

# 知识动力 Phonegap 资料

## 目录

Phonegap 配置与安装.....	1
Phonegap 照相插件的使用.....	6
Phonegap Notification 震动 响铃 调用原生弹窗功能.....	13
Phonegap 插件开发方法 plugin.....	18
Phonegap 跳转 Activity android 插件调用原生.....	30
Phonegap/cordova 升级版本.....	40
Phonegap 之 config.xml 标签.....	41
Phonegap 的 deviceready 事件.....	43
Phonegap 网络连接状态 online/offline html5.....	46
Phonegap 事件之菜单键 html5 菜单事件.....	47
Phonegap 事件之 resume 事件.....	49
Phonegap 事件之 pause 事件.....	53
Phonegap 返回键 backbutton 功能 cordova.....	55
Phonegap 获取 imei (修改 phonegap 官方提供文件).....	58
Phonegap 联系人 获取 Contacts.....	85
methods.....	86
Arguments.....	86
Object.....	86
<1> 在控制台 创建一个 phonegap 工程 命令如下.....	86
<2> 我们从命令行进入 到工程目录下的 plugins 文件夹.....	86
<3> 现在开始下载插件.....	87
<4> 添加 android 平台工程 (ios 把 "android" 替换).....	87
<5> 编译 android 工程.....	87
example:.....	90
代码详解:.....	93
Phonegap 是否连接网络状态 online/offline.....	94
Phonegap 网络连接 HTML5.....	97
<1> 在控制台 创建一个 phonegap 工程 命令如下.....	97
Phonegap Device 获取设备信息.....	104
Phonegap 修改全屏/非全屏设置.....	111
Phonegap 解决错误:Error initializing Cordova:Class not found.....	113
Phonegap 中文乱码解决方法.....	115
phonegap 导入工程报错解决办法 错误解析 android.....	118
附件.....	119

## Phonegap 配置与安装

本文主讲 windows 系统下的 phonegap 的环境搭建 ,以 android 视角为主.ios 用户可参考;

首先介绍搭建环境的基本准备:

1 java JDK (搜索一下.....)

2 android SDK 最新版(当前的版本是 API 19

<http://developer.android.com/sdk/index.html>)

3 Ant 打包工具 (<http://ant.apache.org/bindownload.cgi>)

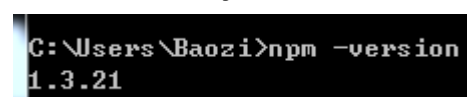
4 eclipse (建议使用 google 的 因为内置 ADT [点击打开链接](#) )

5 ADT (eclipse 里的插件,安卓开发工具插件 下载上面的链接,内部集成了 ADT)

6 Node.js (<http://nodejs.org/>)

7 sublime

下图是 Node.js 安装成功后



```
C:\Users\Baozi>npm -version
1.3.21
```

将以上配置好环境变量 (配置方法略 [点击](#) )

特别提一下需要配环境变量的东西: java JDK ; android SDK;Ant ;

贴上我的 path 供大家参考:

(建议配成 ;%ANT\_HOME%\bin; 去掉\bin)

F:\工具\Ant\ant-1.8\bin;

C:\Develop\jdk\jdk\platform-tools;C:\Develop\jdk\jdk\tools;%Android%\.;C:\Develop\nodejs;

%JAVA\_HOME%\bin;C:\Develop\nodejs\

拥有以上工具后

接下来开始 phonegap 的安装了

phonegap 官网 <http://phonegap.com/>在控制台运行命令:

```
npm install -g phonegap ( Mac 使用: sudo npm install -g phonegap )
```

//等待安装 完成后安装 cordova:

```
npm install -g cordova ( Mac 使用: sudo npm install -g cordova )
```

安装完成后

```
npm install -g phonegapnpm install -g phonegapnpm install -g phonegapnpm install -g
```

phonegap 检验语句:

```
phonegap -version
```

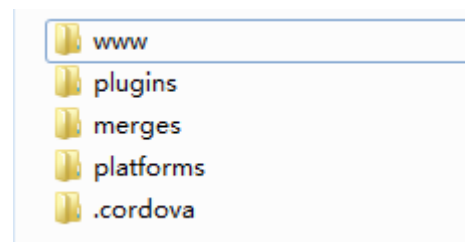
```
cordova -version
```

在命令行 使用:

1. phonegap create my-app
2. cd my-app
3. phonegap run android

第一个 phonegap 写的程序就出现了

会在 my-app 目录下出现 5 个文件夹



www 文件夹 开发的 HTML5 ; CSS ; JS 文件都拷贝到这下面

plugins 文件夹 存放的是 phonegap 插件 以后 文件,摄像头等插件都下载到这里,在下篇文章中介绍用法

merges 我也没用过 以后知道了再补上

platforms 文件夹 存放的是编译好后的 android 文件 ,(如果这个文件夹为空,需要你在命令行编译一次才能生成. 如上面 phonegap run android)

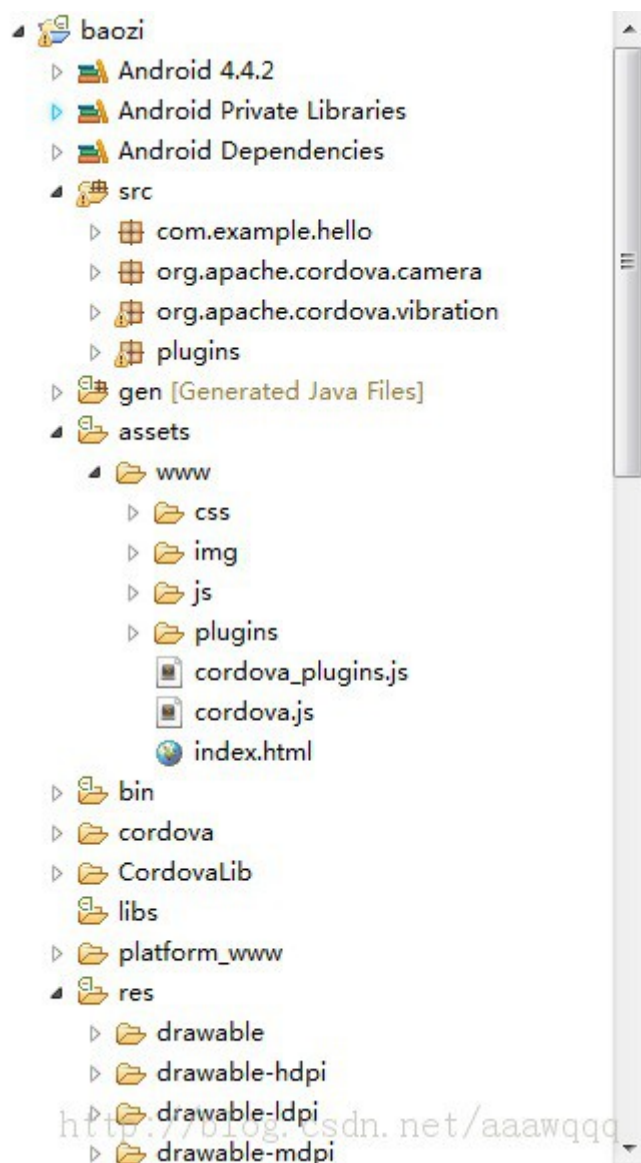
.cordova 存放的是配置文件

将此文件夹导入 eclipse 就可以使用

phonegap 将以第三方库的形式导入

如果 src 文件报错 检查是否是 phonegap 的功能没有导入进来

导入 eclipse 后 文本结构....下图打的 camera 和 vibration 是 相机与铃声的插件 下章节介绍



(图 1)

下图是 phonegap 的启动页面 调用 `super.loadUrl();` 方法

```
1. public class baozi extends CordovaActivity
2. {
3.     @Override
4.     public void onCreate(Bundle savedInstanceState)
5.     {
6.         super.onCreate(savedInstanceState);
7.         super.init();
8.         // Set by <content src="index.html" /> in config.xml
9.         // super.loadUrl(Config.getStartUrl());
10.        super.loadUrl("file:///android_asset/www/index.html");
11.
12.
13.    }
14. }
```

在 `super.loadUrl(" ");` 中填写你的启动页面

参数: "`file:///android_asset/www/index.html`" 指的是(图 1)的 `assets` 目录下 `www` 文件夹下的 `index.html`



## Phonegap 照相插件的使用

phonegap 的 API 提供照相功能, 使用很方便.

此篇文章以 cordova 3.3 版本编写

据我总结核心步骤: 创建工程 ; 下载插件 ; 编译工程; 调用插件;

按照如下步骤就能生产出代码:

打开 cmd 控制台

1 使用命令行 建立 phonegap 工程

2 进入到 phonegap 工程下的 plugins 文件夹当中 , 使用命令行下载官方插件 ; (输入命令:  
cordova plugin add org.apache.cordova.camera)

4 添加生成 android 或者 IOS 工程 ; android 命令如下 cordova platform add android  
(注:oc 的把 android 换成 ios)

5 在命令行输入 cordova build

6 最后 将生成的工程导入编辑器(我用的 eclipse)

在 assets/www 目录下在 index.html 中添加 javascript 调用语句

接下来给大家演示:

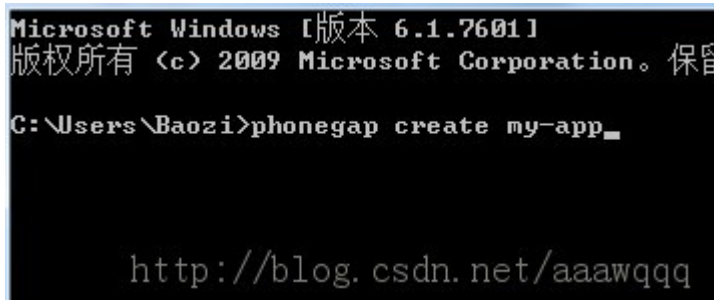
目测大家都是已经安装好环境的, 如果没有搭好环境 可以查看我的 phonegap 配置文章 [点击打开链接](#)

调用系统的 API 官方文档地址 :

[http://docs.phonegap.com/en/3.3.0/cordova\\_camera\\_camera.md.html#Camera](http://docs.phonegap.com/en/3.3.0/cordova_camera_camera.md.html#Camera)

<1> 在控制台 创建一个 phonegap 工程 命令如下

```
1. phonegap create my-app
2. cd my-app
3. phonegap run android
```



Microsoft Windows [版本 6.1.7601]  
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。  
C:\Users\Baozi>phonegap create my-app\_  
  
<http://blog.csdn.net/aaawqqq>

<2> 我们从命令行进入 到工程目录下的 plugins 文件夹

1. cd my-app
- 2.
3. cd plugins

```
C:\Develop\phonegap>phonegap create my-app
[phonegap] created project at C:\Develop\phonegap\my-app

C:\Develop\phonegap>cd my-app

C:\Develop\phonegap\my-app>cd plugins

C:\Develop\phonegap\my-app\plugins>

http://blog.csdn.net/aaawqqq
```

<3> 现在开始下载插件 本篇讲述 camera :

输入:

1. cordova plugin add org.apache.cordova.camera

```
C:\Develop\phonegap\my-app\plugins>cordova plugin add org.apache.cordova.camera
Fetching plugin "org.apache.cordova.camera" via plugin registry

C:\Develop\phonegap\my-app\plugins>_

http://blog.csdn.net/aaawqqq
```

插件下载完毕

<4> 添加 android 平台工程 (ios 把 "android" 替换)

1. cordova platform add android

```
C:\Develop\phonegap\my-app\plugins>cordova platform add android
Creating android project...
Preparing android project
Starting installation of "org.apache.cordova.camera" for android
Preparing android project
org.apache.cordova.camera installed on android.

C:\Develop\phonegap\my-app\plugins>_

http://blog.csdn.net/aaawqqq
```

<5> 编译 android 工程

1. cordova build



```
C:\Develop\phonegap\my-app\plugins>cordova build
Generating config.xml from defaults for platform "android"
Preparing android project
Compiling app on platform "android" via command "cmd" /c C:\Develop\phonegap\my-app\platforms\android\cordova\build
Platform "android" compiled successfully.
C:\Develop\phonegap\my-app\plugins>_
```

<http://blog.csdn.net/aaawqqq>

至此 camera 已经生成...

现在大家只需要将工程导入到 eclipse 当中 使用官方 API 语句在 javascript 中调用

将下方语句考到 assets 目录下 www/index.html 当中 完全复制过去;

example:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Capture Photo</title>
5. <script type="text/javascript" charset="utf-8"
src="cordova.js"></script>
6. <script type="text/javascript" charset="utf-8">
7. var pictureSource; // picture source
8. var destinationType; // sets the format of returned value
9. // Wait for device API libraries to load //
10. document.addEventListener("deviceready",onDeviceReady,false);
11. // device APIs are available //
12. function onDeviceReady() {
pictureSource=navigator.camera.PictureSourceType;
destinationType=navigator.camera.DestinationType; }
13. // Called when a photo is successfully retrieved //
14. function onPhotoDataSuccess(imageData) {
15. // Uncomment to view the base64-encoded image data //
16. console.log(imageData);
```

```
17. // Get image handle //
18. var smallImage = document.getElementById('smallImage');
19. // Unhide image elements //
20. smallImage.style.display = 'block';
21. // Show the captured photo // The in-line CSS rules are used to
    resize the image //
22. smallImage.src = "data:image/jpeg;base64," + imageData; }
23. // Called when a photo is successfully retrieved //
24. function onPhotoURISuccess(imageURI) {
25. // Uncomment to view the image file URI //
26. console.log(imageURI);
27. // Get image handle //
28. var largeImage = document.getElementById('largeImage');
29. // Unhide image elements //
30. largeImage.style.display = 'block';
31. // Show the captured photo // The in-line CSS rules are used to
    resize the image //
32. largeImage.src = imageURI; }
33. // A button will call this function //
34. function capturePhoto() {
35. // Take picture using device camera and retrieve image as base64-
    encoded string
36. navigator.camera.getPicture(onPhotoDataSuccess, onFail, { quality:
    50, destinationType: destinationType.DATA_URL }); }
37. // A button will call this function //
38. function capturePhotoEdit() {
39. // Take picture using device camera, allow edit, and retrieve
    image as base64-encoded string
```

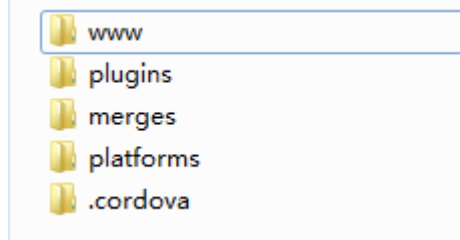
```
40.     navigator.camera.getPicture(onPhotoDataSuccess, onFail, { quality:
20, allowEdit: true, destinationType: destinationType.DATA_URL }); }
41.     // A button will call this function //
42.     function getPhoto(source) {
43.         // Retrieve image file location from specified source
44.         navigator.camera.getPicture(onPhotoURISuccess, onFail, { quality:
50, destinationType: destinationType.FILE_URI, sourceType: source }); }
45.     // Called if something bad happens. //
46.     function onFail(message) { alert('Failed because: ' + message); }
47.     </script>
48.     </head>
49.     <body>
50.         <button onclick="capturePhoto();">Capture Photo</button> <br>
51.         <button onclick="capturePhotoEdit();">Capture Editable
Photo</button> <br>
52.         <button onclick="getPhoto(pictureSource.PHOTOLIBRARY);">From Photo
Library</button><br>
53.         <button onclick="getPhoto(pictureSource.SAVEDPHOTOALBUM);">From
Photo Album</button><br>
54.         <img style="display:none;width:60px;height:60px;" id="smallImage"
src="" />
55.         <img style="display:none;" id="largeImage" src="" />
56.
57.     </body>
58. </html>
```

相机参数详解:

调用系统的 API 官方文档地址

[http://docs.phonegap.com/en/3.3.0/cordova\\_camera\\_camera.md.html#Camera](http://docs.phonegap.com/en/3.3.0/cordova_camera_camera.md.html#Camera)

phonegap 的工程:



工程下载 将 phonegap 的 platforms 导入到 eclipse 中

如果报错 clear 一下 查看导的 lib 包 有没有报错

Demo 下载:

<http://download.csdn.net/detail/aaawqqq/6974247>

## Phonegap Notification 震动 响铃 调用原生弹窗功能.

打开 cmd 控制台

1 使用命令行 建立 phonegap 工程

2 进入到 phonegap 工程下的 plugins 文件夹当中 ,使用命令行下载官方插件 ;(输入命令:  
cordova plugin add org.apache.cordova.vibration)

4 添加生成 android 或者 IOS 工程 ;android 命令如下 cordova platform add android  
(注:oc 的把 android 换成 ios)

5 在命令行输入 cordova build

6 最后将生成的工程导入编辑器(我用的 eclipse)

在 assets/www 目录下在 index.html 中添加 javascript 调用语句

<2> 我们从命令行进入 到工程目录下的 plugins 文件夹

```
cd my-app cd plugins
```

```
C:\Develop\phonegap>phonegap create my-app  
[phonegap] created project at C:\Develop\phonegap\my-app  
  
C:\Develop\phonegap>cd my-app  
  
C:\Develop\phonegap\my-app>cd plugins  
  
C:\Develop\phonegap\my-app\plugins>  
  
http://blog.csdn.net/aaawqqq
```

<3> 现在开始下载插件 本篇讲述 camera :

输入:

```
cordova plugin add org.apache.cordova.dialogs
```

```
cordova plugin add org.apache.cordova.vibration
```

```

C:\Develop\phonegap\my-app\plugins>cordova plugin add org.apache.cordova.dialogs
Fetching plugin "org.apache.cordova.dialogs" via plugin registry
Starting installation of "org.apache.cordova.dialogs" for android
Preparing android project
org.apache.cordova.dialogs installed on android.

