key好？还是分开多个key存储比较好？
- 2 localStorage 存爆了应该怎么办
- 3 怎么解决数据需要踢出，用什么方式踢出？

sessionStorage

很少使用的东西

IndexedDB

持续化存储，可在 Web Worker 中使用

App Cache

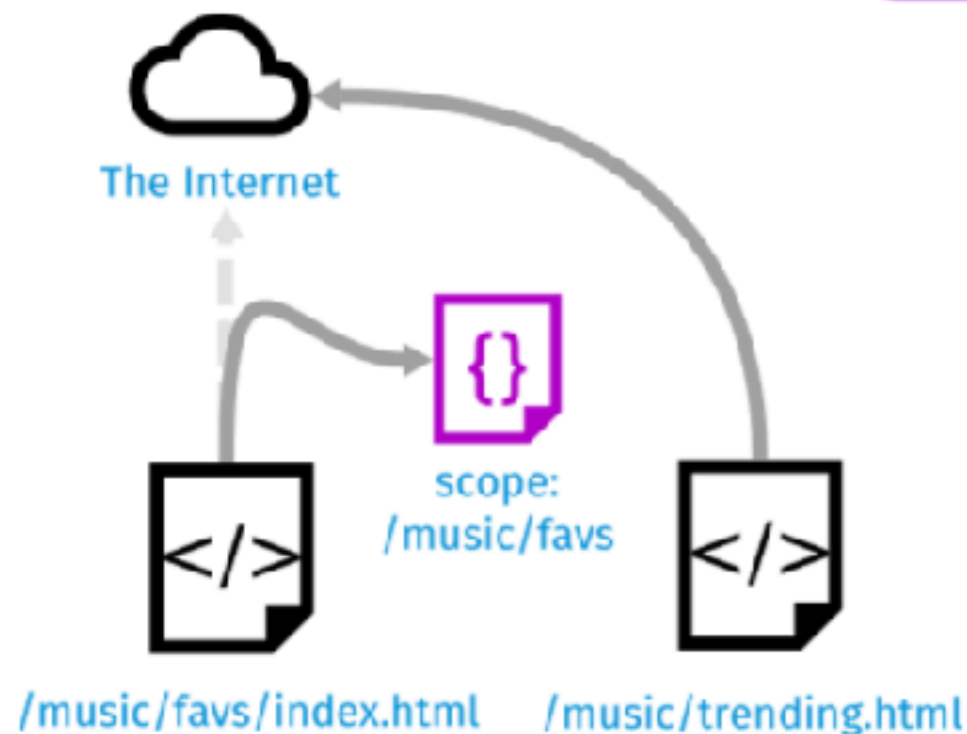
反正是大坑货，不要用就是了

- 1 可能会缓存错误页面，甚至缓存运营商的劫持广告
- 2 更新中有一个文件失败了，就会全部退回上一个版本
- 3 由于是 Lazy 更新，如果后台接口有 break change 将会是大麻烦
- 4 manifest 文件也有可能不小心被缓存了

Service Worker

App Cache 太失败，以至于 W3C 终于明白用户需要更细致的缓存控制

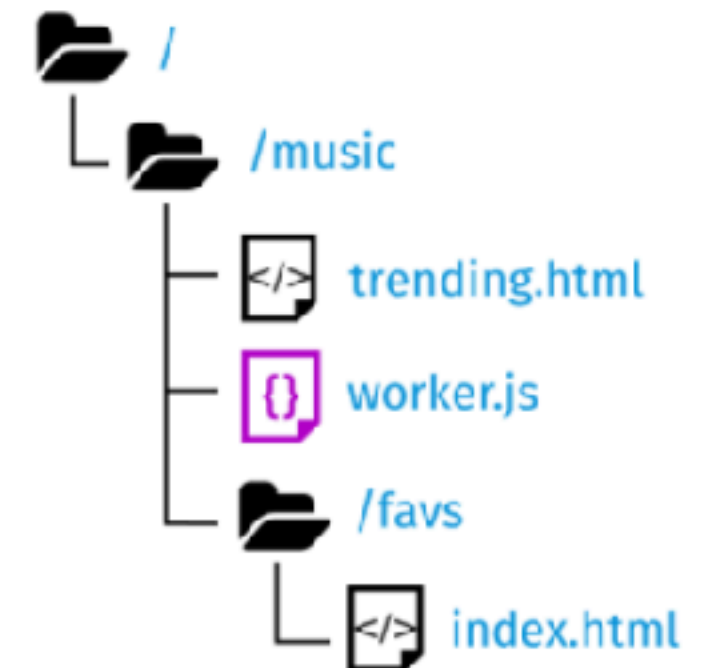
Important!



The service worker will catch requests from the clients **under scope only**.



