Android-WebView

课程简介

什么是WebView

- WebView是Android中Ui组件的一种
- WebView基于webkit内核（Chromium）

课程简介

WebView有什么用呢？

- WebView可以用来展示网页，并且与网页进行交互

慕课网 imooc.com

课程简介

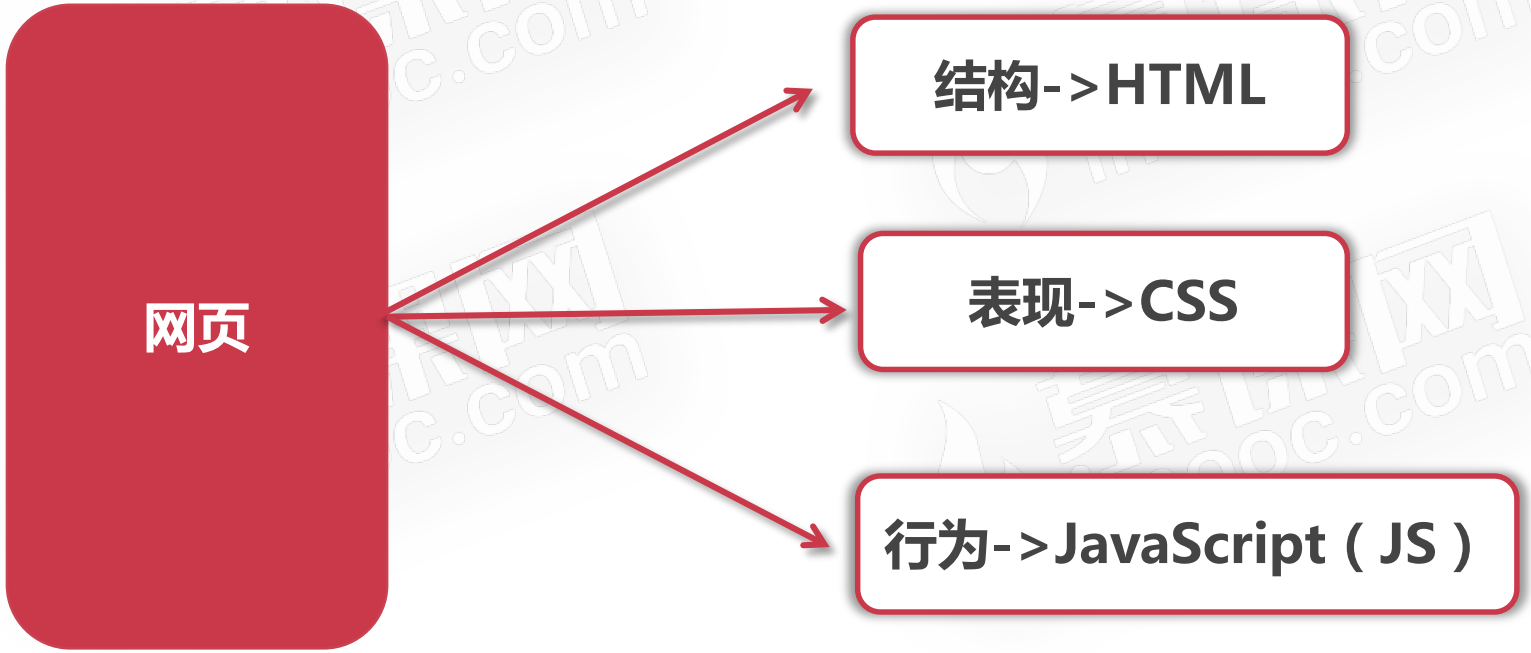
本课我们将要学到的内容

- WebView的常用方法
- WebView的常用类
- WebView如何与网页进行交互

网页的基本组成

慕课网 imooc.com

网页的基本组成



网页的基本组成

```
<a style="color: red">alert</a>

<script>
    console.log('hello js');
</script>
```

HTML标签 -> <a>

HTML信息 -> alert

css -> color:red

js-> console.log(‘hello js’)

搭建本地的http-server

- 安装node环境：https://nodejs.org/zh-cn/
- 安装CNPM：npm install -g cnpm --registry=https://registry.npm.taobao.org
- 安装http-server：cnpm install http-server -g