C:\Develop\phonegap\my-app\plugins>cordova plugin add org.apache.cordova.vibration
Fetching plugin "org.apache.cordova.vibration" via plugin registry
Starting installation of "org.apache.cordova.vibration" for android
Preparing android project
org.apache.cordova.vibration installed on android.

C:\Develop\phonegap\my-app\plugins>

```

<http://blog.csdn.net/aaawqqq>

插件下载完毕

<4> 添加 android 平台工程 (ios 把 "android" 替换)

cordova platform add android

```

C:\Develop\phonegap\my-app\plugins>cordova platform add android
Creating android project...
Preparing android project
Starting installation of "org.apache.cordova.camera" for android
Preparing android project
org.apache.cordova.camera installed on android.

```

<http://blog.csdn.net/aaawqqq>

C:\Develop\phonegap\my-app\plugins>

<5>编译 android 工程

```

C:\Develop\phonegap\my-app\plugins>cordova build
Generating config.xml from defaults for platform "android"
Preparing android project
Compiling app on platform "android" via command "cmd" /c C:\Develop\phonegap\my-app\platforms\android\cordova\build
Platform "android" compiled successfully.

```

<http://blog.csdn.net/aaawqqq>

C:\Develop\phonegap\my-app\plugins>

cordova build

至此 Notification 已经生成...

检查下配置文件有没有生成 如果没有生成就再次 cordova.build 编译一次  
(in app/res/xml/config.xml)

```
<feature name="Notification">
<param name="android-package" value="org.apache.cordova.dialogs.Notification" />
</feature>
<feature name="Vibration">
<param name="android-package" value="org.apache.cordova.vibration.Vibration" />
</feature>
(in app/AndroidManifest.xml)
<uses-permission android:name="android.permission.VIBRATE" />
```

现在大家只需要将工程导入到 **eclipse** 当中 使用官方 **APi** 语句在 **javascript** 中调用

将下方语句考到 **assets** 目录下 **www/index.html** 当中 完全复制过去;

Full example:

```
<!DOCTYPE html>
<html>
<head>
<title>Capture Photo</title>
<script type="text/javascript" charset="utf-8" src="cordova.js"></script>
<script type="text/javascript" charset="utf-8">
    var pictureSource; // picture source
    var destinationType; // sets the format of returned value
    // Wait for device API libraries to load  //
    document.addEventListener("deviceready",onDeviceReady,false);
        // device APIs are available  //
        function onDeviceReady() {      pictureSource=navigator.camera.PictureSourceType;
destinationType=navigator.camera.DestinationType; }
        // Called when a photo is successfully retrieved  //
        function onPhotoDataSuccess(imageData) {
            // Uncomment to view the base64-encoded image data  //
            console.log(imageData);
            // Get image handle  //
            var smallImage = document.getElementById('smallImage');
            // Unhide image elements  //
            smallImage.style.display = 'block';
            // Show the captured photo  // The in-line CSS rules are used to resize the
image  //
            smallImage.src = "data:image/jpeg;base64," + imageData; }
        // Called when a photo is successfully retrieved  //
```

```

function onPhotoURISuccess(imageURI) {
// Uncomment to view the image file URI    //
console.log(imageURI);
// Get image handle    //
var largeImage = document.getElementById('largeImage');
// Unhide image elements    //
largeImage.style.display = 'block';
// Show the captured photo    // The in-line CSS rules are used to resize the
image //
largeImage.src = imageURI; }
// A button will call this function    //
function capturePhoto() {
// Take picture using device camera and retrieve image as base64-encoded string
navigator.camera.getPicture(onPhotoDataSuccess, onFail, { quality: 50,
destinationType: destinationType.DATA_URL }); }
// A button will call this function    //
function capturePhotoEdit() {
// Take picture using device camera, allow edit, and retrieve image as base64-encoded
string
navigator.camera.getPicture(onPhotoDataSuccess, onFail, { quality: 20, allowEdit: true,
destinationType: destinationType.DATA_URL }); }
// A button will call this function    //
function getPhoto(source) {
// Retrieve image file location from specified source
navigator.camera.getPicture(onPhotoURISuccess, onFail, { quality: 50,
destinationType: destinationType.FILE_URI,    sourceType: source }); }
// Called if something bad happens.    //
function onFail(message) {    alert('Failed because: ' + message); }
</script>
</head>
<body>
<button onclick="capturePhoto();">Capture Photo</button> <br>
<button onclick="capturePhotoEdit();">Capture Editable Photo</button> <br>
<button    onclick="getPhoto(pictureSource.PHOTOLIBRARY);">From    Photo
Library</button><br>
<button    onclick="getPhoto(pictureSource.SAVEDPHOTOALBUM);">From    Photo
Album</button><br>
<img style="display:none;width:60px;height:60px;" id="smallImage" src="" />
<img style="display:none;" id="largeImage" src="" />

</body>
</html>

```

3 个按钮

show alert 按钮 调用 Dialog

play Beep 按钮调用 声音提示



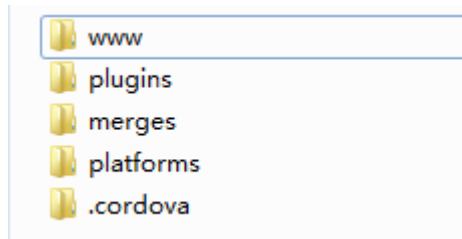
Vibrate 按钮调用 震动

参数详解:

调用系统的 API 官方文档地址

[http://docs.phonegap.com/en/3.3.0/cordova\\_notification\\_notification.md.html#Notification](http://docs.phonegap.com/en/3.3.0/cordova_notification_notification.md.html#Notification)

phonegap 的工程:



工程下载

将 phonegap 的 platforms 导入到 eclipse 中

如果报错 clear 一下 查看导的 lib 包 有没有报错

Demo 我亲自用真机测过 测试手机为 联想 S939 与三星 s3 9300

如果还是报错 请查看 是否是 因为没有用 android 最新的 SDK 我目前的版本的是 4.4.2 (API 19)

## ~~Phonegap 插件开发方法 plugin~~

据我总结核心步骤: 创建工程 ; 编写插件 ;编译工程; 调用插件;

按照如下步骤就能生产出代码:

打开 cmd 控制台

1 使用命令行 建立 phonegap 工程

2 将工程导入 eclipse

3 在 assents 目录下的 cordova-plugins.js 文件添加配置

4 在 plugin 目录下编写 javascript 接口

5 在 res/xml 目录下配置 config.xml 文件

6 在 src 目录下编写 java 文件

最后在 javascript 文件中调用接口

总体说 主要是后 4 个步骤详细分解讲解

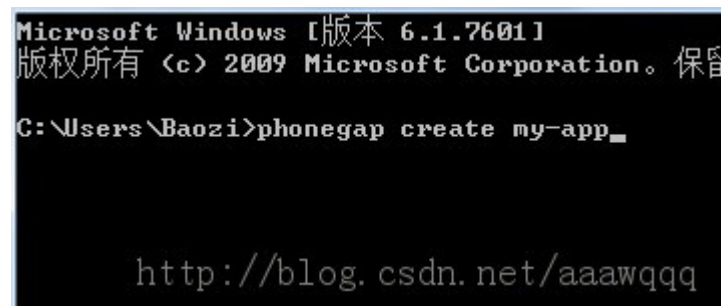
接下来给大家分解演示：

目测大家都是已经安装好环境的, 如果没有搭好环境 可以查看我的 phonegap 配置文章[点击查看链接](#)

调用系统的 API 官方文档地址 [点击查看链接](#)

<1> 在控制台 创建一个 phonegap 工程 命令如下

```
[java] view=plaincopyprint?C
1. phonegap create my-app
2. cd my-app
3. phonegap run android
```

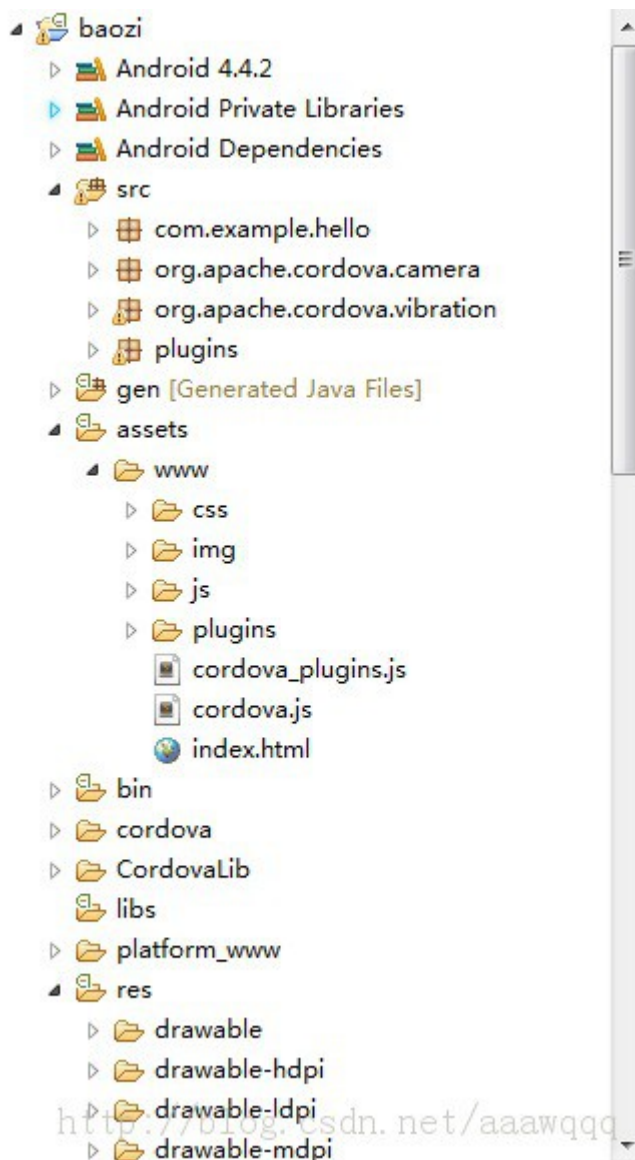


```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Baozi>phonegap create my-app_

http://blog.csdn.net/aaawqqq
```

<2> 将工程导入 eclipse



<3> 配置 cordova\_plugins.js 文件

首先给大家看看 cordova\_plugins.js 文件:

```
[javascript] view plaincopyprint?  
1. cordova.define('cordova/plugin_list', function(require, exports,  
module) {  
2. module.exports = [  
3. {  
4. "file": "plugins/org.apache.cordova.camera/www/CameraConstants.js",  
5. "id": "org.apache.cordova.camera.Camera",  
6. "clobbers": [  
7. "Camera"
```

```
8. ]
9. },
10. {
11.   "file":
"plugins/org.apache.cordova.camera/www/CameraPopoverOptions.js",
12.   "id": "org.apache.cordova.camera.CameraPopoverOptions",
13.   "clobbers": [
14.     "CameraPopoverOptions"
15.   ]
16. },
17. {
18.   "file": "plugins/org.apache.cordova.camera/www/Camera.js",
19.   "id": "org.apache.cordova.camera.camera",
20.   "clobbers": [
21.     "navigator.camera"
22.   ]
23. },
24. {
25.   "file":
"plugins/org.apache.cordova.camera/www/CameraPopoverHandle.js",
26.   "id": "org.apache.cordova.camera.CameraPopoverHandle",
27.   "clobbers": [
28.     "CameraPopoverHandle"
29.   ]
30. },
31. {
32.   "file": "plugins/org.apache.cordova.dialogs/www/notification.js",
33.   "id": "org.apache.cordova.dialogs.notification",
34.   "merges": [
```

```
35.     "navigator.notification"
36.   ]
37. },
38. {
39.   "file":
"plugins/org.apache.cordova.dialogs/www/android/notification.js",
40.   "id": "org.apache.cordova.dialogs.notification_android",
41.   "merges": [
42.     "navigator.notification"
43.   ]
44. },
45. {
46.   "file": "plugins/org.apache.cordova.vibration/www/vibration.js",
47.   "id": "org.apache.cordova.vibration.notification",
48.   "merges": [
49.     "navigator.notification"
50.   ]
51. },
52. {
53.   "file": "plugins/intent.js",
54.   "id": "org.apache.cordova.intent",
55.   "merges": [
56.     "navigator.intent"
57.   ]
58. },
59. ];
60. module.exports.metadata =
61. // TOP OF METADATA
```

```
62.  {
63.    "org.apache.cordova.camera": "0.2.7",
64.    "org.apache.cordova.dialogs": "0.2.6",
65.    "org.apache.cordova.vibration": "0.3.7",
66.    "org.apache.cordova.intent" : "0.0.1",
67.
68.  }
69.    // BOTTOM OF METADATA
70.  });
```

我之前配置了 camera ,dialog , vibration ,大家可以参考

现在来分解 ,这里要配置 2 个地方

```
module.exports= {};
```

```
module.exports.metadata = { }
```

在 module.exports 的花括号里面配置

```
1. {
2.   "file": "plugins/intent.js",
3.   "id": "org.apache.cordova.intent",
4.   "merges": [
5.     "navigator.intent"
6.   ]
7. },
```

file 代表 javascript 写的接口位置

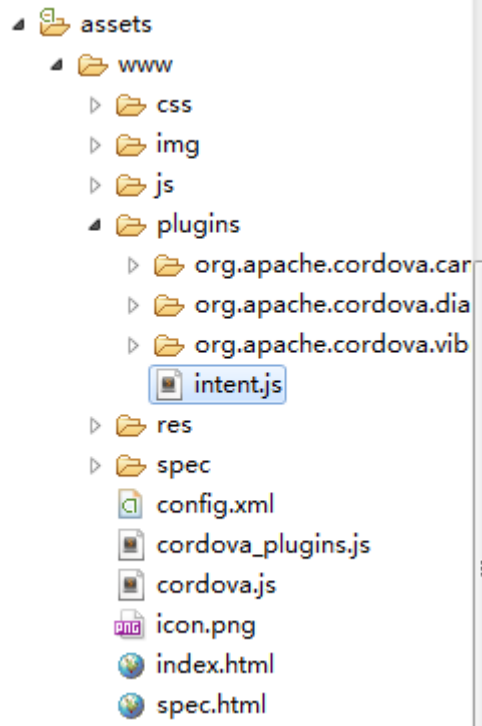
id 代表 唯一

merges 代表你在 javascript 中调用该接口的语句 (类似 activity 中的 `getApplication()` 等等 ;就是个调用语句)

在 module.exports.metadata 中配置 id

标号随意

<4> 在 plugin 目录下编写 javascript 接口



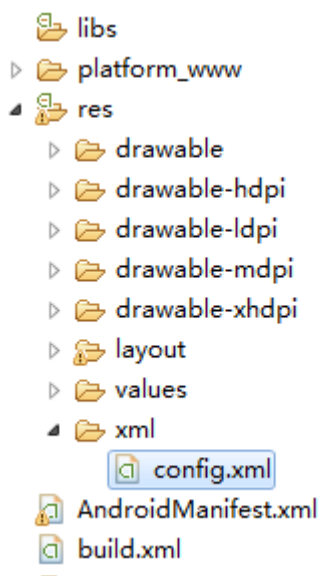
贴上 intent.js 的接口代码

