

页面返回刷新

背景

两个后端渲染的页面

a.html



b.html



a.html



背景

一、

a.html



history:1

b.html



history:2

a.html



history:3

背景

a.html



history:1

b.html



history:2

a.html



history:3

背景

二、

a.html



b.html



a.html



location.href

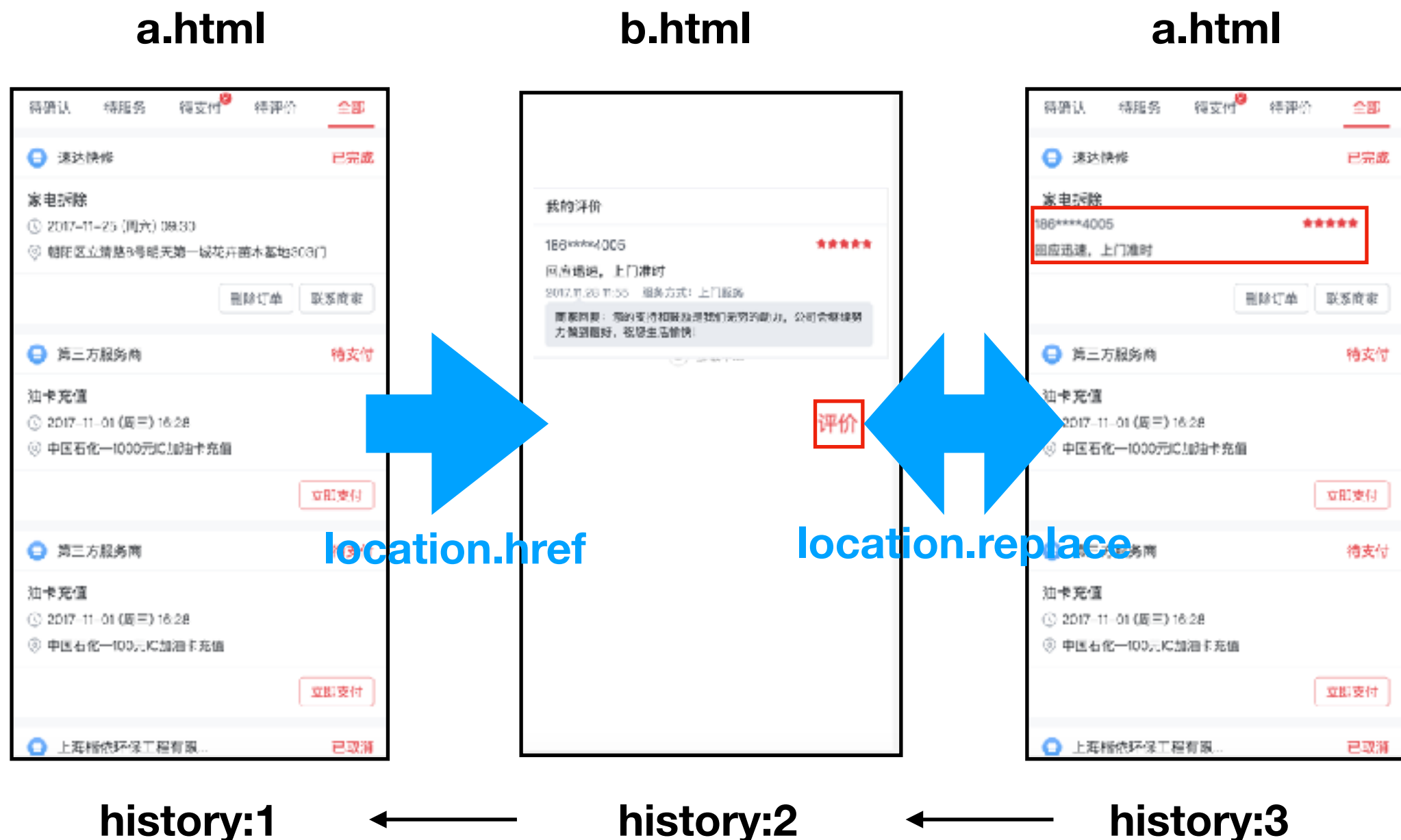
location.replace

history:1

history:2

背景

问题：到家App Android端，replace仍然会保留之前页面的history



背景

三、

a.html



b.html



a.html



location.replace

location.replace

history:1

