

SDK

技术选型:

SessionStorage,userData,Cookie

SessionStorage:

临时存储神奇: SessionStorage

优点: 临时存储神奇, 关闭页面标签自动回收, 不可以跨页面交互。

缺点: 临时成也萧何败萧何, 因为是临时所以不能存储持久化的东西。

userData:

优点:

缺点: 存储限制太小, 单个文件限制是128KB, 一个域名下总共可保存1024KB的文件, 文件个数应该没有限制。在受限站点里这两个值分别是64KB和640KB。

Cookie: 兼容性最好的本地存储

优点: 兼容性最好, 几乎所有浏览器都支持

缺点: 大小有限制, 而且每次发送请求, 请求头都会带着cookie一起发送过去, 现在基本大多数登录的合法性验证都是用cookie验证的。

openDatabase: 没有他做不了的

优点: 就是一个完整的数据库。

缺点: 对于前端来说, 成本高。

常用浏览器的存储大小(极限值):

IE 9 > 4999995 + 5 = 5000000

firefox 22.0 > 5242875 + 5 = 5242880

chrome 28.0 > 2621435 + 5 = 2621440

safari 5.1 > 2621435 + 5 = 2621440

opera 12.15 > 5M (超出则会弹出允许请求更多空间的对话框)

目标: 提取用户第一时间看到的, 提取页面第一时间执行的。

4-99

1、初始化JS。。

。。。。。

目的：

1. 可以拉取更新的新资源
2. 可以存储拉取下来的资源
3. 可以根据版本迭代，置换过期的资源



功能拆解：

1. 网络交互功能，XHR
2. 本地存储功能，localSDK
3. 缓存展示能力，浏览器的dom解析器

网络交互功能：

- 1.create：创建xhr对象
- 2.readystate：回调函数处理
- 3.para：参数转换
- 4.get：请求发送

拆解模块细化

拆解模块细化-存储读取能力

- resourceVersion：维护本地缓存版本，更新依据
- resourceJavascriptList：需要缓存的文件列表
- needUpdate：检测文件是否需要更新的方法
- isIE：判断文件是否为IE的方法
- checkHedge：检测本地缓存是否溢出
- startup：启动方法，也是读取本地缓存
- save：保存缓存与startup相对应

如果想自动更新：

- 1、此处需要一个访问接口，这个接口的作用是判断我们的资源是否需要更新

待优化：

- 1、做一个自动获取版本号的功能