```
1. cordova.define("org.apache.cordova.intent", function(require, exports,
module) {
2.
3. var exec = require('cordova/exec');
4.
5.
6.
7. module.exports = {
8.
9. /**
10.  * 一共 5 个参数
11.  * 第一个 :成功会掉
12.  * 第二个 :失败回调
13.  * 第三个 :将要调用的类的配置名字(在 config.xml 中配置 稍后在下面会讲解)
14.  * 第四个 :调用的方法名(一个类里可能有多个方法 靠这个参数区分)
```

```
15.     第五个 :传递的参数 以 json 的格式
16.     */
17.     demo: function(mills) {
18.         exec(function(winParam){
19.             alert(winParam);
20.         }, null, "Demo", "intent", [mills]);
21.     },
22.     };
23.
24.     });
```

Demo 中成功返回 会弹出一个 Alert();

<5> 在 res/xml 目录下配置 config.xml 文件



config.xml 配置 加上 如下

```
<feature name="Demo">
    <param name="android-package" value="plugins.Plugin_intent" />
</feature>
```

feature 的 name 属性 非常重要

name 必须是步骤< 4 >中 function 中调用的类名



`value` 属性指定插件在 `src` 目录下的 `java` 文件 (命名空间)

example:

插件 Demo 下载地址: <http://download.csdn.net/detail/aaawqqq/6992627>

工程下载 将 `phonegap` 的 `platforms` 导入到 `eclipse` 中

如果报错 `clear` 一下 查看导的 `lib` 包 有没有报错

如果还有错 那么就是您选用了 `google` 的 `API` 改成最新版的 `android API` 就好了

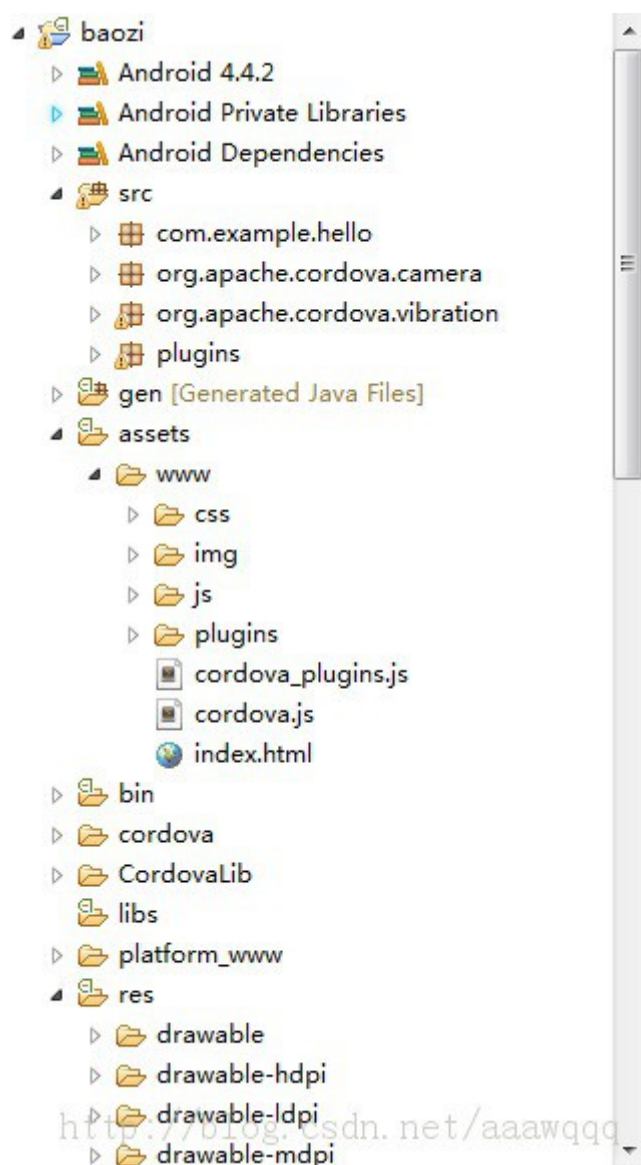
具体排查错误 Blog: <http://blog.csdn.net/aaawqqq/article/details/20463183>

## Phonegap 跳转 Activity android 插件调用原生

插件开发 4 个步骤:

- 1 在 assents 目录下的 cordova-plugins.js 文件添加配置
- 2 在 assets/www 的 plugin 目录下编写 javascript 接口
- 3 在 res/xml 目录下配置 config.xml 文件
- 4 在 src 目录下编写 java 文件 实现跳转

工程:



- 1 配置 cordova\_plugins.js 文件

首先给大家展示 cordova\_plugins.js 文件:

```
cordova.define('cordova/plugin_list', function(require, exports, module) {
```

```
module.exports = [
  {
    "file": "plugins/org.apache.cordova.camera/www/CameraConstants.js",
    "id": "org.apache.cordova.camera.Camera",
    "clobbers": [
      "Camera"
    ]
  },
  {
    "file": "plugins/org.apache.cordova.camera/www/CameraPopoverOptions.js",
    "id": "org.apache.cordova.camera.CameraPopoverOptions",
    "clobbers": [
      "CameraPopoverOptions"
    ]
  },
  {
    "file": "plugins/org.apache.cordova.camera/www/Camera.js",
    "id": "org.apache.cordova.camera.camera",
    "clobbers": [
      "navigator.camera"
    ]
  },
  {
    "file": "plugins/org.apache.cordova.camera/www/CameraPopoverHandle.js",
    "id": "org.apache.cordova.camera.CameraPopoverHandle",
    "clobbers": [
      "CameraPopoverHandle"
    ]
  },
  {
    "file": "plugins/org.apache.cordova.dialogs/www/notification.js",
    "id": "org.apache.cordova.dialogs.notification",
    "merges": [
      "navigator.notification"
    ]
  },
  {
    "file": "plugins/org.apache.cordova.dialogs/www/android/notification.js",
    "id": "org.apache.cordova.dialogs.notification_android",
    "merges": [
      "navigator.notification"
    ]
  },
  {
    "file": "plugins/org.apache.cordova.vibration/www/vibration.js",
    "id": "org.apache.cordova.vibration.notification",
    "merges": [
      "navigator.notification"
    ]
  }
]
```

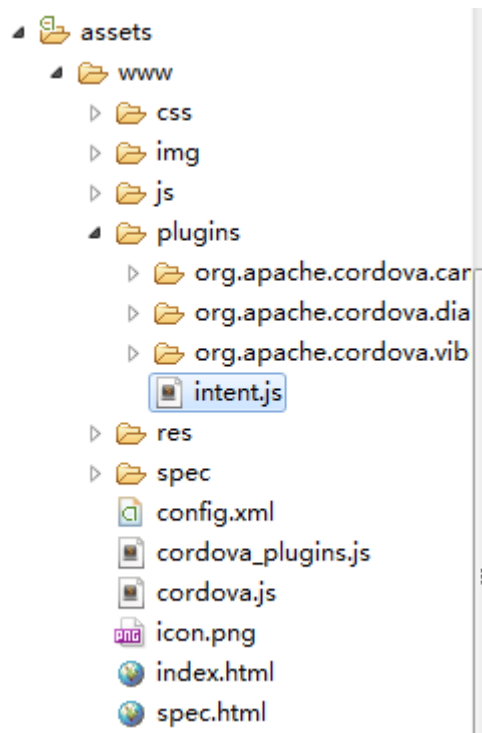
```

    },
    {
      "file": "plugins/intent.js",
      "id": "org.apache.cordova.intent",
      "merges": [
        "navigator.intent"
      ]
    }
  ],
  module.exports.metadata =
  // TOP OF METADATA
  {
    "org.apache.cordova.camera": "0.2.7",
    "org.apache.cordova.dialogs": "0.2.6",
    "org.apache.cordova.vibration": "0.3.7",
    "org.apache.cordova.intent": "0.0.1",

  }
  // BOTTOM OF METADATA
});

```

## 2 在 plugin 目录下编写 javascript 接口



贴上 intent.js 的接口代码

```
cordova.define("org.apache.cordova.intent", function(require, exports, module) {
```

```
var exec = require('cordova/exec');
```

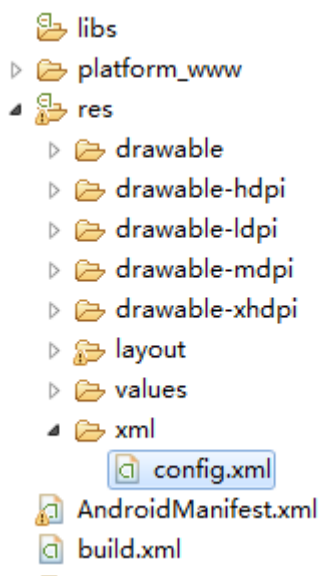
```
module.exports = {
```

```
    /**
     * 一共 5 个参数
     * 第一个 :成功会掉
     * 第二个 :失败回调
     * 第三个 :将要调用的类的配置名字(在 config.xml 中配置 稍后在下面会讲解)
     * 第四个 :调用的方法名(一个类里可能有多个方法 靠这个参数区分)
     * 第五个 :传递的参数 以 json 的格式
     */
    demo: function(mills) {
        exec(function(winParam){
            alert(winParam);
        }, null, "Demo", "intent", [mills]);
    },
};

});
```

Demo 中成功返回 会弹出一个 Alert();

<3> 在 res/xml 目录下配置 config.xml 文件



config.xml 配置 加上 如下

```
<feature name="Demo">
    <param name="android-package" value="plugins.Plugin_intent" />
</feature>
```

feature 的 name 属性 非常重要

name 必须是步骤< 4 >中 function 中调用的类名

value 属性指定插件在 src 目录下的 java 文件 (命名空间)

前 3 个步骤 参考上一章 phonegap 插件开发

这里主要描述

<4> 编写 phonegap 的 java 插件类 调用本地的 activity

首先继承 CordovaPlugin

重写 execute 方法 execute 返回值为 true 时 此插件可用;返回为 false 时 此插件失效

在插件类当中获取 this 跳转

```
Intent intent = new Intent(cordova.getActivity(), DemoActivity.class);
cordova.startActivityForResult((CordovaPlugin) this,intent, 200);
```

如果使用

```
cordova.getActivity().startActivityForResult(intent,200);
```

被调用的 Activity 在 setResult 之后;

并不会返回到当前的插件中 它将返回到的 webView 的 CordovaActivity 当中 ,

Plugin\_intent 类的 onActivityResult 将收不到消息 ;

我查看源码后得知 cordova 这个是 CordovaInterface 类型的 已由 CordovaPlugin 实现

```
*/
public class CordovaActivity extends Activity implements CordovaInterface {
    public static String TAG = "CordovaActivity";

    // The webview for our app
    protected CordovaWebView appView;
    protected CordovaWebViewClient webViewClient;
```

在 CordovaInterface 当中找到了

```
/**
 * Launch an activity for which you would like a result when it finished. When this activity
 * exits,
 * your onActivityResult() method will be called.
 *
 * @param command The command object
 * @param intent The intent to start
```

```
    * @param requestCode The request code that is passed to callback to identify the activity
    */
    abstract public void startActivityResult(CordovaPlugin command, Intent intent, int
requestCode);
```

完整实例

```
package plugins;

import org.apache.cordova.CallbackContext;
import org.apache.cordova.CordovaPlugin;
import org.json.JSONArray;
import org.json.JSONException;

import android.content.Intent;
import android.util.Log;
import android.widget.Toast;

import com.phonegap.helloworld.DemoActivity;

public class Plugin_intent extends CordovaPlugin {

    /**
     * 注意 构造方法不能为
     *
     * Plugin_intent(){}
     *
     * 可以不写或者 定义为如下
     *
     */
    public Plugin_intent() {
    }

    CallbackContext callbackContext;

    @Override
    public boolean execute(String action, JSONArray args,
        CallbackContext callbackContext) throws JSONException {
        this.callbackContext = callbackContext;
        Log.i("123", action);

        if (action.equals("intent")) {
            // 获取 args 的第一个参数
            String message = args.getString(0);
            this.function();
            return true;
        }
        return false;
    }
}
```

```
private void function() {  
    // cordova.getActivity() 获取当前 activity 的 this  
    Log.i("123", cordova.getActivity().toString());  
    Intent intent = new Intent(cordova.getActivity(), DemoActivity.class);  
    cordova.startActivityForResult((CordovaPlugin) this,intent, 200);  
  
}  
  
@Override  
public void onActivityResult(int requestCode, int resultCode, Intent intent) {  
  
    super.onActivityResult(requestCode, resultCode, intent);  
    //传递返回值 给 js 方法  
    callbackContext.success("activity 跳转成功了");  
    Toast.makeText(cordova.getActivity(), "123", 0).show();  
  
}  
}
```

设置返回值

成功返回值:

```
callbackContext.success("成功");
```

失败返回值:

```
callbackContext.error("失败");
```

提供下载 Demo 下载地址: 审核中.....

补 :插件 Demo 下载地址: <http://download.csdn.net/detail/aaaawqqq/6992627>



## ~~phonegap 创建工程 生成指定的包名 类名 应用名~~

phonegap 创建工程 代码

用以创建自己需要的 工程名 ; 包名 ; 类名 ; 应用名