背景

a.html



b.html

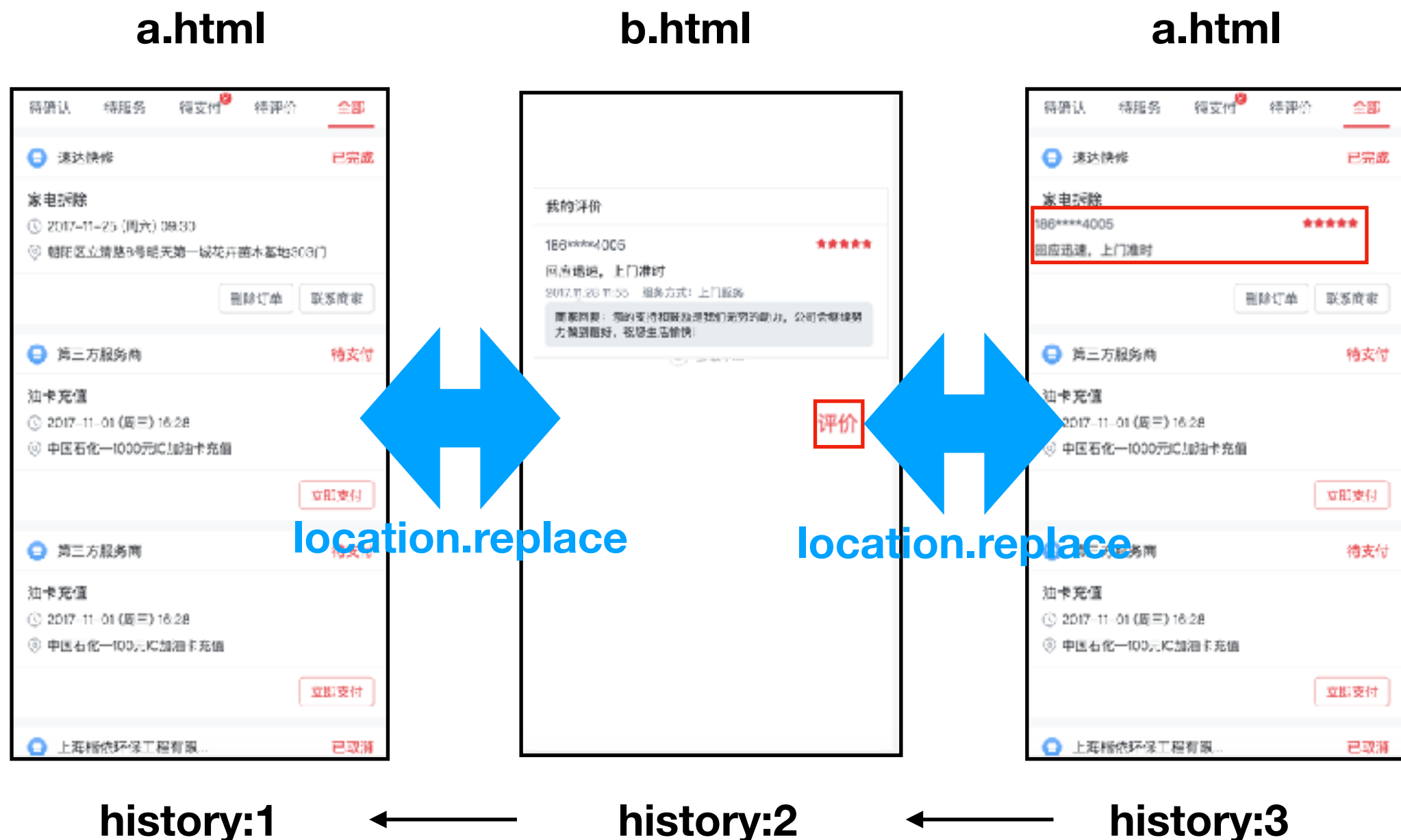


location.replace

history:1

背景

问题too：到家App Android端，replace仍然会保留之前页面的history

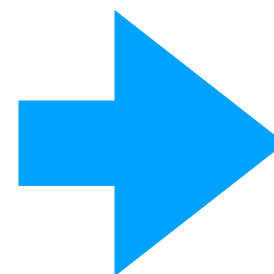


方法

a.html



history:1



location.href

b.html



history:2

方法

a.html



history:1


location.href


history.back()

b.html

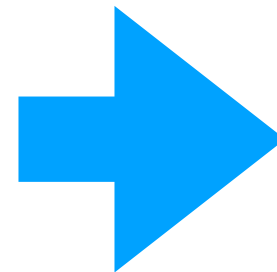


现实

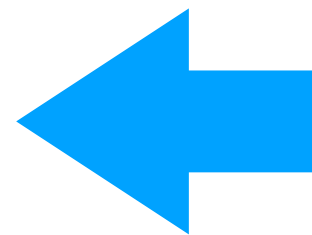
a.html



history:1



location.href



history.back()

