

Evan

Only let oneself become strong enough, good enough, can afford the life that you want to.

目录视图

摘要视图

分享

从创业到再就业，浅述对程序员职业生涯的看法 征文 | 你会为 AI 转型么？ 赠书：7月大咖新书机器学习/Android/python

PhoneGap之自定义插件

标签： android phonegap 自定义插件

2016-05-12 23:55 322人阅读 评论(0) 收藏 举报

分类： 移动开发 (38) PhoneGap (1)

版权声明：本文为博主原创文章，未经博主允许不得转载。

PhoneGap：作为原生App，Java（这里是指Android的）与JavaScript的通信桥梁，使得我们的混合开发更加得心应手，我是与android结合的混合开发。

但在这里不得不吐槽一下PhoneGap这个坑：更新速度很快，但是学习资料的更新却远远跟不上节奏。而且必须按照它的套路走，而且就算按着套路走也可能会出错，

最后还是得靠自己摸索。。。。

好了，不说那么多了，下面介绍PhoneGap自定义插件的使用（也可以说是CordovaPlugin），主要分以下步骤：

1) 下载最新版的cordova.jar，我在使用的是cordova3.5.0.jar 版本的，把下载的 cordova-x.x.x.jar导入到工程的libs目录下来。

说明：这个cordova.jar 是用于Android原生Java语言这边使用的，有了这个架包就可以在Android这边编写自定义的插件。

2) 把cordova.js，cordova_plugin.js 这两个javascript文件复制到www文件目录下。

说明：这两个js文件是用于JavaScript调用插件中方法使用。

3) 在www文件目录下新建一个 plugins 文件夹目录。

说明：用于存放自定义插件的JavaScript文件

4) 编写自定义插件中的 java 代码部分：FirstPlugin.java

```
[java]
01. <span style="font-size:14px;">package com.customcordovaplugin.plugin;
02.
03. import org.apache.cordova.CallbackContext;
04. import org.apache.cordova.CordovaPlugin;
05. import org.json.JSONArray;
06. import org.json.JSONException;
07.
08. /**
09.  * Created by long on 2016/5/10.
10.  */
11. public class FirstPlugin extends CordovaPlugin {
12.     @Override
13.     public boolean execute(String action, JSONArray args, CallbackContext callbackContext)
14.         throws JSONException {
15.         //检查方法名，例如我们定义一个名为sayHello的方法
16.         if("sayHello".equals(action)){
```

```

17.      /**
18.       * 关于回调函数
19.       * 成功就执行 callbackContext.success(param)，从而触发成功回调函数
20.       * 失败就执行 callbackContext.error(param)，从而触发失败回调函数
21.       * 如果不需要回调函数，那么就没必要调用这两个方法
22.       */
23.
24.       //获取第一个参数，如果有多个参数的话可以使用索引号分别获取
25.       String message = args.getString(0);
26.       //检查输入值是否为空，如果为空值就会调用 callbackContext.error()方法
27.       //这会导致失败回调函数被调用
28.       if(message!=null && message.length()>0){
29.           callbackContext.success("Hello: "+message+"!");
30.       }else{
31.           callbackContext.error("输入值为空！");
32.       }
33.       //最后还需要返回逻辑值 true 来表明方法存在
34.       return true;
35.     }
36.     //否则，如果方法名不符，那么就返回false
37.     //在网页的 JavaScript 程序中调用时就会导致“MethodNotFound”错误
38.     return false;
39. }
40. }
41. </span>

```

5) 编写自定义插件中的 JavaScript 部分：firstplugin.js

```

[javascript]
01. <span style="font-size:14px;">cordova.define("firstplugin_id",
02.     function(require, exports, module) {
03.         //引入必要的类
04.         var exec = require('cordova/exec');
05.
06.         //创建一个JavaScript类
07.         function FirstPlugin(){};
08.
09.         //为其添加一个方法，在方法中调用exec()方法
10.         FirstPlugin.prototype.sayHello = function(name,successCallback,failureCallback){
11.             exec(successCallback, failureCallback,
12.                 'FirstPlugin', 'sayHello', [name]);
13.         };
14.
15.         //公共的对象是 FirstPlugin 实例
16.         module.exports = new FirstPlugin();
17.     }
18. );</span>

```

6) 编写程序入口html文件：index.html

```

[html]
01. <span style="font-size:14px;"><!DOCTYPE html>
02. <html>
03.     <head>
04.         <meta charset="utf-8" />
05.         <meta name="format-detection" content="telephone=no" />
06.         <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-
width, height=device-height, target-densitydpi=device-dpi" />
07.         <script type="text/javascript" src="cordova.js"></script>
08.         <script type="text/javascript" src="plugins/firstplugin.js"></script>
09.         <title>Simple</title>
10.     </head>
11.     <body onload="onLoaded()">
12.         <label for="inputName">请输入名字: </label>
13.         <input type="text" id="inputName">
14.         <input type="button" onclick="clickHandler()" value="sayHello" />
15.         <script>
16.             //当DOM加载完毕，注册deviceready事件
17.             function onLoaded(){
18.                 //注册deviceready事件监听，指定deviceready事件处理函数

