关于UIWebView的总结

发表于 2012-03-24 20:19

文章目录

前言

今天参加了 Adobe 和 CSDN 组织的一个关于 PhoneGap 的开发讲座，而 PhoneGap 在 iOS 设备上的实现就是通过 UIWebView 控件来展示 html 内容，并且与 native 代码进行交互的。

正好我们在做有道云笔记的 iPad 版，因为我们也是使用 UIWebView 来展示笔记内容，所以也需要做 js 与 native 代码相互调用的事情。所以在这儿顺便总结一下 UIWebView 在使用上的细节，以及谈谈我对 PhoneGap 的看法。

机制

首先我们需要让 UIWebView 加载本地 HTML。使用如下代码完成：

```
NSString * path = [[NSBundle mainBundle] bundlePath];
NSURL * baseURL = [NSURL fileURLWithPath:path];
NSString * htmlFile = [[NSBundle mainBundle] pathForResource:@"index" ofType:@"html"];
NSString * htmlString = [NSString stringWithContentsOfFile:htmlFile encoding:(NSUTF8StringEncoding) error:nil];
[self.webView loadHTMLString:htmlString baseURL:baseURL];
```

接着，我们需要让 js 能够调用 native 端。iOS SDK 并没有原生提供 js 调用 native 代码的 API。但是 UIWebView 的一个 delegate 方法使我们可以做到让 js 需要调用时，通知 native。在 native 执行完相应调用后，可以用 stringByEvaluatingJavaScriptFromString 方法，将执行结果返回给 js。这样，就实现了 js 与 native 代码的相互调用。

以下是 PhoneGap 相关调用的示例代码：

```
// Objective-C 语言
```

```
- (BOOL)webView:(UIWebView *)webView
    shouldStartLoadWithRequest:(NSURLRequest *)request
    navigationType:(UIWebViewNavigationType)navigationType {
       NSURL * url = [request URL];
       if ([[url scheme] isEqualToString:@"gap"]) {
            // 在这里做 js 调 native 的事情
            // ....
            // 做完之后用如下方法调回 js
            [webView stringByEvaluatingJavaScriptFromString:@"alert('done')"];
            return NO;
        }
        return YES;
}
```

具体让 js 通知 native 的方法是让 js 发起一次特殊的网络请求。这里，我们和 PhoneGap 都是使用加载一个隐藏的 iframe 来实现的，通过将 iframe 的 src 指定为一个特殊的 URL，实现在 delegate 方法中截获这次请求。

以下是 PhoneGap 相关调用的示例代码：

```
// Javascript 语言
// 通知 iPhone UIWebView 加载 url 对应的资源
// url 的格式为: gap:something
function loadURL(url) {
    var iFrame;
    iFrame = document.createElement("iframe");
    iFrame.setAttribute("src", url);
    iFrame.setAttribute("style", "display:none;");
    iFrame.setAttribute("height", "0px");
    iFrame.setAttribute("width", "0px");
    iFrame.setAttribute("frameborder", "0");
    document.body.appendChild(iFrame);
    // 发起请求后这个 iFrame 就没用了，所以把它从 dom 上移除掉
    iFrame.parentNode.removeChild(iFrame);
    iFrame = null;
}
```

在这里，可能有些人说，通过改 document.location 也可以达到相同的效果。关于这个，我和 zyc 专门试过，一般情况下，改 document.location 是可以，但是改 document.location 有一个很严重的问题，就是如果我们连续 2 个 js 调 native，连续 2 次改 document.location 的话，在 native 的 delegate 方法中，只能截获后面那次请求，前一次请求由于很快被替换掉，所以被忽略掉了。

我也专门去 Github 上查找相关的开源代码，它们都是用过 iframe 来实现调用的，例如这个：https://github.com/marcuswestin/WebViewJavascriptBridge

关于这个，我也做了一个 Demo 来简单示例，地址如下：https://github.com/tangqiaoboy/UIWebViewSample

参数的传递

以上的示例代码为了讲清楚机制，所以只是示例了最简单的相互调用。但实际上 js 和 native 相互调用时，常常需要传递参数。

例如，有道云笔记 iPad 版用 UIWebView 显示笔记的内容，当用户点击了笔记中的附件，这个时候，js 需要通知 native 到后台下载这个笔记附件，同时通知 js 当前的下载进度。对于这个需求，js 层获得用户点击事件后，就需要把当前点击的附件的 ID 传递给 native，这样 native 才能知道下载哪个附件。

参数传递最简单的方式是将参数作为 url 的一部分，放到 iFrame 的 src 里面。这样 UIWebView 通过截取分析 url 后面的内容即可获得参数。但是这样的问题是，该方法只能传递简单的参数信息，如果参数是一个很复杂的对象，那么这个 url 的编解码将会很复杂。对此，我们的有道云笔记和 PhoneGap 采用了不同的技术方案。