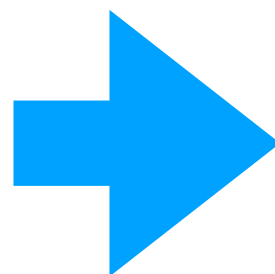
b.html



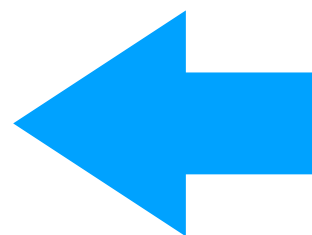
现实

浏览器强大的缓存能力

a.html



location.href



history.back()

b.html



出路

可以检测到页面的返回，并做刷新

a.html



b.html



location.href

history.back()

解决方案

- pageshow事件
- 借助重新执行js对比Html版本号
- Page Visibility API

解决方案

A

服务器渲染时间: 22:30:24:282

[转跳到B页面](#)

pageshow 初始

js 初始

visibilitychange 初始

B

服务器渲染时间: 22:31:6:890

[返回A页面](#)

A

服务器渲染时间: 22:30:24:282

[转跳到B页面](#)

pageshow 初始

js 初始

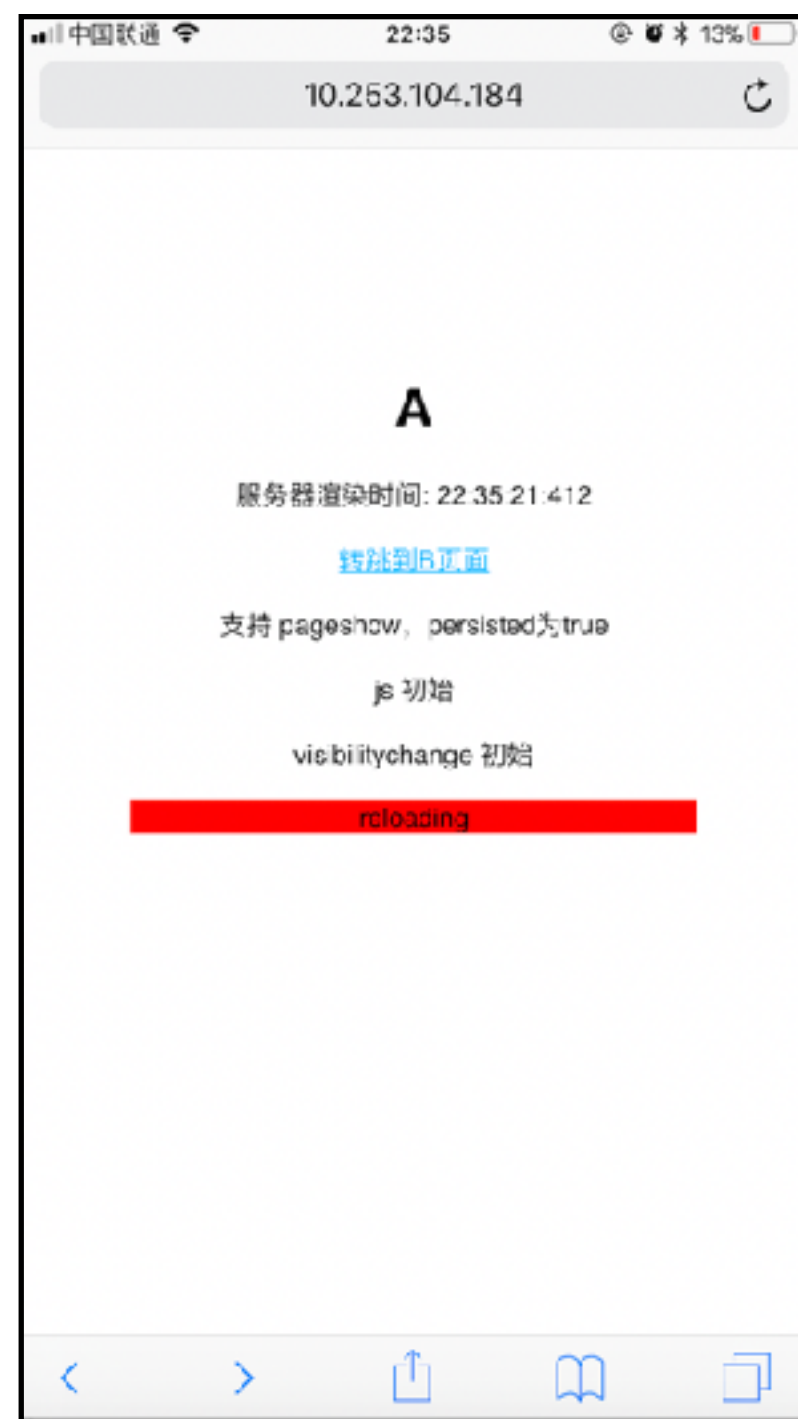
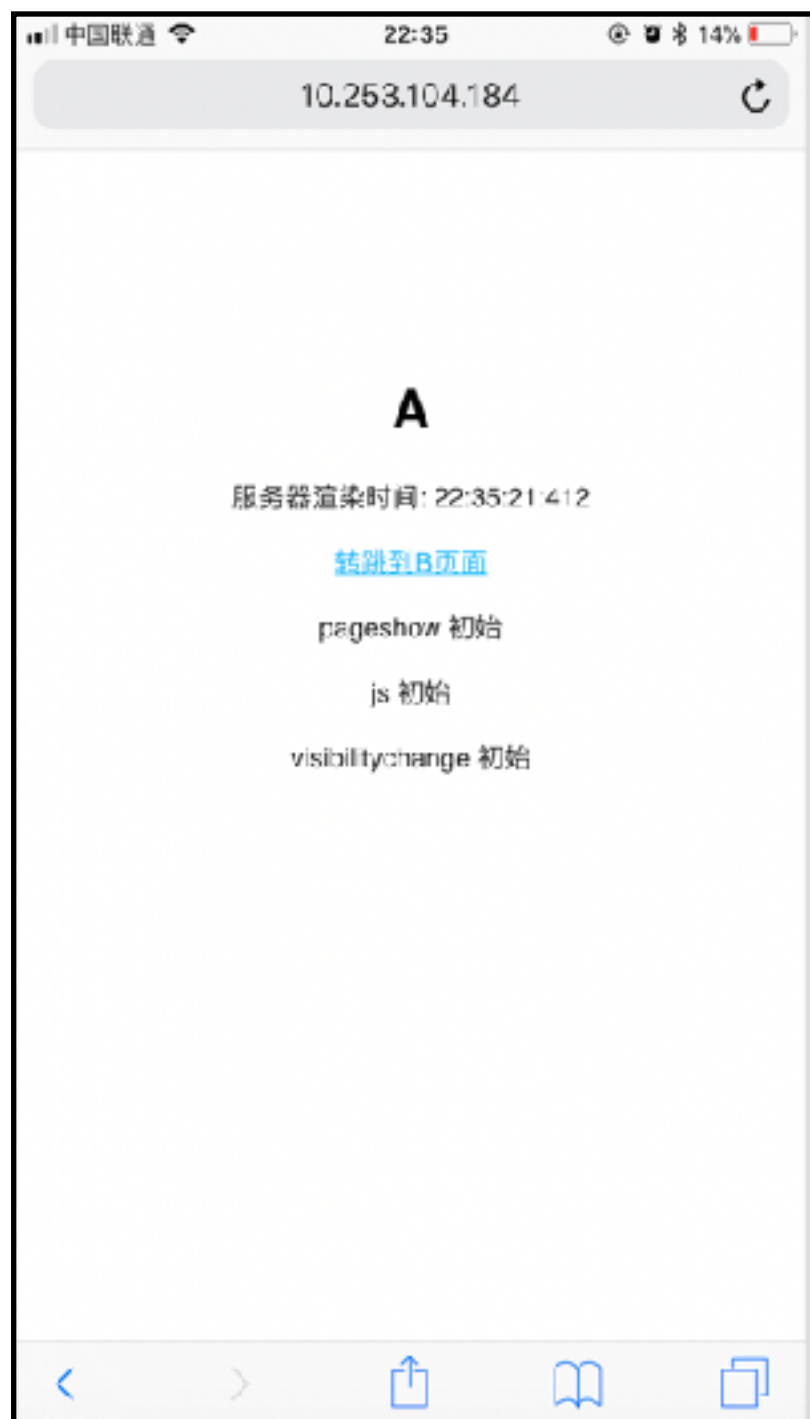
visibilitychange 初始