```
cordova create hello com.example.hello HelloWorld
```

```
C:\Develop\phonegap>cordova create hello com.example.hello HelloWorld_
http://blog.csdn.net/aaawqqq
```

分解

cordova create 代表 phonegap 工程的创建命令

hello 代表 phonegap 工程名

com.example.hello 代表 导入 eclipse 后的 类名

HelloWorld 代表 导入 eclipse 后的 工程名 (包括 APP  
)

## ~~phonegap 导入工程报错解决办法 错误解析 android~~

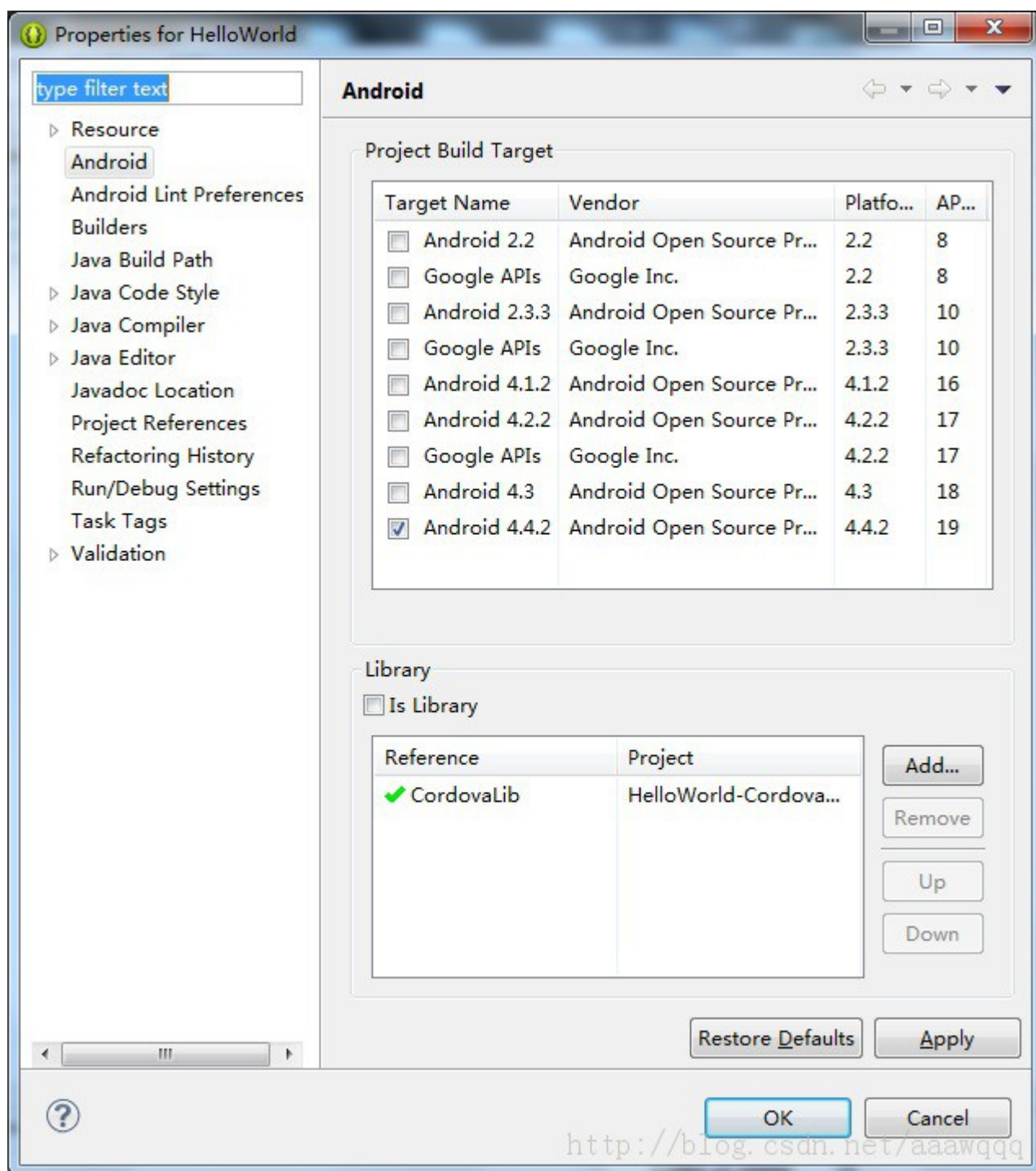
phonegap 导入工程到 eclipse 报错 无非三种情况

- 1 没有导系统 lib 库 或者 错误
- 2 没有选择 android 最新的 API (非 google APIs )
- 3 没有改成 UTF-8 格式

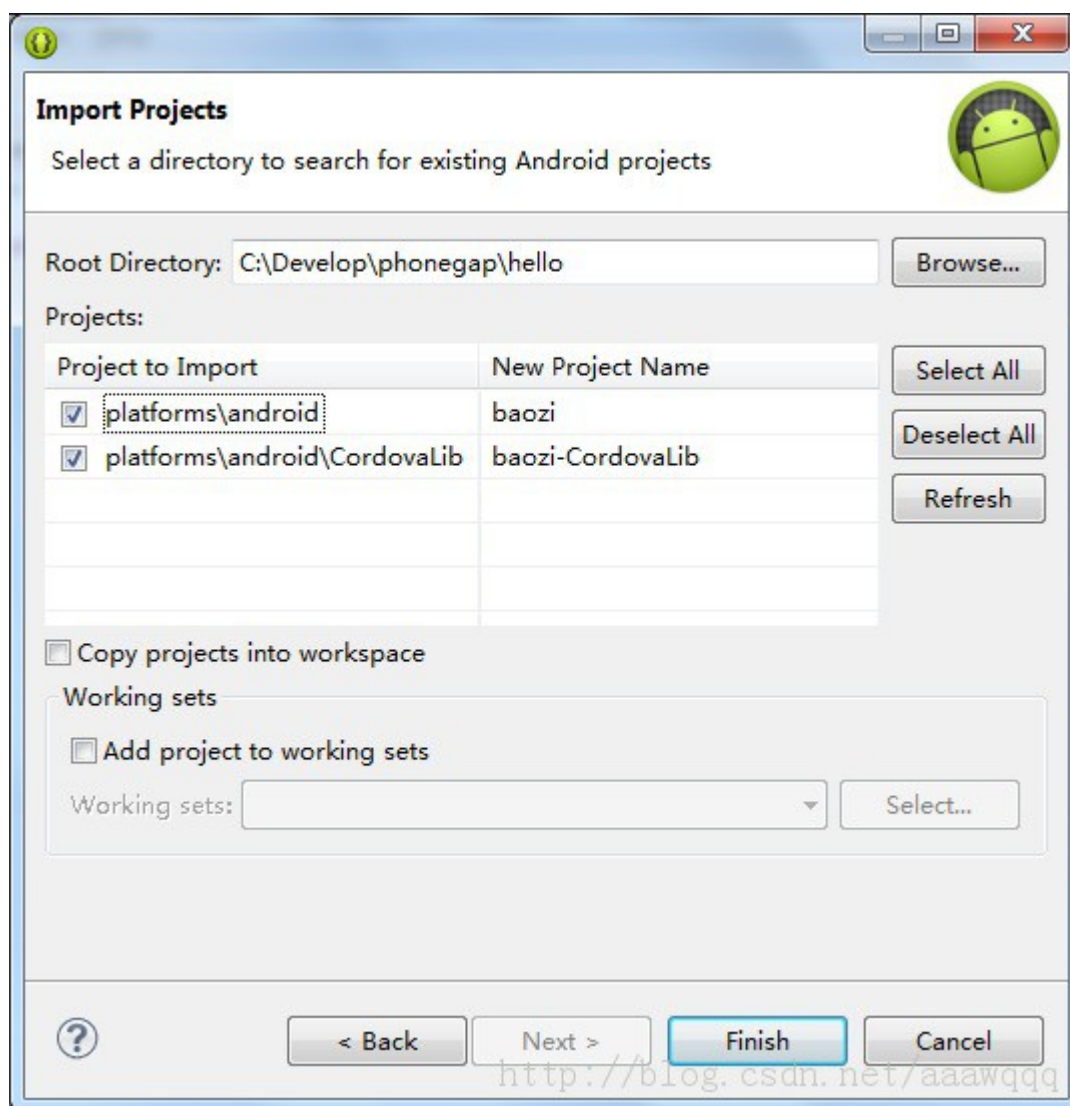
最后 clean 一下就好

-----解决办法-----

先看正确的工程 直接上图



导入工程的时候 是2份 工程的下方是 phonegap 的第三方库



## ~~Phonegap/cordova 升级版本~~

随着 phonegap 的版本升级 大家都需要同步更新

升级是在大家以官方的 node.js 安装方法的基础之上

想重新安装的同学请看:<http://blog.csdn.net/aaawqqq/article/details/19755179>

调用语句：

windows 用户

```
npm update -g cordova
```

如果是 mac 系统的用户 使用:

```
sudo npm update -g cordova
```

查看 cordova 信息

```
npm info cordova
```

查看 cordova 版本

```
cordova -v
```

查看 cordova 命令

```
cordova help
```

## Phonegap 之 config.xml 标签

Phonegap 之 config.xml 标签

在 phonegap 的 res/xml 目下的 config.xml 文件的整个 phonegap 参数配置的关键

包含了 调用本地 API ,自定义插件 ,屏幕 Orientation Fullscreen 等关键配置

核心配置元素 Core Configuration Elements

代码:

```
<widget id="com.example.hello" version="0.0.1">
```

```
<name>HelloWorld</name>
```

```
<description>      A sample Apache Cordova application that responds to the  
deviceready event.    </description>
```

```

<author email="dev@callback.apache.org" href="http://phonegap.com"> Apache
Cordova Team </author>

<content src="index.html" />

<access origin="*" />

</widget>

```

此核心配置 支持所有的平台。

**<widget>** 元素的 **id** 属性提供应用的反向域名定义, **version** 的 3 个数字: 主要/小的/补丁 符号

**<name>** 元素指定 **app** 正式的名字 在 **home screen** 和 **app-store** 列表

**<description>**和**<author>** 元素指定 元数据和联系信息 (描述和作者)

**<content>** 元素定义 应用的开始页面 在顶级网络资产目录中 .默认定义的值是 **index.html** 。这通常出现在一个项目的顶级 **www** 目录。

**<access>** 元素定义的一组应用程序可以与外部域 。上面所示的默认值允许它访问任何服务器。有关详细信息,请参阅域名白名单指南。

**<preference>** 是全局的参数 标签设置多种选项 , 参数有一对 **name/value** 属性 。每个 **name** 的值区分大小写

例如: 应用 是否 全屏设置

```

<preference name="Fullscreen" value="true" />

```

**<feature>**标签 在 调用系统的 **api** 和 自定义插件开发时用到

如下是 调用 **phonegap** 联系人的 **feature** 标签

```

<!-- 联系人 -->

<feature name="Contacts" >

  <param

    name="android-package"

    value="org.apache.cordova.contacts.ContactManager" />

</feature>

```

## Phonegap 的 deviceready 事件

该事件是在 PhoneGap 载入完成后 发生的事件

它表示 cordova 加载完成并准备访问 . yourCallbackFunction

相当于程序的入口功能

如下 onDeviceReady 函数

```
1. document.addEventListener("deviceready", yourCallbackFunction, false);
```

简单例子:

```
1. document.addEventListener("deviceready", onDeviceReady, false);
2. function onDeviceReady() {
3.     // Now safe to use device APIs
4.
5. }
```

完整例子:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Device Ready Example</title>
5. <script type="text/javascript" charset="utf-8" src="cordova.js"></script>
6. <script type="text/javascript" charset="utf-8">
7.     // Wait for device API libraries to load //
```

```
8. function onLoad() {  
9. document.addEventListener("deviceready", onDeviceReady, false);  
10. } // device APIs are available //  
11. function onDeviceReady() {  
12. // Now safe to use device APIs  
13.  
14. }  
15. </script>  
16. </head>  
17. <body onload="onLoad()">  
18. </body>  
19. </html>
```

支持:

1. •Amazon Fire OS
2. •Android
3. •BlackBerry 10
4. •iOS
5. •Tizen
6. •Windows Phone 7 and 8
7. •Windows 8



## Phonegap 网络连接状态 online/onffline html5

### Phonegap 网络连接状态 html5

在当前的界面自动判断是否连接上网络 或者 断网

事件监听为 `offline` 和 `online` 提供 Demo 下载

cordova 3.4 提供了新的 API `org.apache.cordova.network-information`

将原来的 Events 的 `online` 和 `offline` 拆分了出来 并入了 `network-information` 模块支持的系统:

- Amazon Fire OS
- Android
- BlackBerry 10
- iOS
- Windows Phone 7 and 8
- Tizen
- Windows 8

下载插件:

```
cordova plugin add org.apache.cordova.network-information
```

例子:

```
function onDeviceReady() {  
    document.addEventListener("offline", onOffline, false);  
    document.addEventListener("online", onOnline, false);  
  
}  
  
function onOffline(e) {  
    // Handle the offline event  
  
    alert("网络连接未连接");  
}  
  
function onOnline() {  
    // Handle the offline event  
    alert("网络连接已连接");  
}
```

当连上网的时候 会自动执行 `onOnline` 方法

当断网的时候 会自动执行 `onOffline` 方法

android 问题:

自己动手检测后发现 : 如果没有网络 第一次启动的时候 `onOffline` 会调用 2 次

ios 问题:

官方文档说 启动需要 1 秒后才回有反应

Demo 演示图:



## Phonegap 事件之菜单键 html5 菜单事件

Phonegap 拦截菜单按键的事件 `menubutton`, 用于处理 html5 当中的菜单拦截事件  
此事件是当用户按下菜单是时候触发

```
document.addEventListener("menubutton", yourCallbackFunction, false);
```

支持平台:

- Amazon Fire OS
- Android
- BlackBerry 10

简单例子:

```
document.addEventListener("menubutton", onMenuKeyDown, false);
function onMenuKeyDown() {
    // Handle the back button
}
```

完整例子:

```
<html> <head> <title>Menu Button Example</title>
<script type="text/javascript" charset="utf-8" src="cordova.js"></script>
<script type="text/javascript" charset="utf-8">
// Wait for device API libraries to load //
function onLoad() {
    document.addEventListener("deviceready", onDeviceReady, false);
}
// device APIs are available //
function onDeviceReady() {
    // Register the event listener
    document.addEventListener("menubutton", onMenuKeyDown, false);
}
// Handle the menu button //
function onMenuKeyDown() {
    //菜单按钮的执行
}
</script>
</head>
<body onload="onLoad()"> </body>
</html>
```

`deviceready` 只需要加载一次即可 它表示当 phonegap 的 API 加载完毕的时候调用

## Phonegap 事件之 resume 事件

Phonegap 事件之 resume 事件

当 应用重新从后台恢复到前台时 事件触发

类似 android 中 `activity` 生命周期里的 `resume`

支持平台:

- Amazon Fire OS

- Android
- BlackBerry 10
- iOS
- Windows Phone 7 and 8
- Windows 8

简单例子:

```
document.addEventListener("resume", onResume, false);
function onResume() {
  // Handle the resume event
}
```

完整例子:

```
<!DOCTYPE html>
<html>
<head>
<title>Resume Example</title>
<script type="text/javascript" charset="utf-8" src="cordova.js"></script>
<script type="text/javascript" charset="utf-8"> // Wait for device API libraries to load //
function onLoad() {
  document.addEventListener("deviceready", onDeviceReady, false);
} // device APIs are available //

function onDeviceReady() {
  document.addEventListener("resume", onResume, false);
}

// Handle the resume event //
function onResume() {
  //回到前台后 调用这里
}
</script>
</head>
<body onload="onLoad()"> </body>
</html>
```

IOS 遇到的问题:

当调用 resume 事件处理时

互动功能如 alert()需要包裹在 setTimeout()() 里面

调用的超时时间值为零,否则应用程序挂起。例如:

```
document.addEventListener("resume", onResume, false);
function onResume() {
  setTimeout(function() {
    // TODO: do your thing!
  }, 0);
}
```



## Phonegap 事件之 pause 事件

Phonegap 暂停事件

pause 事件

类似 android 的生命周期 **pause** 代码功能是当前台应用将要转向后台时调用 (类似点击了 home 键后回到了桌面 )

可以用 **pause** 事件保存数据

应用应使用 **document.addEventListener** 去加载

简单例子:

```
document.addEventListener("pause", onPause, false);
function onPause() {
// Handle the pause event
}
```

完整例子:

```
<!DOCTYPE html><html> <head>
<title>Pause Example</title>
<script type="text/javascript" charset="utf-8" src="cordova.js"></script>
<script type="text/javascript" charset="utf-8"> // Wait for device API libraries to load //
function onLoad() { document.addEventListener("deviceready", onDeviceReady, false); }
// device APIs are available //
function onDeviceReady() {
    document.addEventListener("pause", onPause, false);

}
// Handle the pause event //
function onPause() {
    //onpause 的执行
}

</script>
</head>
<body onload="onLoad()"> </body>
</html>
```

支持的系统:

- Amazon Fire OS
- Android
- BlackBerry 10
- iOS

- Windows Phone 7 and 8
- Windows 8

ios 在使用过程中的问题:

[http://docs.phonegap.com/en/3.4.0/cordova\\_events\\_events.md.html#pause](http://docs.phonegap.com/en/3.4.0/cordova_events_events.md.html#pause)

## Phonegap 返回键 backbutton 功能 cordova

Phonegap 拦截返回键 这个功能是当用户按下返回键事件触发.

backbutton 监听.调用 回调函数 如下:

```
document.addEventListener("backbutton", yourCallbackFunction, false);
```

注册事件监听,代表性的调用 `document.addEventListener`,一旦你注册了 `deviceready` 事件不再需要调用任何其他方法来覆盖 `theback-button` 行为。