- 我们的技术方案是将参数以 JSON 的形式传递，但是因为要附加在 url 之后，所以我们将 JSON 进行了 Base64 编码，以保证 url 中不会出现一些非法的字符。
- PhoneGap 的技术方案是，也是用 JSON 传递参数，但是将 JSON 放在 UIWebView 中的一个全局数组中，UIWebView 当需要读取参数时，通过读取这个全局数组来获得相应的参数。

相比之下，应该说 PhoneGap 的方案更加全面，适用于多种场景。而我们的方案简洁高效，满足了我们自己产品的需求。

同步和异步

因为 iOS SDK 没有天生支持 js 和 native 相互调用，大家的技术方案都是自己实现的一套调用机制，所以这里面有同步异步的问题。细心的同学就能发现，js 调用 native 是通过插入一个 iframe，这个 iframe 插入完了就完了，执行的结果需要 native 另外用 stringByEvaluatingJavaScriptFromString 方法通知 js，所以这是一个异步的调用。

而 stringByEvaluatingJavaScriptFromString 方法本身会直接返回一个 NSString 类型的执行结果，所以这显然是一个同步调用。

所以 js call native 是异步，native call js 是同步。在处理一些逻辑的时候，不可避免需要考虑这个特点。

这里顺便说一个 android，其实在 android 开发中，js 调 native 是同步的，但是 PhoneGap 为了将自己做成一个跨平台的框架，所以在 android 的 js call native 的 native 端，用 new Thread 新建了一个执行线程，这样把 android 的 js call native 也变成了异步调用。

UIWebView 的问题

线程阻塞问题

我们在开发中发现，当在 native 层调用 stringByEvaluatingJavaScriptFromString 方法时，可能由于 javascript 是单线程的原因，会阻塞原有 js 代码的执行。这里我们的解决办法是在 js 端用 defer 将 iframe 的插入延后执行。

主线程的问题

UIWebView 的 stringByEvaluatingJavaScriptFromString 方法必须是主线程中执行，而主线程的执行时间过长就会 block UI 的更新。所以我们应该尽量让 stringByEvaluatingJavaScriptFromString 方法执行的时间短。

有道云笔记在保存的时候，需要调用 js 获得笔记的完整 html 内容，这个时候如果笔记内容很复杂，就会执行很长一段时间，而因为这个操作必须是主线程执行，所以我们显示 “正在保存” 的 UIAlertView 完全无法正常显示，整个 UI 界面完全卡住了。在新的编辑器里，我们更新了获得 html 内容的代码，才将这个问题解决。

键盘控制

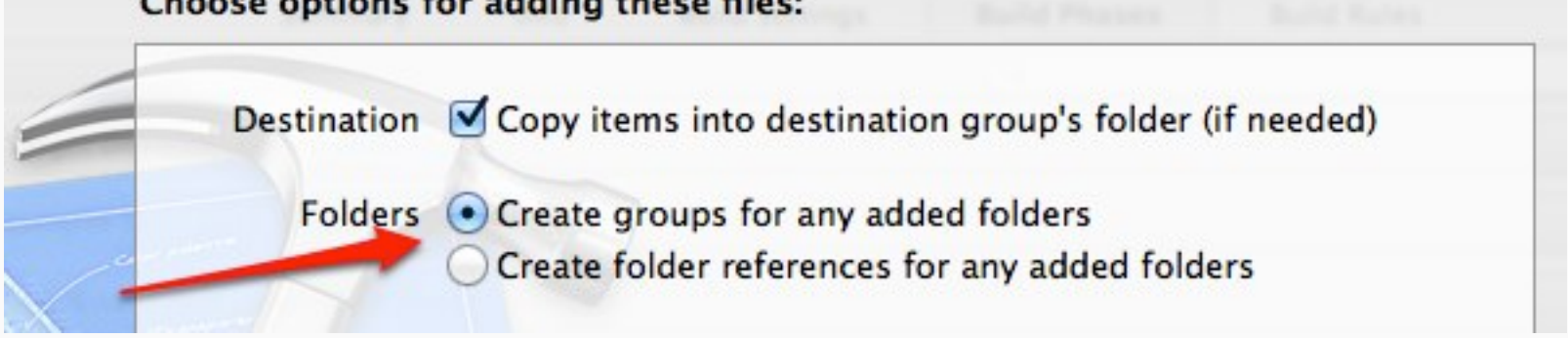
做 iOS 开发的都知道，当我们需要键盘显示在某个控件上时，可以调用 [obj becomeFirstResponder] 方法来让键盘出来，并且光标输入焦点出现在该控件上。

但是这个方法对于 UIWebView 并不可用。也就是说，我们无法通过程序控制让光标输入焦点出现在 UIWebView 上。
关于这个问题，我在 stackoverflow 上专门 问了一下，还是没有得到很好的解决办法。