pageShow事件

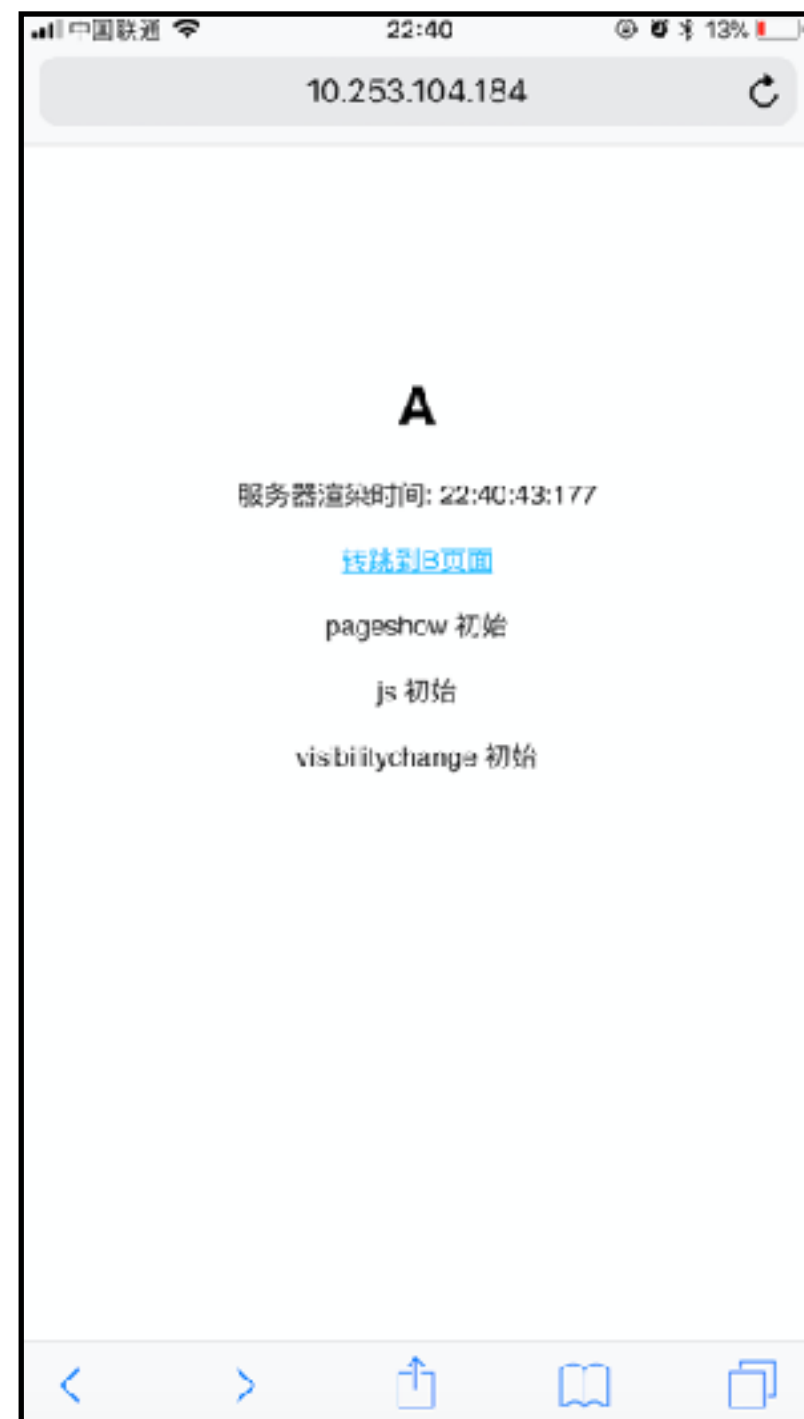
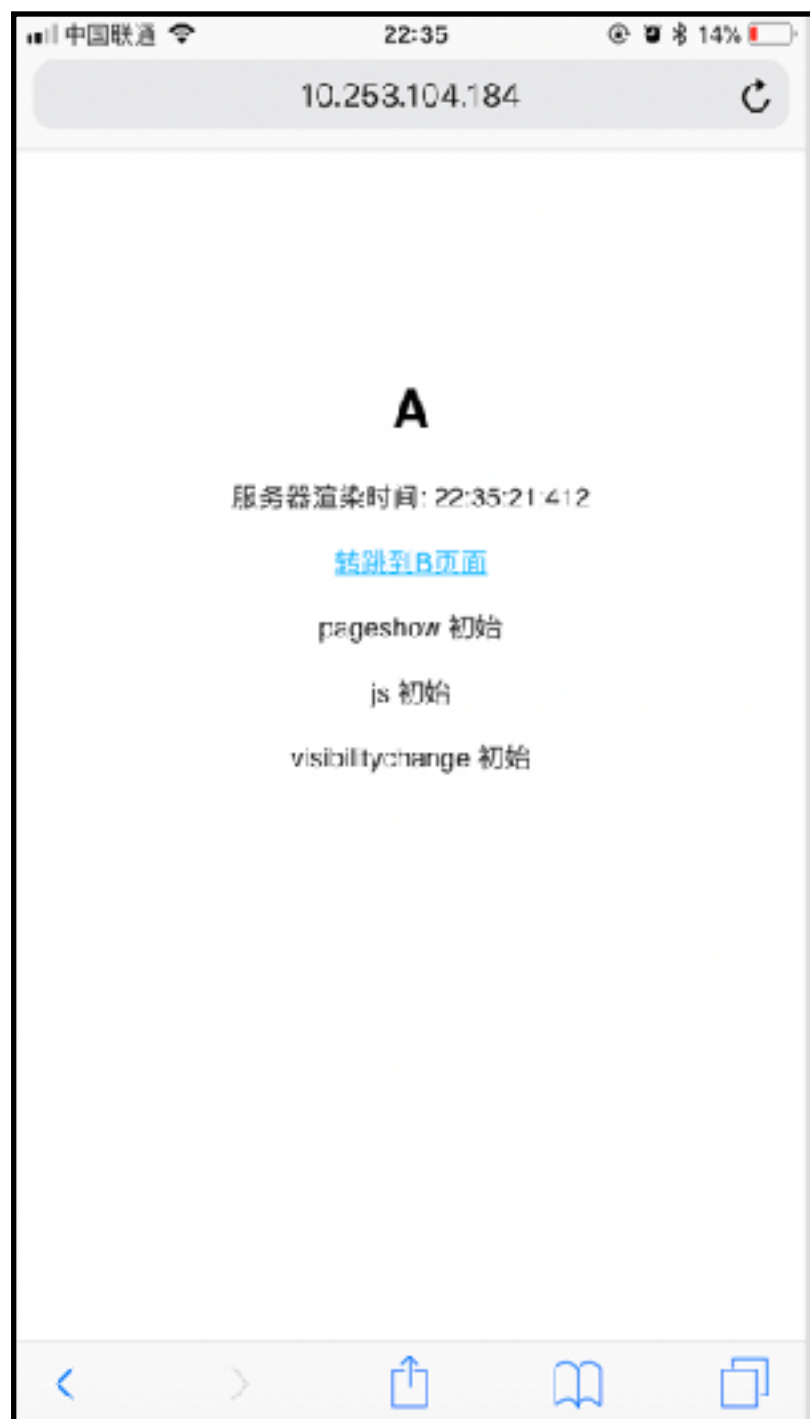
在页面显示时触发，无论页面是否来自bfcache

```
window.addEventListener('pageshow', function (evt) {  
    if(evt.persisted){  
        document.getElementById("pageshow").innerHTML = '支持 pageshow, persisted为true';  
        toReload();  
    }  
});
```