支持平台:

- Amazon Fire OS
- Android

- BlackBerry 10
- Windows Phone 7 and 8

简单例子:

```
document.addEventListener("backbutton", onBackKeyDown, false);
function onBackKeyDown() {
    // Handle the back button
}
```

完整例子:

```
<!DOCTYPE html><html> <head>
<title>Back Button Example</title>
<script type="text/javascript" charset="utf-8" src="cordova.js"></script>
<script type="text/javascript" charset="utf-8">
// Wait for device API libraries to load //
function onLoad() {
    document.addEventListener("deviceready", onDeviceReady, false);
}
// device APIs are available //
function onDeviceReady() { // Register the event listener
    document.addEventListener("backbutton", onBackKeyDown, false); }
    // Handle the back button //
    function onBackKeyDown() {

    }
}
</script>
</head>
<body onload="onLoad()"> </body>
</html>
```

按下返回键后 `function onBackKeyDown()` 就会执行 ;

把将要执行的方法放在里面就行



## Phonegap 获取 imei (修改 phonegap 官方提供文件)

Phonegap 获取手机设备信息 IMEI

phonegap 的 Device 提供有:

**device.model** :返回设备的模型或产品的名称

**device.cordova** :返回 cordova 的版本

**device.uuid** :返回手机 uuid

**device.version** :返回系统版本

**device.platform** :返回手机的平台信息 (android/ios 等等)

唯独没有 Imei 的获取方法

本节在 phonegap 提供的插件上 增加一个 imei 的获取

Device 调用方法参考 <http://blog.csdn.net/aaawqqq/article/details/21169587>

<1> 在控制台 创建一个 phonegap 工程 命令如下

```
1. phonegap create my-app
2. cd my-app
3. phonegap run android
```

```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留
C:\Users\Baozi>phonegap create my-app_

http://blog.csdn.net/aaawqqq
```

<2> 我们从命令行进入 到工程目录下的 plugins 文件夹

1. cd my-app
- 2.
3. cd plugins

```
C:\Develop\phonegap>phonegap create my-app
[phonegap] created project at C:\Develop\phonegap\my-app
C:\Develop\phonegap>cd my-app
C:\Develop\phonegap\my-app>cd plugins
C:\Develop\phonegap\my-app\plugins>

http://blog.csdn.net/aaawqqq
```

<3> 现在开始下载插件

cordova plugin add org.apache.cordova.device

1. cordova plugin add org.apache.cordova.device

```
C:\Users\Baozi>cd C:\Develop\phonegap\Constacts\plugins
C:\Develop\phonegap\Constacts\plugins>cordova plugin add org.apache.cordova.device
Fetching plugin "org.apache.cordova.device" via plugin registry
Starting installation of "org.apache.cordova.device" for android
Preparing android project
org.apache.cordova.device installed on android
http://blog.csdn.net/aaawqqq
```

<4> 添加 android 平台工程 (ios 把 "android" 替换)

cordova platform add android

```
C:\Develop\phonegap\my-app\plugins>cordova platform add android
Creating android project...
Preparing android project
Starting installation of "org.apache.cordova.camera" for android
Preparing android project
org.apache.cordova.camera installed on android.
C:\Develop\phonegap\my-app\plugins>
```

<http://blog.csdn.net/aaawqqq>

#### <5> 编译 android 工程

1. cordova build

至此 devices 已经生成...

现在大家只需要将工程导入到 eclipse 当中 使用官方 API 语句在 javascript 中调用

要获取 imei 需要改动 原生的 Device 2 个地方

第一个是 assets 目录下 www/plugins 里面的 org.apache.cordova.device / device.js

```
1. cordova.define("org.apache.cordova.device.device", function(require,
    exports, module) { /*
2. *
3. * Licensed to the Apache Software Foundation (ASF) under one
4. * or more contributor license agreements. See the NOTICE file
5. * distributed with this work for additional information
6. * regarding copyright ownership. The ASF licenses this file
7. * to you under the Apache License, Version 2.0 (the
8. * "License"); you may not use this file except in compliance
9. * with the License. You may obtain a copy of the License at
10. *
11. * http://www.apache.org/licenses/LICENSE-2.0
12. *
13. * Unless required by applicable law or agreed to in writing,
14. * software distributed under the License is distributed on an
15. * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
16. * KIND, either express or implied. See the License for the
```

```
17. * specific language governing permissions and limitations
18. * under the License.
19. *
20. */
21.
22. var argscheck = require('cordova/argscheck'),
23.     channel = require('cordova/channel'),
24.     utils = require('cordova/utils'),
25.     exec = require('cordova/exec'),
26.     cordova = require('cordova');
27.
28. channel.createSticky('onCordovaInfoReady');
29. // Tell cordova channel to wait on the CordovaInfoReady event
30. channel.waitForInitialization('onCordovaInfoReady');
31.
32. /**
33. * This represents the mobile device, and provides properties for inspecting
34. * the model, version, UUID of the
35. * phone, etc.
36. * @constructor
37. */
38. function Device() {
39.     this.available = false;
40.     this.platform = null;
41.     this.version = null;
42.     this.uuid = null;
43.     this.cordova = null;
44.     this.model = null;
```

```
44. //添加 imei
45. this.imei = null;
46.
47. var me = this;
48.
49. channel.onCordovaReady.subscribe(function() {
50. me.getInfo(function(info) {
51. //ignoring info.cordova returning from native, we should use value from
    cordova.version defined in cordova.js
52. //TODO: CB-5105 native implementations should not return info.cordova
53. var buildLabel = cordova.version;
54. me.available = true;
55. me.platform = info.platform;
56. me.version = info.version;
57. me.uuid = info.uuid;
58. me.cordova = buildLabel;
59. me.model = info.model;
60. //添加 imei
61. me.imei = info.imei;
62. channel.onCordovaInfoReady.fire();
63. },function(e) {
64. me.available = false;
65. utils.alert("[ERROR] Error initializing Cordova: " + e);
66. });
67. });
68. }
69.
70. /**
```

```
71. * Get device info
72. *
73. * @param {Function} successCallback The function to call when the heading
    data is available
74. * @param {Function} errorCallback The function to call when there is an
    error getting the heading data. (OPTIONAL)
75. */
76. Device.prototype.getInfo = function(successCallback, errorCallback) {
77.   argscheck.checkArgs('fF', 'Device.getInfo', arguments);
78.   exec(successCallback, errorCallback, "Device", "getDeviceInfo", []);
79. };
80.
81. module.exports = new Device();
82.
83. });
```

第二个是在 `src` 当中 修改 `device` 类

```
1.  /*
2.   Licensed to the Apache Software Foundation (ASF) under one
3.   or more contributor license agreements. See the NOTICE file
4.   distributed with this work for additional information
5.   regarding copyright ownership. The ASF licenses this file
6.   to you under the Apache License, Version 2.0 (the
7.   "License"); you may not use this file except in compliance
8.   with the License. You may obtain a copy of the License at
9.
10.  http://www.apache.org/licenses/LICENSE-2.0
11.
12.  Unless required by applicable law or agreed to in writing,
```

```
13. software distributed under the License is distributed on an
14. "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
15. KIND, either express or implied. See the License for the
16. specific language governing permissions and limitations
17. under the License.
18. */
19. package org.apache.cordova.device;
20.
21. import java.util.TimeZone;
22.
23. import org.apache.cordova.CallbackContext;
24. import org.apache.cordova.CordovaInterface;
25. import org.apache.cordova.CordovaPlugin;
26. import org.apache.cordova.CordovaWebView;
27. import org.json.JSONArray;
28. import org.json.JSONException;
29. import org.json.JSONObject;
30.
31. import android.content.Context;
32. import android.provider.Settings;
33. import android.telephony.TelephonyManager;
34. import android.util.Log;
35.
36. public class Device extends CordovaPlugin {
37. public static final String TAG = "Device";
38.
39. public static String cordovaVersion = "dev"; // Cordova version
40. public static String platform; // Device OS
```

```
41. public static String uuid; // Device UUID
42.
43. private static final String ANDROID_PLATFORM = "Android";
44. private static final String AMAZON_PLATFORM = "amazon-fireos";
45. private static final String AMAZON_DEVICE = "Amazon";
46.
47. /**
48. * Constructor.
49. */
50. public Device() {
51. }
52.
53. /**
54. * Sets the context of the Command. This can then be used to do things like
55. * get file paths associated with the Activity.
56. *
57. * @param cordova The context of the main Activity.
58. * @param webView The CordovaWebView Cordova is running in.
59. */
60. public void initialize(CordovaInterface cordova, CordovaWebView webView) {
61. super.initialize(cordova, webView);
62. Device.uuid = getUuid();
63. }
64.
65. /**
66. * Executes the request and returns PluginResult.
67. *
68. * @param action The action to execute.
```



```
69. * @param args JSONArray of arguments for the plugin.
70. * @param callbackContext The callback id used when calling back into
    JavaScript.
71. * @return True if the action was valid, false if not.
72. */
73. public boolean execute(String action, JSONArray args, CallbackContext
    callbackContext) throws JSONException {
74. if (action.equals("getDeviceInfo")) {
75. JSONObject r = new JSONObject();
76. r.put("uuid", Device.uuid);
77. r.put("version", this.getOSVersion());
78. r.put("platform", this.getPlatform());
79. r.put("cordova", Device.cordovaVersion);
80. r.put("model", this.getModel());
81. //添加 imei 的返回值
82. r.put("imei", this.imei());
83. callbackContext.success(r);
84. }
85. else {
86. return false;
87. }
88. return true;
89. }
90.
91. //-----
92. // LOCAL METHODS
93. //-----
94.
95. // 获取本地 Imei 号码
```

```
96. private String imei() {
97. // String Imei = ((TelephonyManager)
    cordova.getActivity().getSystemService(cordova.getActivity().TELEPHONY_SERVI
    CE))
98. // .getDeviceId();
99. // return Imei;
100.
101. TelephonyManager systemService =
    (TelephonyManager)cordova.getActivity().getSystemService(Context.TELEPHONY_S
    ERVICE);
102. String deviceId = systemService.getDeviceId();
103. Log.i("123", deviceId);
104.
105. return systemService.getDeviceId();
106. }
107.
108. /**
109. * Get the OS name.
110. *
111. * @return
112. */
113. public String getPlatform() {
114. String platform;
115. if (isAmazonDevice()) {
116. platform = AMAZON_PLATFORM;
117. } else {
118. platform = ANDROID_PLATFORM;
119. }
120. return platform;
```

```
121. }  
  
122.  
  
123. /**  
  
124. * Get the device's Universally Unique Identifier (UUID).  
  
125. *  
126. * @return  
  
127. */  
  
128. public String getUuid() {  
129. String uuid =  
130.     Settings.Secure.getString(this.cordova.getActivity().getContentResolver(),  
131.         android.provider.Settings.Secure.ANDROID_ID);  
132.  
133. return uuid;  
134. }  
  
135.  
  
136. /**  
  
137. * Get the Cordova version.  
  
138. *  
139. * @return  
  
140. */  
  
141. public String getCordovaVersion() {  
142. return Device.cordovaVersion;  
143. }  
  
144.  
  
145. public String getModel() {  
146. String model = android.os.Build.MODEL;  
147. return model;  
148. }  
  
149.  
  
150. public String getProductName() {
```

```
148. String productname = android.os.Build.PRODUCT;
149. return productname;
150. }
151.
152. /**
153.  * Get the OS version.
154.  *
155.  * @return
156.  */
157. public String getOSVersion() {
158. String osversion = android.os.Build.VERSION.RELEASE;
159. return osversion;
160. }
161.
162. public String getSDKVersion() {
163. @SuppressWarnings("deprecation")
164. String sdkversion = android.os.Build.VERSION.SDK;
165. return sdkversion;
166. }
167.
168. public String getTimeZoneID() {
169. TimeZone tz = TimeZone.getDefault();
170. return (tz.getID());
171. }
172.
173. /**
174.  * Function to check if the device is manufactured by Amazon
175.  *
```

```
176. * @return
177. */
178. public boolean isAmazonDevice() {
179. if (android.os.Build.MANUFACTURER.equals(AMAZON_DEVICE)) {
180. return true;
181. }
182. return false;
183. }
184.
185. }
```

最后 在 `androidmanifest.xml` 当中 添加权限

```
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
```

这是 获取 imei 必要的权限

修改完成!

将下方语句考到 `assets` 目录下 `www/index.html` 当中 完全复制过去;

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5. <title>Contact Example</title>
6. <script type="text/javascript" charset="utf-8" src="cordova.js"></script>
7. <script type="text/javascript" charset="utf-8">
8. // Wait for device API libraries to load
9. // document.addEventListener("deviceready", onDeviceReady, false);
10. // device APIs are available
11.
12. function onDeviceReady() {
```

