

WKWebView相比于UIWebView浏览器之间内核引擎的区别

1 2018.04.15 11:16:26 字数 1563 阅读 3882

翻译文，原文地址

WKWebView: Differences from UIWebView browsing engine

优点

多进程，在app的主进程之外执行

使用更快的Nitro JavaScript引擎

异步执行处理JavaScript

消除某些触摸延迟

支持服务端的身份校验

支持对错误的自签名安全证书和证书进行身份验证

•

问题

需要iOS9或更高版本(WKWebView在iOS8引入，但是很多功能，支持比较全面在iOS9以后的版本)

不支持通过AJAX请求本地存储的文件

不支持"Accept Cookies"的设置

不支持"Advanced Cache Settings"(高级缓存设置)

App退出会清除HTML5的本地存储的数据

不支持记录WebKit的请求

不能进行截屏操作

优点(Advantages)

多进程，在app的主进程之外执行

WKWebView为多进程组件，也意味着会从App内存中分离内存到单独的进程(Network Process

华为云

0元试用 [4核8G云服务器]

数据库、安全、存储0元体验

立即体验

广告

mdiep

拥有18钻

关注

华为云

0元试用 [4核8G云服务器]

数据库、安全、存储0元体验

立即体验

广告

载页面)

相反的，UIWebView是和app同一个进程，UIWebView加载页面占用的内存被计算为app内存占用的一部分，当app超过了系统分配的内存，则会被操作系统crash。在整个过程中，会经常收到iOS系统的通知用来防止app被系统kill，但是在某些时候，这些通知不够及时，或者根本没有返回通知。

使用更快的Nitro JavaScript引擎

WKWebView使用和手机Safari浏览器一样的Nitro JavaScript引擎，相比于UIWebView的JavaScript引擎有了非常重要的性能提升

异步执行处理JavaScript

WKWebView是异步处理app原生代码与JavaScript之间的通信，因此普遍上执行速度会更快。

在实践操作过程中，JavaScript API调用原生(native)中方法不会阻塞线程，等待回调函数的执行。(在JavaScript代码会继续向下执行，而回调函数会由native端异步去回调)。举一个例子，之前一个"Save Data"的操作如下：

```
1 // JavaScript code - 笔者注
2 // 注释 - 笔者注
3
4 var filenameID;
5 function getFilenameID() {
6     // 向native端发起请求，获取kioskId，结果返回由callback方式返回
7     window.kp_requestKioskId("kp_requestKioskId_callback");
8 }
9 // callback回调函数 - 由native端发起
10 function kp_requestKioskId_callback(kioskId) {
11     filenameID = kioskId.split(" ").join("");
12 }
13 // kp_FileAPI_writeToFile方法不会等待kp_requestKioskId_callback回调函数执行，此时filenameID为undefined
14 function saveData(fileName, data) {
15     getFilenameID();
16     kp_FileAPI_writeToFile(filenameID + ".xls", data, "writeFile_callback");
17 }
```

原先的假定是在'saveData'方法被触发时，在'kp_FileAPI_writeToFile'方法调用前，'getFilenameID'方法会返回filenameID

但是，在WKWebView中JavaScript和native代码之间的通信是异步的，'kp_FileAPI_writeToFile'方法被调用之前，'getFilenameID'方法还没有完成(回调还没有被执行-笔者注)，导致的结果是在'kp_FileAPI_writeToFile'中filename为undefined。为了正确的得到filename，必须重构之前的代码，在callback中完成。如下：

```
1 var filenameID;
2 function getFilenameID() {
3     window.kp_requestKioskId("kp_requestKioskId_callback");
4 }
5 function kp_requestKioskId_callback(kioskId) {
6     filenameID = kioskId.split(" ").join("");
7     kp_FileAPI_writeToFile(filenameID + ".xls", data, "writeFile_callback");
8 }
9 function saveData(fileName, data) {
10     getFilenameID();
11 }
```