WebView的常用方法

加载网页的四种方式

- loadUrl(String url)
- loadUrl(String url, Map<String, String> additionalHttpHeaders)
- loadData(String data, String mimeType, String encoding)
- loadDataWithBaseURL(String baseUrl, String data, String mimeType, String encoding, String historyUrl)

控制网页的前进和后退

- boolean canGoBack()
- boolean canGoForward()
- boolean canGoBackOrForward(int steps)
- void clearHistory()
- void goBack()
- void goForward()
- void goBackOrForward(int steps)

WebView的状态管理

- onPause ()
- onResume()
- pauseTimers ()
- resumeTimers()
- destroy()

WebView的常用类

- WebSettings：对webview进行配置和管理
- WebViewClient：处理webview加载时的各种回调通知
- WebChromeClient：辅助webview去处理JavaScript对话框、标题进度等。

WebSettings

对webview进行配置和管理

- webview是否支持要访问的页面中的JavaScript代码运行
- webview是否支持缩放操作
- webview对网页的缓存策略

控制网页的缩放

- setSupportZoom(boolean)：是否支持缩放
- setBuiltInZoomControls(boolean)：设置内置的缩放控件
- setDisplayZoomControls(boolean)：是否隐藏原生的缩放控件

控制webview的缓存策略

- LOAD_CACHE_ONLY：永远不使用网络，只去本地缓存，没有缓存则不会加载
- LOAD_CACHE_ELSE_NETWORK：只要本地有缓存，无论是否过期都会去使用本地缓存,没有缓存才会去加载网络
- LOAD_DEFAULT：（默认）根据cache-control决定是否从网络获取
- LOAD_NO_CACHE：永远不使用缓存，只从网络获取

WebViewClient

处理网页加载时的各种回调通知

- WebResourceResponse shouldInterceptRequest(WebView view, String url)：进行资源请求的时候回调
- void onPageStarted(WebView view, String url, Bitmap favicon)：网页已经开始加载的时候回调
- void onLoadResource(WebView view, String url)：加载网页资源之前回调
- void onPageFinished(WebView view, String url)：网页加载完成的时候回调

处理网页加载时的各种回调通知

- boolean shouldOverrideUrlLoading(WebView view, String url)：webview将要加载新的url时进行回调

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

WebChromeClient

用回调方法处理网页中对话框、标题、进度等

- void onProgressChanged(WebView view, int newProgress)：获取网页加载进度

- void onReceivedTitle(WebView view, String title)：获取网页标题

用回调方法处理网页中对话框、标题、进度等

- boolean onJsAlert(WebView view, String url, String message, JsResult result)：在网页将要打开一个alert警告对话框的时候回调
- boolean onJsConfirm(WebView view, String url, String message, JsResult result)：在网页将要打开一个confirm对话框的时候回调
- boolean onJsPrompt(WebView view, String url, String message, String defaultValue, JsPromptResult result)：在网页将要打开一个prompt对话框的时候回调

通过WebView与Js交互

Android与JS的交互

Android去调用js代码

- loadUrl(‘javascript:方法名(’参数’…)’)
- evaluateJavascript(‘javascript:方法名(’参数’…), ValueCallback<String> resultCallback)

Android与JS的交互

JS去调用Android代码

- 通过拦截JavaScript请求的回调方法
- 对象映射

Android与JS的交互

拦截JavaScript请求的回调方法

- 监听webViewClient的shouldOverrideUrlLoading(WebView view, String url)方法
- 根据接受到的url参数去判断，在android中需要执行的内容

Android与JS的交互

通过对象映射

- 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

Android与JS的交互

Android去调用js代码

- loadUrl(‘javascript:方法名()’)：无法方便的获取JS中的返回值
- evaluateJavascript(String script, ValueCallback<String> resultCallback)：需要android4.4以上的版本

Android与JS的交互

JS去调用Android代码

- 通过WebView的addJavascriptInterface（object, name）进行对象映射：在android4.2以下存在安全漏洞
- 通过拦截JavaScript请求的回调方法：JS中无法方便的获取android中的返回值，使用繁琐

Android-WebView

我们学到的内容

- WebView的常用方法
- WebView的常用类
- WebView如何与网页进行交互