pageShow事件



pageShow事件



pageShow事件

iOS：微信、safari、Chrome、Firefox、Yandex

Android：QQ浏览器、Firefox、UC、手机百度

这些浏览器中的A页面会完全地缓存下来，时刻监听pageshow/pagehide，当从B返回到A页面时，不会重新请求html文档，也不会重新执行js

借助重新执行js对比Html版本号

有些浏览器不支持pageshow/pagehide

但这些浏览器大部分都会在A页面重新被展示时，虽然不会拉取html文档，但是会重新执行一遍A页面的js。这时我们可以借助重新执行js的机会，比较文档版本号（我们自己定义的），然后在判断是否是旧的A页面时，reload页面

借助重新执行js对比Html版本号

Server

```
<input type="hidden" id="SERVER_TIME" value="{{time}}"/>
```

Front

```
//每次webview重新打开H5首页,就把server time记录本地存储
var SERVER_TIME = document.getElementById("SERVER_TIME");
var REMOTE_VER = SERVER_TIME && SERVER_TIME.value;
if(REMOTE_VER){
    var LOCAL_VER = sessionStorage && sessionStorage.PAGEVERSION;

    if(LOCAL_VER && parseInt(LOCAL_VER) >= parseInt(REMOTE_VER)){
        //说明html是从本地缓存中读取的
        toReload();
    }else{
        //说明html是从server端重新生成的,更新LOCAL_VER
        sessionStorage.PAGEVERSION = REMOTE_VER;
    }
}
```

借助重新执行js对比Html版本号

A

服务器渲染时间: 22:24:14:782

[转跳到B页面](#)

pageshow 初始

js 初始

visibilitychange 初始

B

服务器渲染时间: 22:26:26:210

[返回A页面](#)

A

服务器渲染时间: 22:24:14:782

[转跳到B页面](#)

pageshow 初始

js重新执行了(22:26:46:432)

visibilitychange 初始

reloading

借助重新执行js对比Html版本号

A

服务器渲染时间: 22:24:14:782

[转跳到B页面](#)

pageshow 初始

js 初始

visibilitychange 初始

B

服务器渲染时间: 22:26:26:210

[返回A页面](#)

A

服务器渲染时间: 22:26:49:443

[转跳到B页面](#)

pageshow 初始

js 初始

visibilitychange 初始

借助重新执行js对比Html版本号

iOS: UC、手机百度

Android: 微信、Chrome、Yandex

回退后不重新请求html, 重新执行js

Page Visibility API

新开Webview来实现location.href ?

借助与浏览器Tab相关的[Page Visibility API](#)，含义大概为当用户最小化页面或者切换到另一个Tab时，将触发visibilitychange方法，并可以通过判断此时页面展示相关的属性 (`document.hidden`、`document.visibilityState`) 来确定页面的显示状态。

```
function handleVisibilityChange() {  
  if (document.hidden) {  
    document.getElementById("visibilitychange").innerHTML = 'visibilitychange hidden';  
  } else {  
    document.getElementById("visibilitychange").innerHTML = 'visibilitychange show';  
    toReload();  
  }  
}  
document.addEventListener("visibilitychange", handleVisibilityChange, false);
```

Page Visibility API

iOS: QQ浏览器

Android: 小米自带浏览器

一览

	iOS	Android	说明
pageshow/pagehide	微信 safari Chrome Firefox Yandex	QQ浏览器 Firefox UC 手机百度	回退后不重新请求html，不重新执行js
重新执行js对比Html文档版本号	UC 手机百度	微信 Chrome Yandex	回退后不重新请求html，重新执行js
Page Visibility API	QQ浏览器	小米自带浏览器	回退后不重新请求html，不重新执行js
三种方法都不支持的情况	-	-	

一览

	iOS	Android	说明
pageshow/pagehide	微信 safari Chrome Firefox Yandex	QQ浏览器 Firefox UC 手机百度	回退后不重新请求html，不重新执行js
重新执行js对比Html文档版本号	UC 手机百度	微信 Chrome Yandex	回退后不重新请求html，重新执行js
Page Visibility API	QQ浏览器	小米自带浏览器	回退后不重新请求html，不重新执行js
三种方法都不支持的情况	-	-	

【Response Header】
cache: no-store

一览

	iOS	Android	说明
pageshow/pagehide	微信 safari Chrome Firefox Yandex	QQ浏览器 Firefox UC 手机百度	回退后不重新请求html，不重新执行js
重新执行js对比Html文档版本号	UC 手机百度	微信 Chrome Yandex	回退后不重新请求html，重新执行js
Page Visibility API	QQ浏览器	小米自带浏览器	回退后不重新请求html，不重新执行js
三种方法都不支持的情况	-	-	

优点
&
优点的局限性

【Response Header】
cache: no-store

题目

刚刚说的是后端渲染的页面，如果是Vue渲染的页面呢？

讨论

- 后端渲染的页面
- 后端给pageData，前端渲染的页面
- 前端异步接口请求数据，前端渲染的页面

讨论

- 后端渲染的页面
- 后端给pageData，前端渲染的页面
- 前端异步接口请求数据，前端渲染的页面
- 单页应用

Case 1

异步接口获取数据，Vue渲染

a.html



b.html



a.html



Case 2

2017.12.27 服务方式: 上门服务

商家回复: 感谢您对师傅工的认可, 我们的专业您放心, 祝您生活愉快!