```
13.
14. // Wait for device API libraries to load //
15.
16. //var element = document.getElementById('deviceProperties');
17. //alert(element);
18.
19. // element.innerHTML =
20. // 'Device Model: ' + device.model + '<br />' +
21. // 'Device Cordova: ' + device.cordova + '<br />' +
22. // 'Device Platform: ' + device.platform + '<br />' +
23. // 'Device UUID: ' + device.uuid + '<br />' +
24. // 'Device Version: ' + device.version + '<br />';
25.
26.
27. alert( device.model + "----"+device.cordova + "-----"+ device.uuid
    + "-----"+device.version+"----"+device.platform );
28. }
29.
30. function baozi(){ alert("S1");}
31.
32. function intent() { onDeviceReady(); }
33.
34.
35. </script>
36. </head>
37. <body>
38. <h1>Example</h1>
39. <p>Find Contacts</p>
```

40.

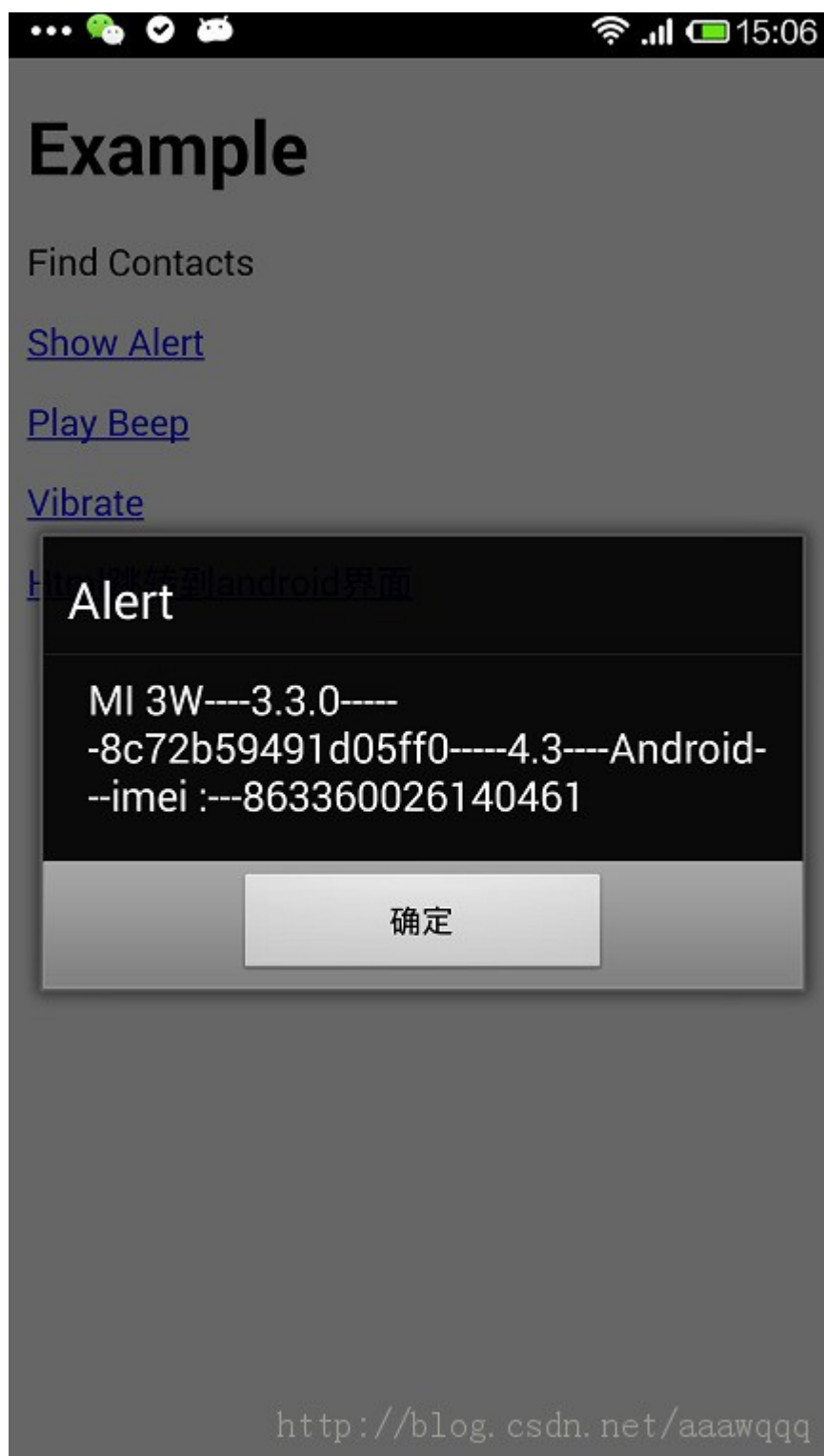
41. `<p><a href="#" onclick="baozi(); return false;">Vibrate</a></p>`

42. `<p><a href="#" onclick="intent(); return false;">Html 跳转到 android 界面</a></p>`

43. `</body>`

44. `</html>`

点击 " Html 跳转到 android 界面 " 就会弹出 **alert** 如下:



依次

device.model :返回设备的模型或产品的名称

device.cordova :返回 cordova 的版本

device.uuid :返回手机 uuid

device.version :返回系统版本



`device.platform` :返回手机的平台信息 (`android/ios` 等等)

第六个参数就是 `Imei` 号码

## Phonegap 联系人 获取 Contacts

Phonegap 获取 所有联系人 此版本是 `cordova 3.4` 的

调用系统的 `API` 经过修改 迭代出所有联系人 贴上完整代码

文章结尾 提供 `Demo` 下载

以下介绍的 `phonegap contacts` 的 `API` 有

## methods

- [contacts.create](#)
- [contacts.find](#)

## Arguments

- [contactFields](#)
- [contactSuccess](#)
- [contactError](#)
- [contactFindOptions](#)

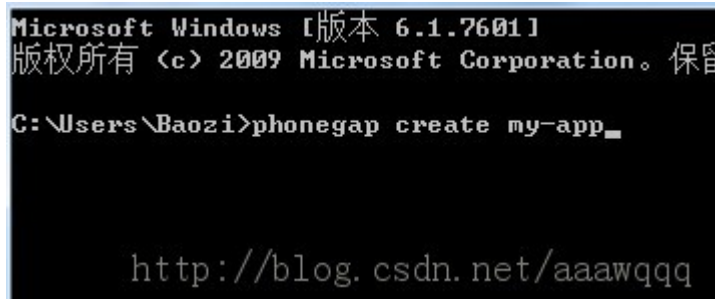
## Object

- [Contact](#)
- [ContactName](#)
- [ContactField](#)
- [ContactAddress](#)
- [ContactOrganization](#)
- [ContactFindOptions](#)
- [ContactError](#)

从建立项目说起:

<1> 在控制台 创建一个 **phonegap** 工程 命令如下

```
phonegap create my-app  
cd my-app  
phonegap run android
```



## <2> 我们从命令行进入到工程目录下的 **plugins** 文件夹

```
cd my-app
```

```
cd plugins
```

```
C:\Develop\phonegap>phonegap create my-app
[phonegap] created project at C:\Develop\phonegap\my-app

C:\Develop\phonegap>cd my-app

C:\Develop\phonegap\my-app>cd plugins

C:\Develop\phonegap\my-app\plugins>

http://blog.csdn.net/aaawqqq
```

## <3> 现在开始下载插件

```
cordova plugin add org.apache.cordova.contacts
```

```
C:\Users\Baozi>cd C:\Develop\phonegap\my-app\plugins

C:\Develop\phonegap\my-app\plugins>cordova plugin add org.apache.cordova.contacts
Fetching plugin "org.apache.cordova.contacts" via plugin registry
Starting installation of "org.apache.cordova.contacts" for android
Preparing android project
org.apache.cordova.contacts installed on android.

C:\Develop\phonegap\my-app\plugins>

http://blog.csdn.net/aaawqqq
```

## <4> 添加 **android** 平台工程 (ios 把 "android" 替换)

```
cordova platform add android
```

```
C:\Develop\phonegap\my-app\plugins>cordova platform add android
Creating android project...
Preparing android project
Starting installation of "org.apache.cordova.camera" for android
Preparing android project
org.apache.cordova.camera installed on android.

C:\Develop\phonegap\my-app\plugins>

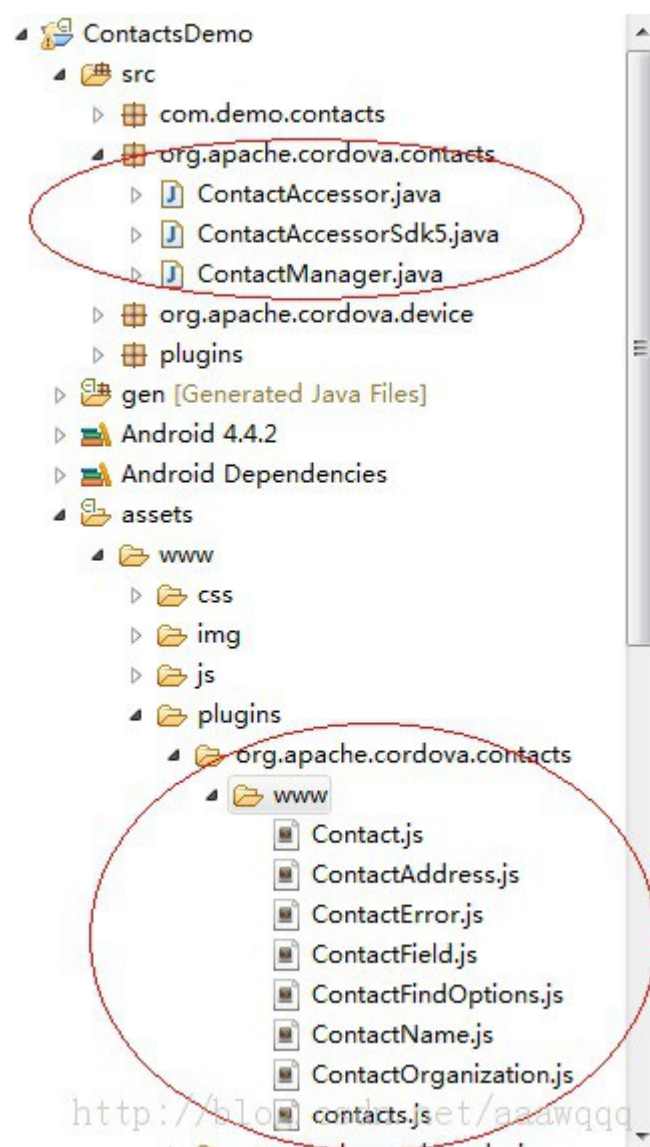
http://blog.csdn.net/aaawqqq
```

## <5> 编译 android 工程

cordova build

至此 **contacts** 插件已经生成并添加到项目当中去...

现在项目应该 有如下文件:



cordova-plugins.js 里面的文件是:

```
cordova.define('cordova/plugin_list', function(require, exports,
module) {
module.exports = [
{
```

```
        "file":
"plugins/org.apache.cordova.contacts/www/contacts.js",
        "id": "org.apache.cordova.contacts.contacts",
        "clobbers": [
            "navigator.contacts"
        ]
    },
    {
        "file": "plugins/org.apache.cordova.contacts/www/Contact.js",
        "id": "org.apache.cordova.contacts.Contact",
        "clobbers": [
            "Contact"
        ]
    },
    {
        "file":
"plugins/org.apache.cordova.contacts/www/ContactAddress.js",
        "id": "org.apache.cordova.contacts.ContactAddress",
        "clobbers": [
            "ContactAddress"
        ]
    },
    {
        "file":
"plugins/org.apache.cordova.contacts/www/ContactError.js",
        "id": "org.apache.cordova.contacts.ContactError",
        "clobbers": [
            "ContactError"
        ]
    },
    {
        "file":
"plugins/org.apache.cordova.contacts/www/ContactField.js",
        "id": "org.apache.cordova.contacts.ContactField",
        "clobbers": [
```

```
        "ContactField"
    ],
    },
    {
        "file":
        "plugins/org.apache.cordova.contacts/www/ContactFindOptions.js",
        "id": "org.apache.cordova.contacts.ContactFindOptions",
        "clobbers": [
            "ContactFindOptions"
        ]
    },
    {
        "file":
        "plugins/org.apache.cordova.contacts/www/ContactName.js",
        "id": "org.apache.cordova.contacts.ContactName",
        "clobbers": [
            "ContactName"
        ]
    },
    {
        "file":
        "plugins/org.apache.cordova.contacts/www/ContactOrganization.js",
        "id": "org.apache.cordova.contacts.ContactOrganization",
        "clobbers": [
            "ContactOrganization"
        ]
    }
];
module.exports.metadata =
// TOP OF METADATA
{
    "org.apache.cordova.contacts": "0.2.8"
}
// BOTTOM OF METADATA
});
```

res/xml 目录下 的 config.xml



在 config.xml 里面有

```
<feature name="Contacts">
    <param name="android-package"
value="org.apache.cordova.contacts.ContactManager" />
</feature>
```

现在大家只需要将工程导入到 **eclipse** 当中 使用官方 **API** 语句在 **javascript** 中调用

将下方语句考到 **assets** 目录下 **www/index.html** 当中 完全复制过去;

## example:

```
<!DOCTYPE html>
<html>
<head>
    <title>Contact Example</title>
    <script type="text/javascript" charset="utf-8"
src="cordova.js"></script>
    <script type="text/javascript" charset="utf-8">
        // Wait for device API libraries to load
        document.addEventListener("deviceready", onDeviceReady, false);
        // device APIs are available

        function onDeviceReady() {
            // find all contacts with ' ' in any name field ,field 如果为
            空 则返回全部的
```

```
var options = new ContactFindOptions();
options.filter = "";
options.multiple = true;
/*
查找关键字
名字: "displayName" ,
电话号码:"phoneNumbers"    //ContactField[]类型

*/
var fields = ["displayName", "name", "phoneNumbers"];
navigator.contacts.find(fields, onSuccess, onError, options);
}

// onSuccess: Get a snapshot of the current contacts
function onSuccess(contacts) {

    // 联系人与电话号码 全写在这儿
    var aResult = [];

    for (var i = 0; contacts[i]; i++) {
        console.log("Display Name = " + contacts[i].displayName);

        if (contacts[i].phoneNumbers &&
contacts[i].phoneNumbers.length) {

            var contactPhoneList = [];

            for (var j = 0; contacts[i].phoneNumbers[j]; j++) {
                // alert(contacts[i].phoneNumbers[j].type+"
"+contacts[i].displayName+"-----" +
contacts[i].phoneNumbers[j].value );

                contactPhoneList.push(
                    {
```



```
        'type' : contacts[i].phoneNumbers[j].type,
        'value' : contacts[i].phoneNumbers[j].value
    }
    );

};

aResult.push({
    name:contacts[i].displayName,
    phone:contactPhoneList
});

};

//
}

//迭代获取 联系人和号码
for (var i = 0; aResult[i]; i++) {
    for (var j = 0 ; aResult[i].phone[j]; j++) {

        alert(aResult[i].name + "-----"+
aResult[i].phone[j].type+"-----"+aResult[i].phone[j].value
        );
    };

};

}

// onError: Failed to get the contacts
function onError(contactError)
{
    alert('onError!');
}

function baozi() { alert("S1");}

function intent() {    onDeviceReady();    }
```

```
</script>
</head>
<body>
  <h1>Example</h1>
  <p>Find Contacts</p>

  <p><a href="#" onclick="showAlert(); return false;">Show
Alert</a></p>
  <p><a href="#" onclick="playBeep(); return false;">Play Beep</a></p>
  <p><a href="#" onclick="baozi(); return false;">Vibrate</a></p>
  <p><a href="#" onclick="intent(); return false;">Html 跳转到 android
界面</a></p>
</body>
</html>
```

## 代码详解:

上段代码 中的

```
var options = new ContactFindOptions();
  options.filter = "";
  options.multiple = true;
  var fields = ["displayName", "phoneNumbers"];
  navigator.contacts.find(fields, onSuccess, onError, options);
}
```

**options.filter** //过滤条件

**options.multiple** //是否 查询多个

**fields** //将要查询的 关键字

**navigator.contacts.find (fields,onSuccess,onError , options );** // 查找操作

其中的 **onSuccess** 是成功找到联系人后 返回后将执行的回调

查找关键字

名字: "displayName",

电话号码:"phoneNumbers" //ContactField[]类型

在注释之前 是js 里面获取 所有联系的方式

注释之后是 方便测试的弹出窗口

点击 " Html 跳转到 android 界面 " 就会弹出 alert 如下:

迭代 弹出所有联系人



至此 获取了所有的电话号码

顺道说一句

```
if (contacts[i].phoneNumbers && contacts[i].phoneNumbers.length) {}
```

上面这一句是判断 联系人是否为空

# Phonegap 是否连接网络状态 online/offline

Phonegap 网络连接状态 html5

在当前的界面自动判断是否连接上网络 或者 断网

事件监听为 **offline** 和 **online** 提供 Demo 下载

cordova 3.4 提供了新的 API `org.apache.cordova.network-information`

将原来的 Events 的 **online** 和 **offline** 拆分了出来 并入了 `network-information` 模块

支持的系统:

- Amazon Fire OS
- Android
- BlackBerry 10
- iOS
- Windows Phone 7 and 8
- Tizen
- Windows 8

下载插件:

```
1. cordova plugin add org.apache.cordova.network-information
```

```
1. function onDeviceReady() {  
2.     document.addEventListener("offline", onOffline, false);  
3.     document.addEventListener("online", onOnline, false);  
4.  
5.  
6. }  
7.  
8. function onOffline(e) {  
9.     // Handle the offline event  
10.  
11. alert("网络连接未连接");
```

```
12. }  
13.  
14. function onOnline() {  
15. // Handle the offline event  
16. alert("网络连接已连接");  
17. }
```

当连上网的时候 会自动执行 **onOnline** 方法

当断网的时候 会自动执行 **onOffline** 方法

**android** 问题:

自己动手检测后发现 : 如果没有网络 第一次启动的时候 **onOffline** 会调用 2 次

**ios** 问题:

官方文档说 启动需要 1 秒后才回有反应

**Demo** 演示图:



## Phonegap 网络连接 HTML5

Phonegap 网络连接判断 官网提供 Network API

cordova 3.4 版本的 API 判断网络的连接状态

判断网络是 4G 3G 2G WIFI none 等

本文在结尾处 提供 Demo 下载

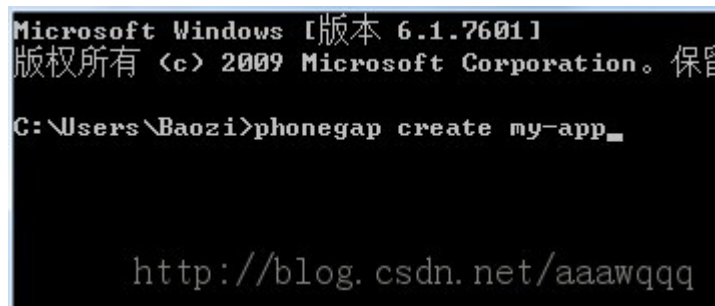
支持的系统:

- Amazon Fire OS
- Android
- BlackBerry 10
- iOS
- Windows Phone 7 and 8
- Tizen
- Windows 8

从建立项目说起:

<1> 在控制台 创建一个 **phonegap** 工程 命令如下