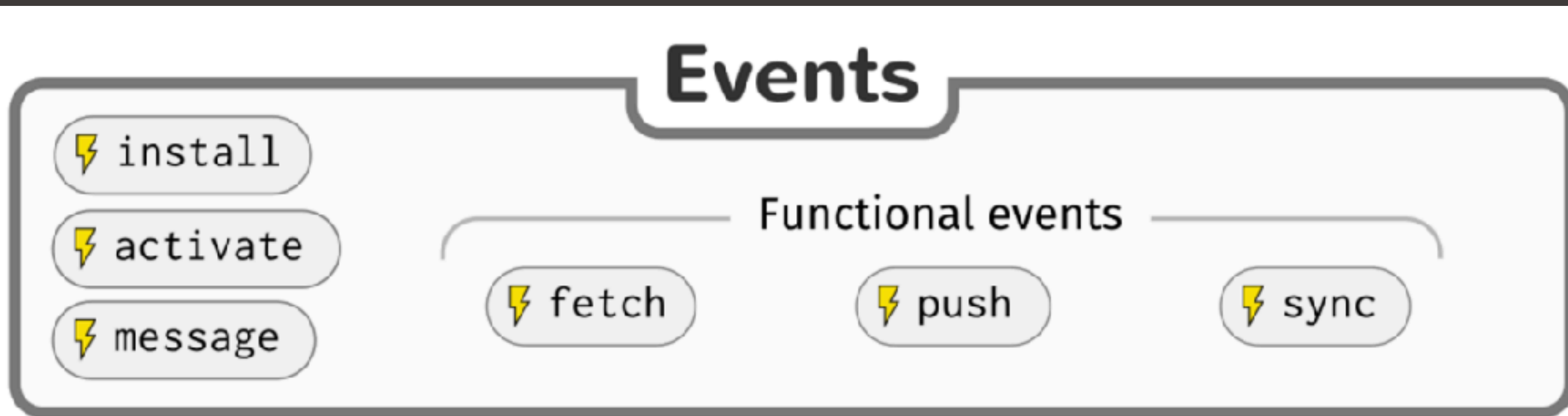
The service worker must be served over **https**.



The **max scope** for a service worker is the location of the worker.

Service Worker

App Cache 太失败，以至于 W3C 终于明白用户需要更细致的缓存控制



<https://serviceworker.rs/>

那么

加上这些技术，我们又能对缓存做什么优化呢？

缓存设计

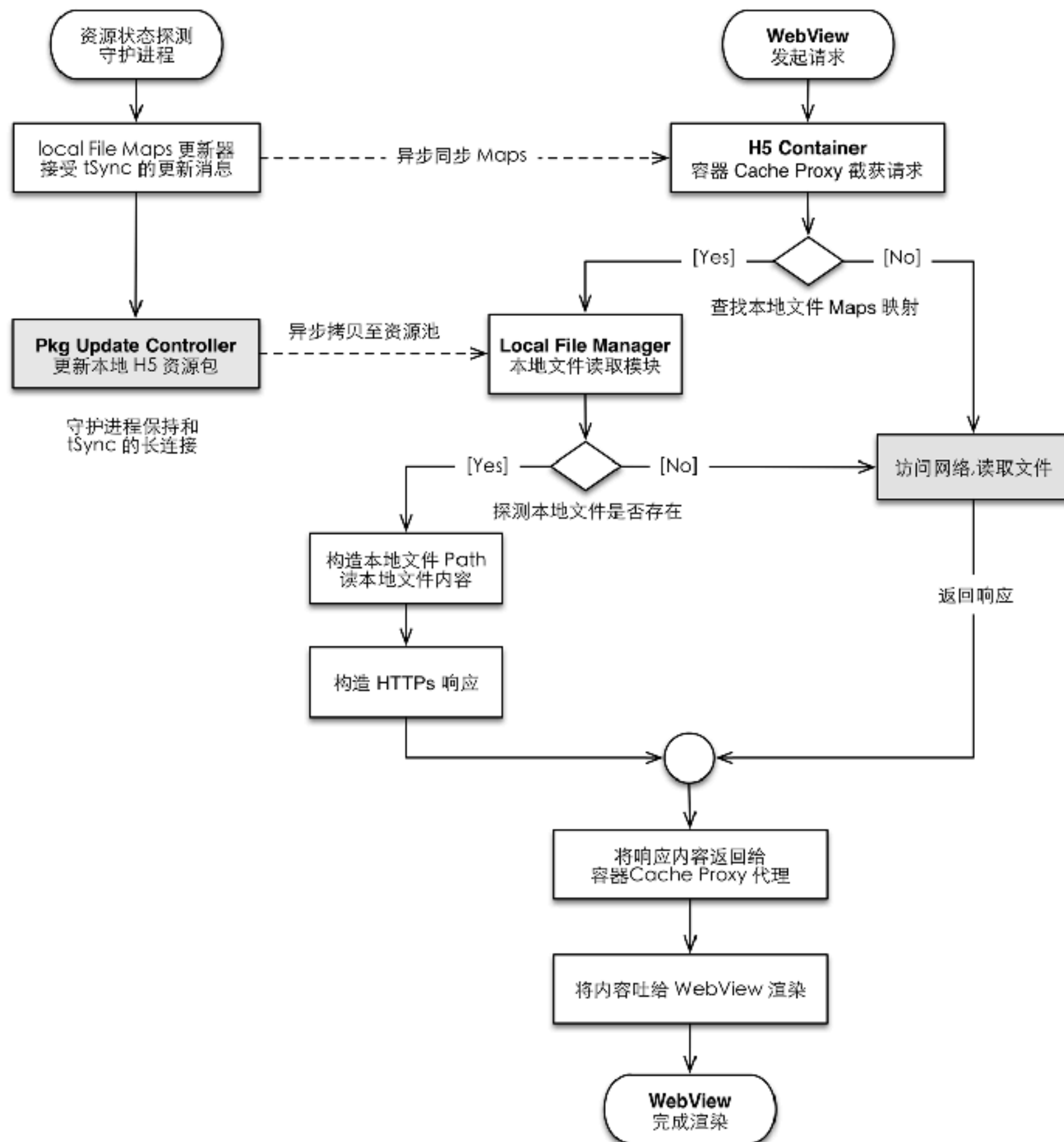
- 1 sessionStorage? localStorage? IndexedDB?
- 2 用什么算法踢出数据?
- 3 数据缓存时间能不能让后台返回?
- 4 能不能用 Service Worker 对缓存做精确控制?
- 5 存 DO(Data Object)? 还是 VO(View Object)?

