

webview的使用方式

webview的常用类

WebSettings

作用：对webview进行配置和管理

具体用法

webview是否支持要访问的页面中的JavaScript代码运行。

setJavaScriptEnabled(boolean)

webview是否支持缩放操作

setSupportZoom(bloean);：是否支持缩放，默认为true。也就是说webview默认是支持缩放功能的

setBuiltInZoomControls(bloean);：设置内置的缩放控件。若为false，则该WebView不可缩放

setDisplayZoomControls(bloean);：是否隐藏原生的缩放控件

webview对网页的缓存策略

我们通过setCacheMode(int mode)方法去设置webview的缓存策略

WebSettings.LOAD_CACHE_ONLY：不使用网络，只读取本地缓存数据

WebSettings.LOAD_CACHE_ELSE_NETWORK：只要本地有缓存，无论是否过期，或者no-cache，都使用缓存中的数据。

WebSettings.LOAD_DEFAULT：根据cache-control决定是否从网络上取数据。

WebSettings.LOAD_NO_CACHE：不使用缓存，只从网络获取数据。

WebViewClient

作用：处 webview加载网页时的各种回调通知

当我们使用webview去打开网页的时候，默认使用android中自带的浏览器去打开，当我们在给webview添加了webviewClient之后，表示我们希望使用我们自己的webview去处理网页

具体的回调方法

WebResourceResponse shouldInterceptRequest(WebView view, String url)：当网页进行资源请求的时候进行回调，我们可以通过这个方法拦截对当前资源的请求。默认返回为null，当返回为null的时候表示webview会对这次资源请求进行处理，当返回为非空的时候，表示我们自己会对这次资源请求进行处理。

void onPageStarted(WebView view, String url, Bitmap favicon)：当网页开始加载之前回调

void onLoadResource(WebView view, String url)：加载网页资源之前回调

void onPageFinished(WebView view, String url)：当网页加载结束的时候回调

boolean shouldOverrideUrlLoading(WebView view, String url)：当网页进行URL请求的时候回调，return true表示我们已经对这次重定向进行了处理了，而return false 表示我们需要webview对这次重定向进行处理

void onReceivedError(WebView view, int errorCode, String description, String failingUrl)：当访问发生错误的时候回调。

WebChromeClient

作用：辅助webview去处JavaScript对话框、图标、标题和进度。

具体用法

void onProgressChanged(WebView view, int newProgress)：网页从开始加载到完成加载，网页加载进度的回调，其中的newProgress就是网页加载的进度，从0开始到100

void onReceivedTitle(WebView view, String title)：在这个方法中我们可以获取到网页的标题

boolean onJsAlert / onJsConfirm / onJsPrompt (***)：当网页想要去打开一个对话框的时候回调。默认为false， return false表示此事件需要webview去处理，Return true表示我们会自己处理这个事件，并且我们需要去告诉网页端我们的处理结果

Android调用JS代码

loadUrl()：通过loadUrl（‘javascript:方法名()’）这种方式来去调用js中的方法。但是这种方法会使页面进行刷新。并且如果我们想要回去js的返回值的话，非常的麻烦，而且一旦js中的方法return 了一个String类型的返回值，那么return的返回值会替换掉整个body标签中的内容

evaluateJavascript()：这个方法接受两个参数，第一个参数为调用的方法名：‘javascript:方法名()’，第二个参数为一个回调方法，他可以接受我们调用的Js方法的返回值。但是需要在android4.4以上才可以使用。

通过webview与JS交互

JS调用Android代码

通过拦截JavaScript请求的回调方法：这种方式没有办法直接给Js一个返回值，如果要给Js返回值的话，我们就需要通过android调用Js代码的方式把返回值传递给JS中的方法。

监听webViewClient的shouldOverrideUrlLoading(WebView view, String url)方法或者webChromeClient的onJsAlert()方法。或者一切可以拦截JavaScript请求的回调方法

根据接受到的参数去判断我们需要在android中执行的逻辑

通过对象映射的方式：这种通过映射的调用方式，非常方便，并且可以直接获得返回值，但是在android4.2之下，这种映射类的方式存在漏洞的，别人可以直接通过反射来直接进行恶意操作，在android4.2之后，引入了@JavascriptInterface 注解，使Js只能调用被@JavascriptInterface 注解的方法。

通过WebView的addJavascriptInterface（object，name）进行对象映射

通过 @JavascriptInterface注解在映射类中声明需要被Js调用的方法

在JS中通过 name.方法名()的方式来调用android的映射类中被@JavascriptInterface注解的方法

webview的常用方法

webview加载网页的四种方式

loadUrl(String url) 直接加载一个网页进来，这个网页可以是网络上的地址，也可以是手机本地的html页面，也可以是项目资源文件下的html页面。

loadUrl(String url, Map<String, String> additionalHttpHeaders) 直接加载一个网页进来，但是它提供了一个可以让我们添加requestHeader的参数，我们在这里加入的集合，会被添加到网页的requestHeader中

loadData(String data, String mimeType, String encoding) 直接加载一段html文本，我们给他指定需要加载的html文本的内容、mime类型，和编码格式。注意：‘#’, ‘%’, ‘\’, ‘?’ 这四种符号需要使用 %23, %25, %27, %3f encode编码来替代。

loadDataWithBaseURL(String baseUrl, String data, String mimeType, String encoding, String historyUrl) 基于一个基本的url，来加载一段html内容，并且我们可以给他指定一个浏览历史的地址，当我们从当前页面跳转出去之后，可以后退到这个浏览历史的地址中。

webview控制网页的前进和后退

canGoBack：当前页面是否可以后退

goBack：后退网页

canGoForward：当前的页面是否可以前进

goForward：前进页面

canGoBackOrForward(int steps)：以当前的index为起始点判断当前页面是否可以前进或者后退到历史记录中指定的steps，如果steps为负数则为后退，正数则为前进

goBackOrForward(int steps)：以当前的index为起始点前进或者后退到历史记录中指定的steps，如果steps为负数则为后退，正数则为前进

clearHistory()：我们可以通过 clearHistory 来去清理 webview的浏览历史

webview的状态管理

onResume()：激活WebView为活跃状态，能正常执行网页的响应

onPause()：当页面被失去焦点被切换到后台不可见状态，需要执行onPause，通过onPause动作通知内核暂停所有的动作，比如DOM的解析、plugin的执行、JavaScript执行。

pauseTimers()：当应用程序(存在webview)被切换到后台时，这个方法不仅仅针对当前的webview而是全局的全应用程序的webview，它会暂停所有webview的layout， parsing， javascript。降低CPU功耗。

resumeTimers()：恢复 pauseTimers状态

destroy()：销毁Webview ,在关闭了Activity时，如果不调用webview的destroy()方法，那么在你activity关闭时，webview本身并不会被销毁，就会出现内存溢出的问题。