```
1. phonegap create my-app
2. cd my-app
3. phonegap run android
```



<2> 我们从命令行进入 到工程目录下的 **plugins** 文件夹

```
1. cd my-app
2.
3. cd plugins
```



```
C:\Develop\phonegap>phonegap create my-app
[phonegap] created project at C:\Develop\phonegap\my-app

C:\Develop\phonegap>cd my-app

C:\Develop\phonegap\my-app>cd plugins

C:\Develop\phonegap\my-app\plugins>

http://blog.csdn.net/aaawqqq
```

<3> 现在开始下载插件

1. cordova plugin add org.apache.cordova.network-information

下载完成:

```
C:\Develop\phonegap\cordova3.4_demo\Network\plugins>cordova plugin add org.apache.cordova.network-information
Fetching plugin "org.apache.cordova.network-information" via plugin registry

C:\Develop\phonegap\cordova3.4_demo\Network\plugins>blog.csdn.net/aaawqqq
```

<4> 添加 android 平台工程 (ios 把 "android" 替换)

1. cordova platform add android

添加完成:

```
C:\Develop\phonegap\cordova3.4_demo\Network\plugins>cordova platform add android

Creating android project...
Creating Cordova project for the Android platform:
  Path: platforms\android
  Package: com.demo.network
  Name: NetWork
  Android target: android-19
Copying template files...
Running: android update project --subprojects --path "platforms\android" --target android-19 --library "CordovaLib"
Resolved location of library project to: C:\Develop\phonegap\cordova3.4_demo\Network\platforms\android\CordovaLib
Updated and renamed default.properties to project.properties
Updated local.properties
No project name specified, using Activity name 'NetWork'.
If you wish to change it, edit the first line of build.xml.
Added file C:\Develop\phonegap\cordova3.4_demo\Network\platforms\android\build.xml
Added file C:\Develop\phonegap\cordova3.4_demo\Network\platforms\android\proguard-project.txt
Updated project.properties
Updated local.properties
No project name specified, using project folder name 'CordovaLib'.
If you wish to change it, edit the first line of build.xml.
Added file C:\Develop\phonegap\cordova3.4_demo\Network\platforms\android\CordovaLib\build.xml
Added file C:\Develop\phonegap\cordova3.4_demo\Network\platforms\android\CordovaLib\proguard-project.txt

Project successfully created.
Generating config.xml from defaults for platform "android"
Preparing android project
Starting installation of "org.apache.cordova.network-information" for android
Preparing android project
org.apache.cordova.network-information installed on android.
C:\Develop\phonegap\cordova3.4_demo\Network\plugins>
```

<http://blog.csdn.net/aaawqqq>

#### <5> 编译 android 工程

##### 1. cordova build

编译完成:

```
管理员: C:\Windows\system32\cmd.exe
rk\platforms\android\res\drawable\icon.png => C:\Develop\phonegap\cordova3.4_demo\Network\platforms\android\ant-build\res\drawable\icon.png
[crunch] <processed image to cache entry C:\Develop\phonegap\cordova3.4_demo\Network\platforms\android\ant-build\res\drawable\icon.png: 53% size of source>

[crunch] Crunched 5 PNG files to update cache

-package-resources:
  [aapt] Creating full resource package...

-package:
[apkbuilder] Current build type is different than previous build: forced apkbuilder run.
[apkbuilder] Creating Network-debug-unaligned.apk and signing it with a debug key...

-post-package:

-do-debug:
  [zipalign] Running zip align on final apk...
  [echo] Debug Package: C:\Develop\phonegap\cordova3.4_demo\Network\platforms\android\ant-build\Network-debug.apk
[propertyfile] Creating new property file: C:\Develop\phonegap\cordova3.4_demo\Network\platforms\android\ant-build\build.prop
[propertyfile] Updating property file: C:\Develop\phonegap\cordova3.4_demo\Network\platforms\android\ant-build\build.prop
[propertyfile] Updating property file: C:\Develop\phonegap\cordova3.4_demo\Network\platforms\android\ant-build\build.prop
[propertyfile] Updating property file: C:\Develop\phonegap\cordova3.4_demo\Network\platforms\android\ant-build\build.prop
[propertyfile] Updating property file: C:\Develop\phonegap\cordova3.4_demo\Network\platforms\android\ant-build\build.prop

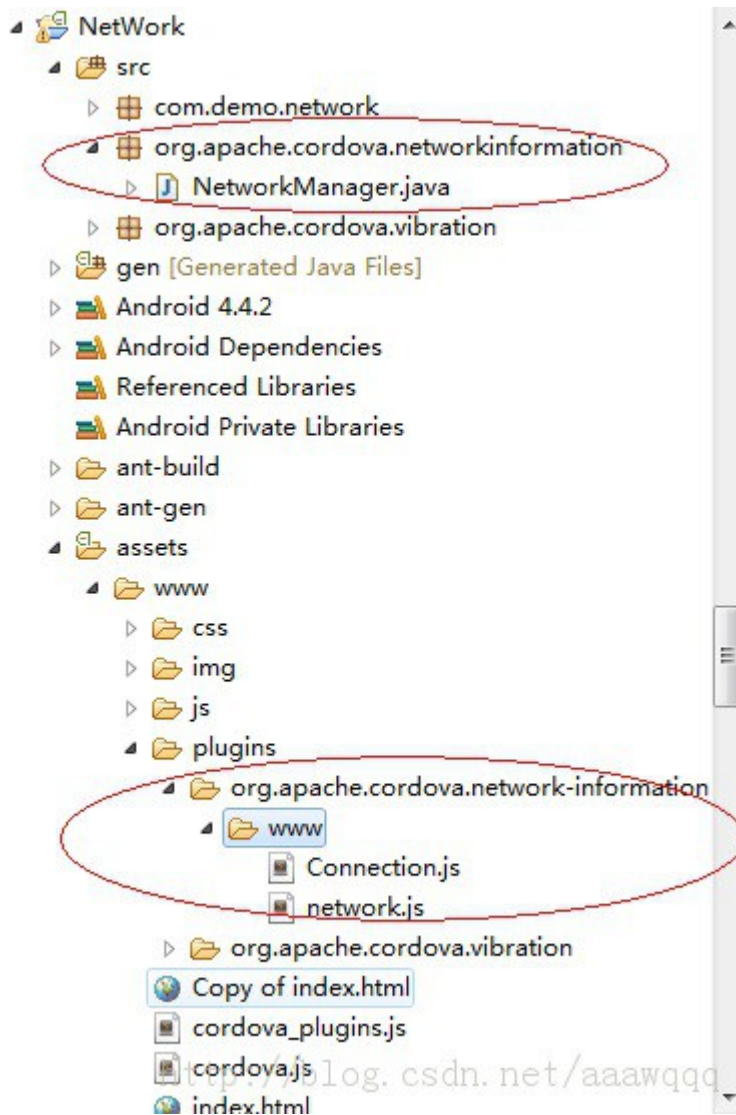
-post-build:
  [move] Moving 1 file to C:\Develop\phonegap\cordova3.4_demo\Network\platforms\android\ant-build
  [move] Moving 1 file to C:\Develop\phonegap\cordova3.4_demo\Network\platforms\android\CordovaLib\ant-build

debug:

BUILD SUCCESSFUL
Total time: 17 seconds

C:\Develop\phonegap\cordova3.4_demo\Network\plugins> http://blog.csdn.net/aaawqq
```

此时将工程导入到 eclipse 中



画圈圈的地方就是 系统生成的代码

将下方语句考到 **assets** 目录下 **www/index.html** 当中 完全复制过去;

example:

```
1. <!DOCTYPE html>
2.
3. <html>
4. <head>
5. <meta charset="utf-8" />
6. <meta name="format-detection" content="telephone=no" />
7. <!-- WARNING: for iOS 7, remove the width=device-width and height=device-
   height attributes. See https://issues.apache.org/jira/browse/CB-4323 -->
```

```
8. <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" />
9. <link rel="stylesheet" type="text/css" href="css/index.css" />
10. <title>Hello World</title>
11. </head>
12. <body>
13.
14. <p><a href="#" style="font-size:30px;" onclick="checkConnection(); return
    false;">包子的网络连接</a></p>
15.
16. <br />
17. <br />
18. <br />
19. <br />
20. <div style="font-size:30px;" onclick="aaa();">包子测试</div>
21.
22.
23. <div class="app">
24. <h1>Apache Cordova</h1>
25. <div id="deviceready" class="blink">
26. <p class="event listening">Connecting to Device</p>
27. <p class="event received">Device is Ready</p>
28. </div>
29. </div>
30. <script type="text/javascript" src="cordova.js"></script>
31. <script type="text/javascript" src="js/index.js"></script>
32. <script type="text/javascript">
33. app.initialize();
```

```
34. </script>
35.
36. <script type="text/javascript">
37.
38. function aaa(){
39. alert(111);
40. };
41.
42. //这里是判断网络连接状态的方法
43. function checkConnection() {
44. var networkState = navigator.connection.type;
45.
46. var states = {};
47. states[Connection.UNKNOWN] = 'Unknown connection';
48. states[Connection.ETHERNET] = 'Ethernet connection';
49. states[Connection.WIFI] = 'WiFi connection';
50. states[Connection.CELL_2G] = 'Cell 2G connection';
51. states[Connection.CELL_3G] = 'Cell 3G connection';
52. states[Connection.CELL_4G] = 'Cell 4G connection';
53. states[Connection.CELL] = 'Cell generic connection';
54. states[Connection.NONE] = 'No network connection';
55.
56. alert('Connection type: ' + states[networkState]);
57. }
58.
59.
60. </script>
61.
```

```
62. </body>
```

```
63. </html>
```

代码详解:

```
1. var networkState = navigator.connection.type;
2.
3. var states = {};
4. states[Connection.UNKNOWN] = 'Unknown connection';
5. states[Connection.ETHERNET] = 'Ethernet connection';
6. states[Connection.WIFI] = 'WiFi connection';
7. states[Connection.CELL_2G] = 'Cell 2G connection';
8. states[Connection.CELL_3G] = 'Cell 3G connection';
9. states[Connection.CELL_4G] = 'Cell 4G connection';
10. states[Connection.CELL] = 'Cell generic connection';
11. states[Connection.NONE] = 'No network connection';
```

`navigator.connection.type` 获取当前网络状态

`states` 数组表示输出的状态 大家可以根据需要自己修改输出的参数



## Phonegap Device 获取设备信息



phonegap 获取手机设备信息

获取的手机的有

**device.model** :返回设备的模型或产品的名称

**device.cordova** :返回 cordova 的版本

**device.uuid** :返回手机 uuid

**device.version** :返回系统版本

**device.platform** :返回手机的平台信息 (android/ios 等等)

使用的版本是 cordova 3.3 ,能兼容 cordova 3.4 ;

以上是系统提供的功能 ;

获取 imei 的办法将在下一篇介绍

老规矩从头创建工程开始:

1 使用命令行 建立 phonegap 工程

2 进入到 phonegap 工程下的 plugins 文件夹当中 ,使用命令行下载官方插件 ;(输入命令:  
cordova plugin add org.apache.cordova.device)

4 添加生成 android 或者 IOS 工程 ;android 命令如下 cordova platform add android  
(注:oc 的把 android 换成 ios)

5 在命令行输入 cordova build

6 最后将生成的工程导入编辑器(我用的 eclipse)

在 assets/www 目录下在 index.html 中添加 javascript 调用语句

<1> 在控制台 创建一个 phonegap 工程 命令如下

```
1. phonegap create my-app
2. cd my-app
3. phonegap run android
```

```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留
C:\Users\Baozi>phonegap create my-app_

http://blog.csdn.net/aaawqqq
```

<2> 我们从命令行进入 到工程目录下的 plugins 文件夹

1. cd my-app
- 2.
3. cd plugins

```
C:\Develop\phonegap>phonegap create my-app
[phonegap] created project at C:\Develop\phonegap\my-app
C:\Develop\phonegap>cd my-app
C:\Develop\phonegap\my-app>cd plugins
C:\Develop\phonegap\my-app\plugins>

http://blog.csdn.net/aaawqqq
```

<3> 现在开始下载插件

1. cordova plugin add org.apache.cordova.device

```
C:\Users\Baozi>cd C:\Develop\phonegap\Constacts\plugins
C:\Develop\phonegap\Constacts\plugins>cordova plugin add org.apache.cordova.device
Fetching plugin "org.apache.cordova.device" via plugin registry
Starting installation of "org.apache.cordova.device" for android
Preparing android project
org.apache.cordova.device installed on android
http://blog.csdn.net/aaawqqq
```

<4> 添加 android 平台工程 (ios 把 "android" 替换)

1. cordova platform add android

```
C:\Develop\phonegap\my-app\plugins>cordova platform add android
Creating android project...
Preparing android project
Starting installation of "org.apache.cordova.camera" for android
Preparing android project
org.apache.cordova.camera installed on android.
http://blog.csdn.net/aaawqqq
C:\Develop\phonegap\my-app\plugins>
```

#### <5> 编译 android 工程

1. cordova build