三、Hybrid App 缓存技术

离线包技术

大公司用了很多年了

<https://yq.aliyun.com/articles/2939>





线上活动页

①

线上资源抓取



o2o 定时程序



②

生成增量包

clam & git-diff



增量包 & 全量包

9ccf5202a74...6dceb76.zip
2015.1014.121421.zip
2015.1014.134605.zip
2015.1014.141237.zip
2015.1014.145649.zip
2015.1014.153704.zip
2015.1014.160828.zip
2015.1014.183027.zip
2015.1014.200212.zip

信鸽平台



tSync Server Sandbox

⑤

tSync 业务服务网关



端信息更新



tSync Server

⑥

tSync 消息推送



手机终端

tSync SDK

⑦

增量包更新



CDN

④

git Hook: 通知 tSync Server 更新消息

③


资源包部署



Gitlab



信鸽平台


拔赤

[简介](#)

[H5 离线包推送](#)

[活动离线化管理](#)

[页面管理](#)

[页面监控](#)

[配置管理](#)

[用户管理](#)

[自动脚本配置](#)

© 2015 阿里旅行·去啊前端团队

技术支持: 弘树

活动离线化管理 > 页面管理

标题	URL	提交者	提交时间	状态	操作
双11环游世界	http://quamarket.m.taobao.com/markets/h5/15qa1111/hc/index-m	弘树	2015-10-13 10:53:18	在线	  下线
我的双11	http://quamarket.m.taobao.com/markets/h5/15qa1111/hc/myCenter	弘树	2015-10-15 19:49:50	在线	  下线
双11玩转中国	http://quamarket.m.taobao.com/markets/h5/15qa1111/hc/index-ne-m?sprn=181.7788770.38200.2	弘树	2015-10-16 17:17:06	在线	  下线
双11酒店	https://quamarket.m.taobao.com/markets/h5/15qa1111/hc/hotel-m	弘树	2015-10-22 12:11:41	在线	  下线
双11未来景区	https://quamarket.m.taobao.com/markets/h5/15qa1111/hc/ticket-brand	弘树	2015-10-22 12:12:33	在线	  下线
双11分期购	https://quamarket.m.taobao.com/markets/h5/15qa1111/hc/tqg-m	弘树	2015-10-22 12:12:57	在线	  下线
双11港台站	https://quamarket.m.taobao.com/markets/h5/hongkong	弘树	2015-10-22 14:08:53	在线	  下线
双11旅行红包	http://quamarket.m.taobao.com/markets/h5/15qa1111/hd/lottery-tianmao	弘树	2015-10-26 12:59:43	在线	  下线
双11分会场-门票品牌红包	http://quamarket.m.taobao.com/markets/h5/15qa1111/hc/ticket-redpack	弘树	2015-10-29 16:39:02	在线	  下线
落跑去台湾	http://h5.m.taobao.com/market/trip/act/luopaobaaiqing.html	弘树	2015-08-28 16:06:12	已下线	  上线

提交新页面

立即同步

☒ 监控中

上一页

1

2

3

4

5

6

下一页

Hybrid Cache

localStorage 不够用? Native 给你提供更多缓存。当然这里也发生过一些安全风险问题

空间的极致优化

充分利用 Native 的 Socket 通道做一些猥琐的事情



THANKS!



扫码了解更多