糖糖
★★★★★

以为128能解决的事情, 多花了几百。没想到在58也遇到这样的事情。我不相信我家的堵塞有多严重, 必须要加你们的特殊药水才能通。药水就2分钟软化。前后搞好就十来分钟, 到底是堵严重还是恶意另外收费。之前不知道物业有疏通管道服务, 就30块钱, 下次绝不会再在58找。

2017.12.26 服务方式: 上门服务

浏览 1次

137****4715
★★★★★

默认好评~

2017.12.26 服务方式: 上门服务

商家回复: 感谢您对师傅服务的满意, 谢谢! 祝您生活愉快!

糖糖
★★★★★

以为128能解决的事情, 多花了几百。没想到在58也遇到这样的事情。我不相信我家的堵塞有多严重, 必须要加你们的特殊药水才能通。药水就2分钟软化。前后搞好就十来分钟, 到底是堵严重还是恶意另外收费。之前不知道物业有疏通管道服务, 就30块钱, 下次绝不会再在58找。

2017.12.26 服务方式: 上门服务

 专业下水管道疏通清理 快速
上门 30元为预约金具体...
¥30

浏览4次

2017.12.27 服务方式: 上门服务

商家回复: 感谢您对师傅工的认可, 我们的专业您放心, 祝您生活愉快!

糖糖
★★★★★

以为128能解决的事情, 多花了几百。没想到在58也遇到这样的事情。我不相信我家的堵塞有多严重, 必须要加你们的特殊药水才能通。药水就2分钟软化。前后搞好就十来分钟, 到底是堵严重还是恶意另外收费。之前不知道物业有疏通管道服务, 就30块钱, 下次绝不会再在58找。

2017.12.26 服务方式: 上门服务

浏览 1次

137****4715
★★★★★

默认好评~


2017.12.26 服务方式: 上门服务

商家回复: 感谢您对师傅服务的满意, 谢谢! 祝您生活愉快!

Case 2

商家回复: 谢谢您对师傅工的认可, 我们的专业您放心, 祝您生活愉快!

👍 赞




糖糖
★★★★★

以为128能解决的事情, 多花了几百。没想到在58也遇到这样的事情。我不相信我家的堵塞有多严重, 必须要加你们的特殊药水才能通。药水就2分钟软化。前后搞好就十来分钟, 到底是堵严重还是恶意另外收费。之前不知道物业有疏通管道服务, 就30块钱, 下次绝不会在58找。

2017.12.26 服务方式: 上门服务

浏览 2次

👍 赞




137****4715
★★★★★

默认好评~


2017.12.26 服务方式: 上门服务

商家回复: 谢谢您对师傅服务的满意, 谢谢! 祝您生活愉快!

👍 赞



153****8880
★★★★★



糖糖
★★★★★

以为128能解决的事情, 多花了几百。没想到在58也遇到这样的事情。我不相信我家的堵塞有多严重, 必须要加你们的特殊药水才能通。药水就2分钟软化。前后搞好就十来分钟, 到底是堵严重还是恶意另外收费。之前不知道物业有疏通管道服务, 就30块钱, 下次绝不会在58找。

2017.12.26 服务方式: 上门服务



专业下水管道疏通清理 快速上门 30元为预约金具...
¥30

浏览3次

👍 赞

默认好评~

2017.12.27 服务方式: 上门服务

商家回复: 谢谢你对师傅服务的满意! 谢谢, 祝你生活愉快!

👍 赞



158****8898
★★★★★

默认好评~

2017.12.27 服务方式: 上门服务

商家回复: 谢谢您对师傅工的认可, 我们的专业您放心, 祝您生活愉快!

👍 赞



138****9728
★★★★★

默认好评~

2017.12.27 服务方式: 上门服务

商家回复: 谢谢您对师傅工的认可, 我们的专业您放心, 祝您生活愉快!

👍 赞

点击加载更多

Todo

58到家App Webkit

```
<script>
  var userAgent = navigator.userAgent;
  var cookiestr = document.cookie;

  if (userAgent.indexOf('cdvsupport') >= 0 || cookiestr.indexOf('cdvsupport') >= 0) {
    document.write("<script type='text/javascript' src='nacodovafile/js/cdplugins.js#nacodovafile'></script>");
    document.write("<script type='text/javascript' src='nacodovafile/js/cordova.js#nacodovafile'></script>");
  }
</script>
```

```
document.addEventListener("refresh", function () {
  }, false);
```