```
C:\Develop\phonegap\my-app\plugins>cordova build
Generating config.xml from defaults for platform "android"
Preparing android project
Compiling app on platform "android" via command "cmd" /c C:\Develop\phonegap\my-app\platforms\android\cordova\build
Platform "android" compiled successfully.
http://blog.csdn.net/aaawqqq
C:\Develop\phonegap\my-app\plugins>
```

至此 devices 已经生成...

现在大家只需要将工程导入到 eclipse 当中 使用官方 API 语句在 javascript 中调用

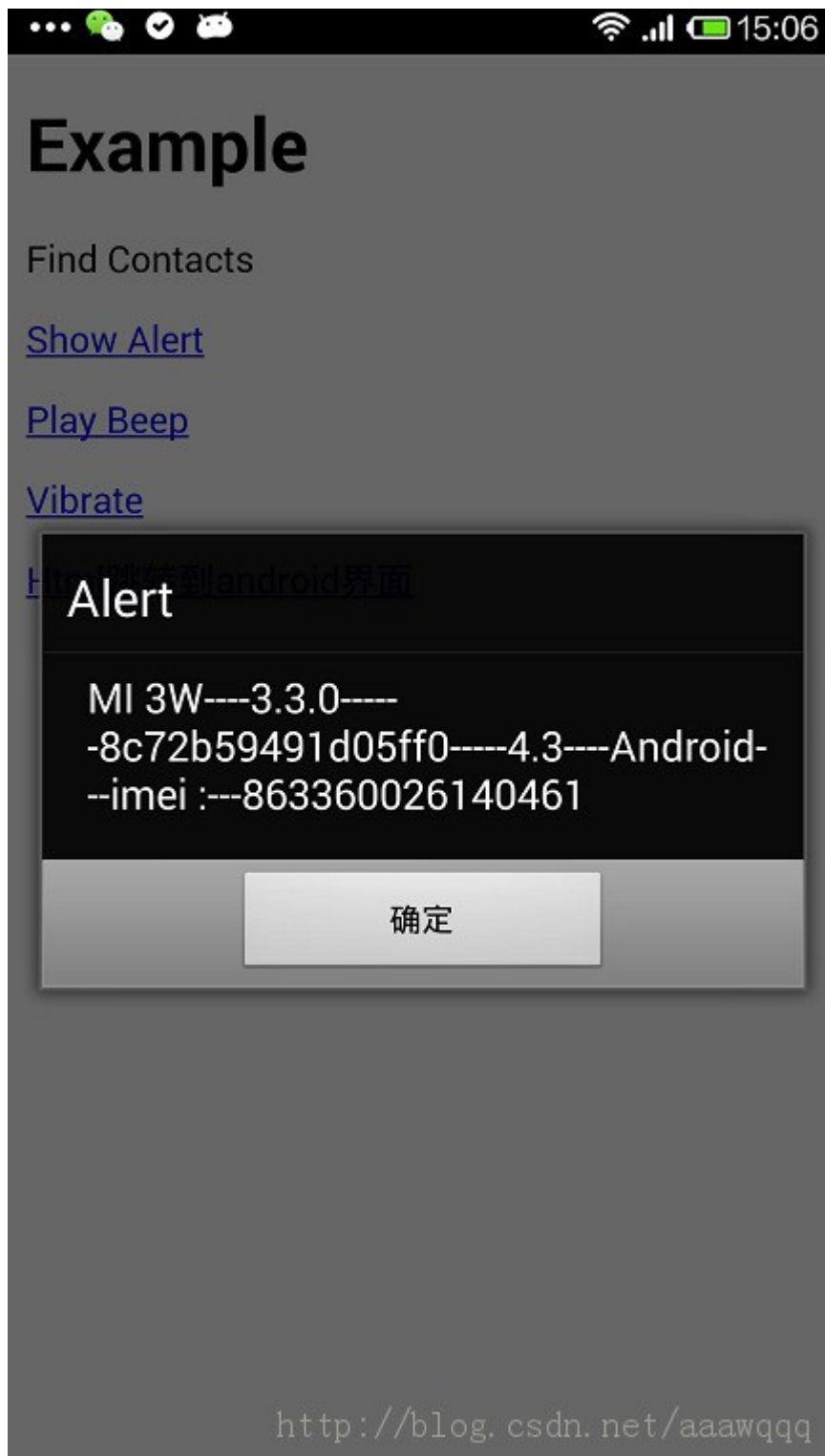
将下方语句考到 assets 目录下 www/index.html 当中 完全复制过去;

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5. <title>Contact Example</title>
6. <script type="text/javascript" charset="utf-8" src="cordova.js"></script>
7. <script type="text/javascript" charset="utf-8">
8. // Wait for device API libraries to load
9. // document.addEventListener("deviceready", onDeviceReady, false);
10. // device APIs are available
11.
12. function onDeviceReady() {
13.
14. // Wait for device API libraries to load //
```

```
15.
16. //var element = document.getElementById('deviceProperties');
17. //alert(element);
18.
19. // element.innerHTML =
20. // 'Device Model: ' + device.model + '<br />' +
21. // 'Device Cordova: ' + device.cordova + '<br />' +
22. // 'Device Platform: ' + device.platform + '<br />' +
23. // 'Device UUID: ' + device.uuid + '<br />' +
24. // 'Device Version: ' + device.version + '<br />';
25.
26.
27. alert( device.model + "----"+device.cordova + "-----"+ device.uuid
    + "-----"+device.version+"----"+device.platform );
28. }
29.
30. function baozi(){ alert("S1");}
31.
32. function intent() { onDeviceReady(); }
33.
34.
35. </script>
36. </head>
37. <body>
38. <h1>Example</h1>
39. <p>Find Contacts</p>
40.
41. <p><a href="#" onclick="baozi(); return false;">Vibrate</a></p>
```

```
42. <p><a href="#" onclick="intent(); return false;">Html 跳转到 android 界面<
    /a></p>
43. </body>
44. </html>
```

点击 " Html 跳转到 android 界面 " 就会弹出 alert 如下:



依次

device.model :返回设备的模型或产品的名称

device.cordova :返回 cordova 的版本

device.uuid :返回手机 uuid

device.version :返回系统版本

`device.platform` :返回手机的平台信息 (android/ios 等等)

大家在 javascript 代码中 使用以上调用语句 就能获取参数了

最后一个参数是关于 imei 的获取 需要修改 phonegap 官方接口 将在下一章介绍 ;  
这里只介绍官方的 API

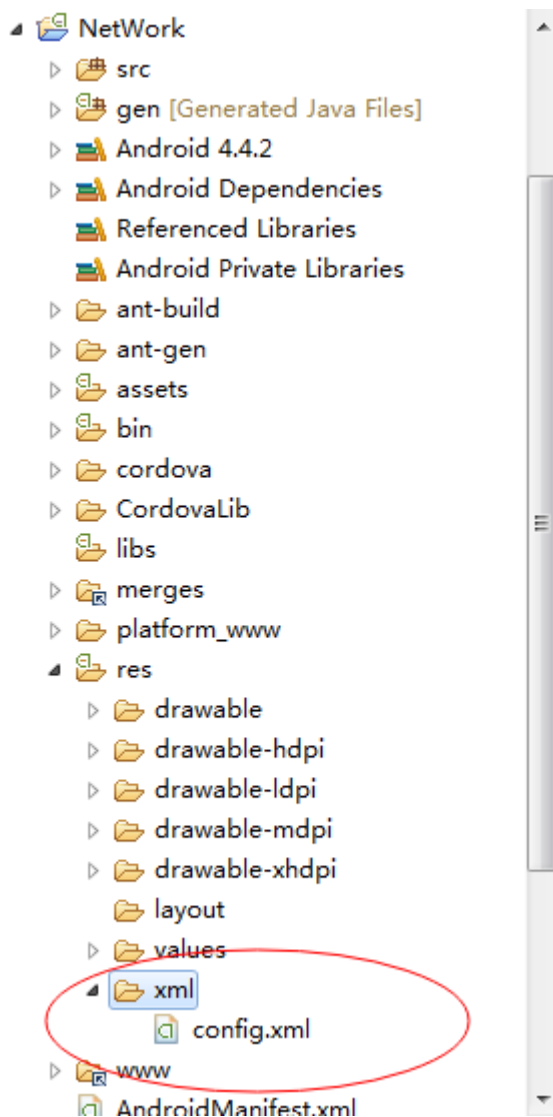
## Phonegap 修改全屏/非全屏设置

Phonegap 全屏/非全屏设置

cordova 3.4 修改屏幕配置 修改参数来设置 WebView 全屏 或者 非全屏

在工程的 `res/xml` 目录下的 `config.xml`

如下图:



在 config.xml 中找到:

```
1. <preference name="Fullscreen" value="true" />
```

将 value 参数配置为 true 就是 全屏

将 value 参数配置为 false 就是 非全屏



Phonegap 解决错误:Error initializing Cordova:Class not found

Phonegap 解决错误: Alert      [ERROR]Error initializing Cordova:Class not found



发现 bug 后找原因 网上说是 因为找不到 `res/xml` 文件夹下的 `config.xml` 文件;

但我有这个 `config.xml`

我想了一下 是不是因为没有找到指定的 `config.xml` 中的 `feature` 配置

最后找到了 是因为我调用系统的 `Device` 信息 没有加标签

```
1. <!-- 设备 -->
```

```
2. <feature name="Device" >
3. <param
4. name="android-package"
5. value="org.apache.cordova.device.Device" />
6. </feature>
```

出现这个 bug 的原因是调用 phonegap 的插件 没有找到相对应的 java 类

解决思路:

- 1 检查 res/xml 目录下有无 config.xml 文件
- 2 检查 config.xml 文件中 feature 标签 是否完善

## Phonegap 中文乱码解决方法

phonegap 传递中文汉字 出现乱码的情况 ;

解决办法:

在 html 中 head 里面, 添加如下代码即可

```
1. <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

整个工程统一编码 用 utf-8



官方地址:[http://cordova.apache.org/docs/en/3.4.0/guide\\_cli\\_index.md.html#The%20Command-Line%20Interface](http://cordova.apache.org/docs/en/3.4.0/guide_cli_index.md.html#The%20Command-Line%20Interface)

官方的升级修改:<http://cordova.apache.org/#news>

phonegap 命令:[http://cordova.apache.org/docs/en/3.4.0/guide\\_cli\\_index.md.html#The%20Command-Line%20Interface](http://cordova.apache.org/docs/en/3.4.0/guide_cli_index.md.html#The%20Command-Line%20Interface)

## phonegap 导入工程报错解决办法 错误解析 android

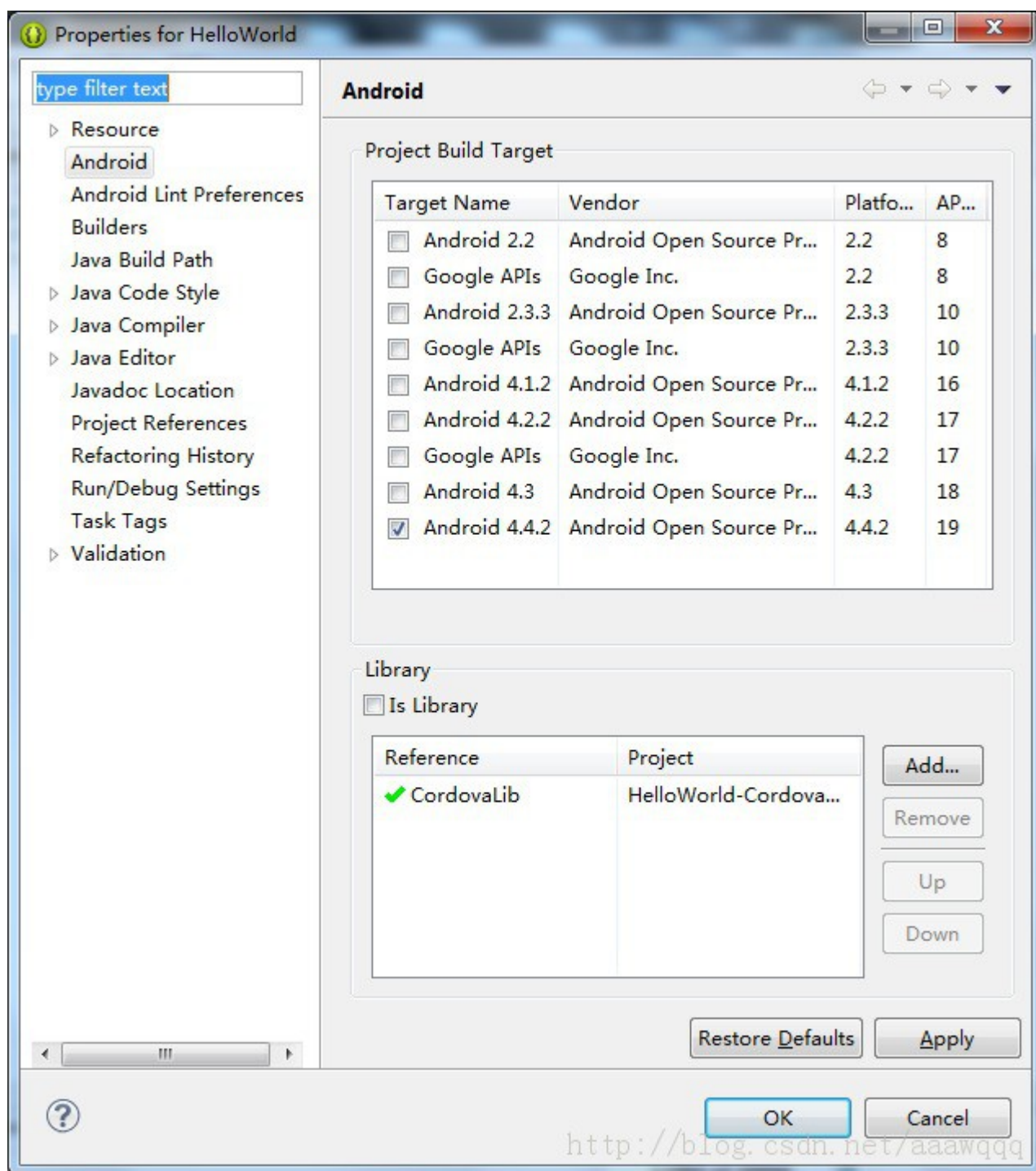
phonegap 导入工程到 eclipse 报错 无非三种情况

- 1 没有导系统 lib 库 或者 错误
- 2 没有选择 android 最新的 API (非 google APIs )
- 3 没有改成 UTF-8 格式

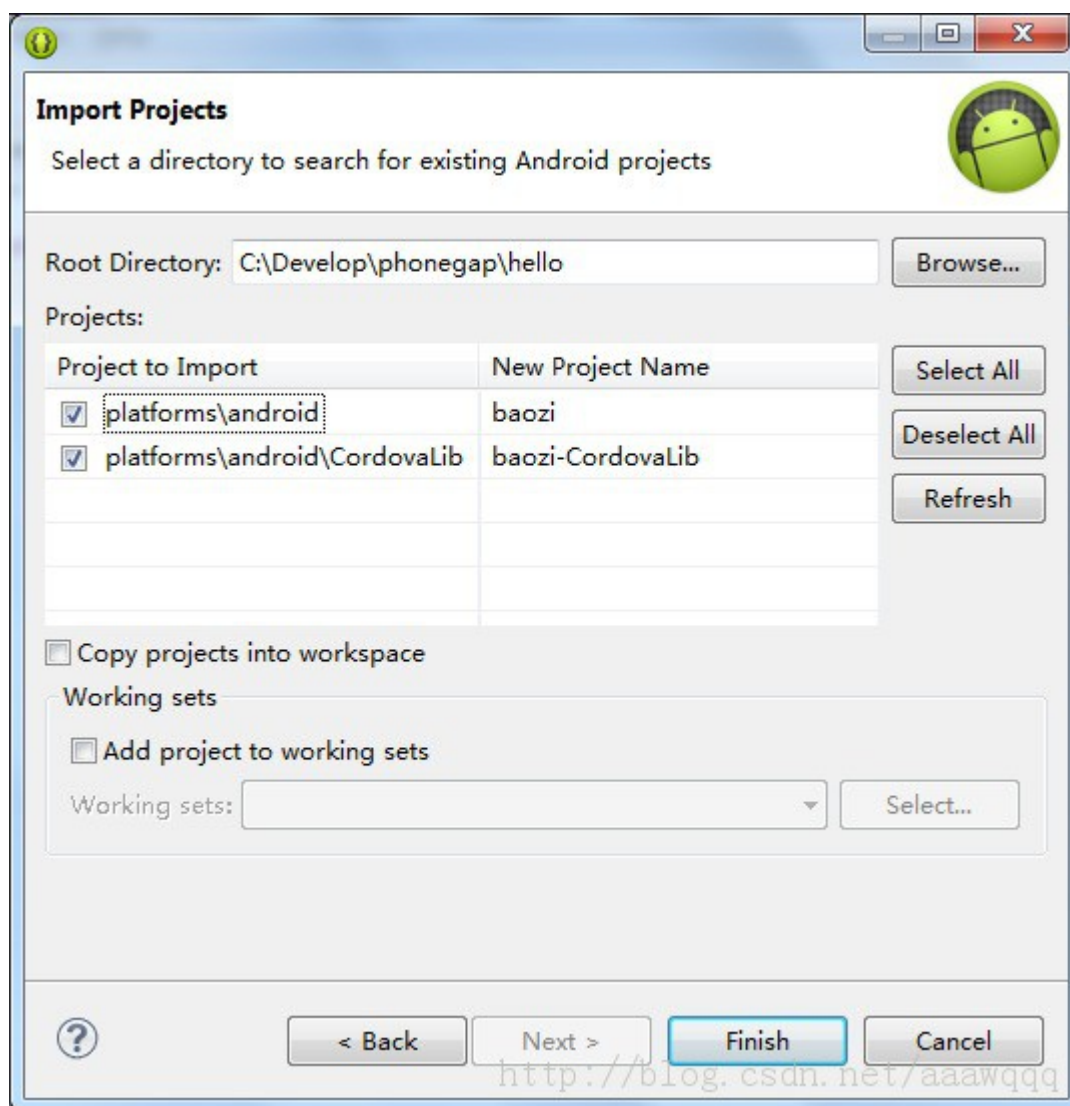
最后 clean 一下就好

-----解决办法-----

先看正确的工程 直接上图



导入工程的时候 是 2 份 工程的下方是 phonegap 的第三方库





# 附件

作者:包晗

联系方式 :

blog: <http://blog.csdn.net/aaawqqq/article/category/1935471>

QQ: 2221673069

邮箱: [baohan@foreveross.com](mailto:baohan@foreveross.com) 或者 2221673069@qq.com