```

```

19.         document.addEventListener("deviceready",onDeviceReady,false);
20.     }
21.     //成功回调函数
22.     function onSuccess(data){
23.         alert(data);
24.     }
25.     //失败回调函数
26.     function onError(err){
27.         alert(err);
28.     }
29.
30.     //deviceready 事件处理函数，PhoneGap 完成加载后调用该函数
31.     function onDeviceReady(){
32.         //启动所有input元素
33.         var nodelist = document.getElementsByTagName('input');
34.         var nodeLen = nodelist.length;
35.         for(var i=0;i<nodeLen;i++){
36.             nodelist[i].disabled = false;
37.         }
38.     }
39.     //按钮单击事件处理函数
40.     function clickHandler(){
41.         //获取输入文本框中的字符，并将其作为参数传递给sayHello()方法
42.         inputName = document.getElementById("inputName");
43.         var dialog = com.mine.FirstPlugin;
44.         dialog.sayHello(inputName.value, onSuccess, onError);
45.     }
46.     </script>
47. </body>
48. </html></span>

```

说明：

[html]

```

01. <span style="font-size:14px;">                var dialog = com.mine.FirstPlugin;    <span style="font-
family: Arial, Helvetica, sans-serif;">是在后面讲到的cordova_plugins.js文件中配置的</span>
02.
03.         dialog.sayHello(inputName.value, onSuccess, onError);</span>

```

[html]

```

01. <span style="font-size:14px;">调用的这个sayHello方法就是调用了 firstplugin.js 文件中的这个方法: </span>

```

[html]

```

01. <span style="font-size:14px;"><pre name="code" class="javascript">                //为其添加一个方法，在方法中调用exec()方法
02.     FirstPlugin.prototype.sayHello = function(name,successCallback,failureCallback){
03.         exec(successCallback, failureCallback,
04.             'FirstPlugin', 'sayHello', [name]);
05.     };</span>

```

7) 接下来就是配置config.xml文件，在res下面新建xml文件夹，并且将config.xml 放到该文件夹下面，config.xml 代码如下：

[html]

```

01. <span style="font-size:14px;"><widget xmlns      = "http://www.w3.org/ns/widgets"
02.     id          = "io.cordova.helloCordova"
03.     version     = "2.0.0">
04.
05.     <feature name="FirstPlugin">
06.         <param name="android-package" value="com.customcordovaplugin.plugin.FirstPlugin"/>
07.     </feature>
08.
09. </widget></span>

```

说明：value的值就是 FirstPlugin.java 文件的路径及名称

8) 配置cordova_plugin.js 文件，cordova_plugins.js代码如下：

[javascript]

```

01. <span style="font-size:14px;">cordova.define('cordova/plugin_list',function(require, exports, module){

```

```
02.     module.exports = [{
03.         "file": "plugins/firstplugin.js",           //表示注册和连接JavaScript文件的位置
04.         "id": "firstplugin_id",                     //表示唯一的标示
05.         "clobbers": ["com.mine.FirstPlugin"]        //表示可以引用的接口(用于html中调用JavaScript的方法（自定义）)
06.     ]
07.
08.     module.exports.metadata = {
09.         "firstplugin_id": "1.0",                     //插件的版本号
10.     }
11. });
12. </span>
```

到这里，PhoneGap自定义插件的工作完成了，PhoneGap的学习到处是坑，大家学习的时候细心点。

相关文章分享：

- <http://blog.csdn.NET/yuhailong626/article/details/30059679>
- <http://blog.csdn.net/kong92917/article/details/44086821>
- <http://blog.csdn.Net/xiaoluo5238/article/details/38711331>
- <http://blog.csdn.net/Maxbalance/article/details/50483490>
- <http://blog.csdn.net/happyflyingave/article/details/25182369>

顶 踩
0 0

- 上一篇 Android 实现对图片 Exif 的修改 (Android 自带的方法)
- 下一篇 struts.xml配置详解

相关文章推荐

- phonegap定义插件在xcode中的命令行安装和卸载
- 自定义jquery插件简单介绍
- android phonegap自定义插件 发送短信
- android PhoneGap 自定义插件
- Phonegap中自定义插件的使用
- 自定义 动态显示图片留白比例插件
- cordova (phonegap) 4.2自定义插件
- Eclipse中使用Checkstyle,checkstyle插件检查jav...
- 自定义 JS 水平滚动字体插件
- cordova 自定义插件



猜你在找

- 机器学习之概率与统计推断
- 机器学习之凸优化
- 响应式布局全新探索
- 深度学习基础与TensorFlow实践
- 前端开发在线峰会
- 机器学习之数学基础
- 机器学习之矩阵
- 探究Linux的总线、设备、驱动模型
- 深度学习之神经网络原理与实战技巧
- TensorFlow实战进阶：手把手教你做图像识别应用

查看评论