消除触摸延迟

UIWebView和WKWebView浏览器组件会将触摸事件解释后发送给app，因此，我们无法提高触

摸事件的灵敏度或速度。

在UIWebView上的任何触摸事件会被[延迟300ms](#)，用以判断用户是单击还是双击。这个机制也是那些基于HTML的web app一直不被用户接受的重要原因。

在WKWebView中，测试显示，只有在点击很快(<~125ms)的时候才会添加300ms的延迟，iOS将其解释为更可能是双击“点击缩放”手势的一部分，而不是慢点击（>~125 ms）后。更多细节在[这里](#)

为了消除所有触摸事件（包括快速点击）的触摸延迟，您可以添加[FastClick](#)或另一个消除此延迟的库到您的内容中。

支持服务端的身份校验

与不支持服务器认证校验的UIWebView不同，WKWebView支持服务端校验。实际上，这意味着在使用WKWebView时，可以输入密码保护网站。

支持对错误的自签名安全证书和证书进行身份验证

通过“继续”/“取消”弹出窗口，WKWebView允许您绕过安全证书中的错误（例如，使用自签名证书或过期证书时）。

问题

需要iOS9或更高版本

我们的WKWebView集成仅适用于运行iOS 9或更高版本的设备。虽然WKWebView是在iOS 8中引入的，但在这些版本中存在重大限制，包括无法访问本地存储的文件，我们无法解决此问题，因此此功能不兼容。

不支持AJAX请求到本地存储的文件

WKWebView不允许XHR请求file：// URI，因为这些URI违反了浏览器引擎的跨源资源共享规则。使用这种类型的请求的项目应该远程托管在服务器上，或使用现有的UIWebView浏览引擎。

不支持"Accept Cookies"的设置

虽然WKWebView确实支持使用cookies，但并没有公开选择哪些cookies被源代码接受的能力。这意味着在使用WKWebView浏览引擎时不会应用“接受Cookie”设置。

WKWebView只允许我们访问cookie的名称，而不是附加信息，如创建/过期日期或路径，这使得更难以解决Cookie出现的问题。

不支持"Advanced Cache Settings"(高级缓存设置)

使用WKWebView浏览引擎时，不会应用“缓存源”和“仅通知服务器重定向事件的浏览器”。

App退出会清除HTML5的本地存储的数据

当应用退出并重新启动时，HTML5本地存储将被清除。

不支持记录WebKit的请求

WKWebView发出请求并呈现内容，无法直接访问此类请求，并且无法记录这些请求。

不能进行截屏操作

尽管我们在测试中没有看到使用Kiosk Pro的JavaScript API进行屏幕捕获的任何问题，但其他iOS开发人员报告说屏幕捕获在WKWebView上随机失败。如果截屏的API是app中的关键操作，建议使用现有的UIWebView浏览引擎。

13人点赞 >



mdiep

拥有18钻

关注

iOS



0元试用

数据库、安全、存储0元体验

华为云4核8G云服务器

立即体验

广告

被以下专题收入，发现更多相似内容

+ 收入我的专题

推荐阅读更多精彩内容 >

iOS 开发：『Runtime』详解（一）基础知识

本文首发于我的个人博客：『不羁阁』 文章链接：传送门本文更新时间：2019-08-16 15:50:49本次更新增...

 行走少年郎

swift-defer

来源于喵神的博客:关于 Swift defer 的正确使用 defer简介 从语言设计上来说，defer 的目的就...

 蚂蚁牙齿不黑

iOS13简单适配

iOS13的beta5版本已经出来了，APP适配也应该提上日程了，本文记录下自己在适配时做的一些工作。 Q：我不想...

 yuyangkk

iOS-常用宏定义

自己常用宏定义 /*打印信息*/#ifdef DEBUG#define BRYLog(s, ...) prin...

 Y_3c23

iOS 简单实现倒计时功能

倒计时功能在App中应用非常常见，例如启动微博启动页面的按钮，或者获取验证码计时，这里主要实现倒计时的基本功能 设...

 JQWONG