CommonJS 规范

commonJS 是一个模块块加载的规范。而 AMD 是该规范的一个草案，CommonJS AMD 规范描述了模块化的定义，依赖关系，引用关系以及加载机制，其规范原文在 这里 。它被 requireJS，NodeJs，Dojo，jQuery 等开源框架广泛使用。这里 还有一篇不错的中文介绍文章。

AMD 规范需要用目录层级当作包层次，这一点就象 java 一样。之前我以为 iOS 打包后的 ipa 资源文件中不能有资源目录层级关系，今天在会上问了一下，原来是我自己弄错了。如果需要将目录层级带入 ipa 资源文件中，只需要将该目录拖入工程中，然后选择 “Create groups for any added folders”。如下图所示，这样目录层级能够打包到 ipa 文件中。



调试

在 iOS 设备中调试 javascript 是一件相当苦逼的事情，拿 pw 的话来说：“一下子回到了 ie6 时代”。当然，业界也有一些调试工具可以用的。

我们在开发时主要采用的是 weinre 这个框架。用这个框架，可以做一些基本的调试工作，但是它现在功能还没有象 pc 上的 js 调试器那么强大，例如它不能下断点，另外如果有 js 执行错误，它也无法正确的将错误信息报出。它还有一些 bug，例如在 mac 机下，如果你同时连接了有线网和无线网，那么 weinre 将无法正确地连接到调试页面。

但终究，它是现在业界现存的唯一相对可用的调试工具了。本次的 PhoneGap 讲座的第一位演讲者董龙飞有一篇博客很好地介绍了 weinre 的使用，地址是 这里，推荐感兴趣的同学看看。即使不用 PhoneGap，weinre 也能给你在移动设备上设计网页带来方便。

（2013 年 10 月 22 日更新）：关于调试这一块儿，从 WWDC2012 开始，苹果已经支持用 safari 来连接 iPhone 模拟器里面的 UIWebView 进行调试了，所以调试上已经方便了很多。详细的教程可以查看: WWDC2012 Session 600《Debuging UIWebViews and Websites on iOS》

我对 PhoneGap 的看法

今天的大会上，2 位演讲者把 PhoneGap 吹得相当牛。但是其实真正用过的人才能知道，PhoneGap 还是有相当多的问题的。至少我知道在网易就有一个使用 PhoneGap 而失败的项目，所以我认为 PhoneGap 还是有它相当大的局限性的。

我认为 PhoneGap 有以下 3 大问题：

1. 首先，PhoneGap 的编程语言其实是 javascript，这对于非前端工作者来说，其实学习起来和学习原生的 objective-C 或 Java 编程语言难度差不多，而且由于历史原因，javascript 语言本身的问题比其它语言都多。要想精通 javascript，相当不易。
2. 然后，PhoneGap 的目标是方便地创建跨平台的应用。但是其实苹果和 google 都发布了自己的人机交互指南，有些情况下，苹果的程序和 android 程序有着不同的

交互原则的。象有道云笔记的 iPhone 版 和 android 版，就有着完全不同的界面和交互。使用 PhoneGap 就意味着你的程序在 UI 和交互上，既不象原生的 iphone 程序，又不象原生的 android 程序。

3. 最后，性能问题。Javascript 终究无法和原生的程序比运行效率，这一点在当你要做一些动画效果的时候，就能显现得很明显。

当然，PhoneGap 的优势也很明显，如果你是做图书类，查询类，小工具类应用的话，这些应用 UI 交互不复杂，也不占用很高的 cpu 资源，PhoneGap 将很好地发挥出它的优势。对于这类应用：

1. 你只需要编写一次，则可以同时完成 iOS, android, windows phone 等版本的开发。
2. 如果改动不大，只是内容升级，那它升级时只需要更新相应的 js 文件，而不需要提交审核，而一般正常提交苹果的 app store 审核的话，常常需要一周时间。

所以 PhoneGap 不是万能的，但也不是没有用，它有它擅长的领域，一切都看你是否合理地使用它。

最后，推荐 PhoneGap 中国网站 ，在这里，你可以找到为数不多的中文资料。

对 js 的感想

现在前端工程师相当牛逼啊。前端工程师不但可以写前端网页，还可以用 Flex 写桌面端程序，可以用 nodejs 写 server 端程序，可以用 PhoneGap 写移动端程序，这一切，都是基于 javascript 语言的，还有最新出的 windows 8，原生支持用 js 来写 Metro 程序，世界已经无法阻止前端工程师了。

iOS

上一篇：
关于iOS测试机个数上限的详细规则

下一篇：
iPhone开发常问的十个问题(2012年3月版)

分类

- books summary
- iOS
- iOS weekly
- mac
- shell
- summary

微信公众号

关注我的微信公众号，和我一起成长：