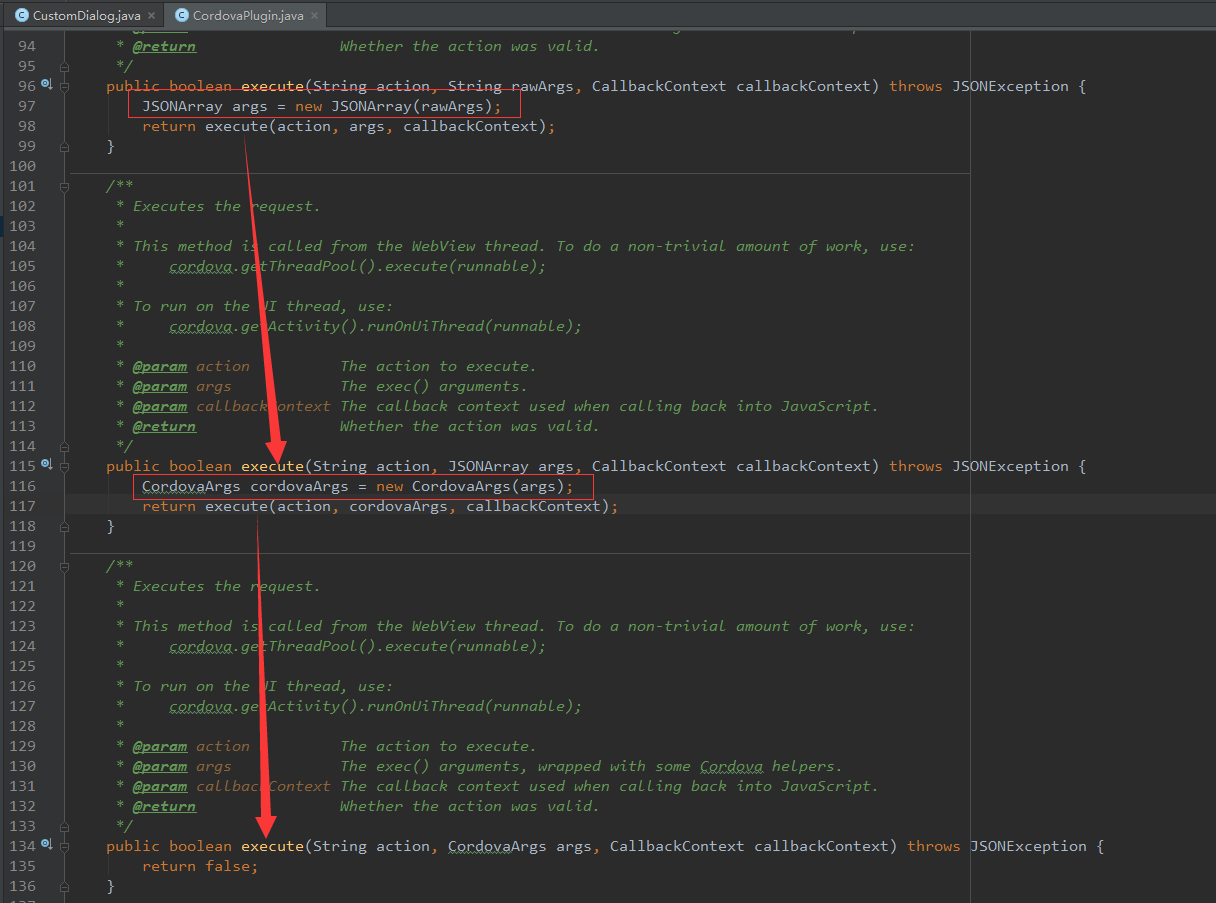
**编写自定义插件类**

本节的内容是，自定义一个dialog插件，供web调用，显示系统弹窗。   
新建一个包名，我这里使用org.apache.cordova.dialog，然后创建个类CustomDialog.java，继承于CordovaPlugin（所有自定义插件，都要继承CordovaPlugin），最后重写execute方法。   
execute有三个重载方法：

这三个方法，如果你看了CordovaPlugin源码的话，会发现，其实最后都调用了第二个方法，但是CordovaArgs只是对JSONArray的一个封装，方便操作json数据而已，所以要重写哪个，按个人喜好。



这里，我是重写了第二个方法，现在来说明下方法参数：   
String action：一个类里面可以提供多个功能，action就是指名了要调用哪个功能。   
CordovaArgs args：web以json的数据格式传递给Android native，CordovaArgs 是对JSONArray 的一个封装。  
CallbackContext callbackContext：这个是回调给web，有success和error两种回调方法。

具体实现如下：



如果web使用了CustomDialog插件，并调用show方法（action）。这时候，会弹出一个系统窗口，会显示web传过来的消息内容，点击确定，回调web，告诉它调用成功，取消则是失败。**最后记得return true（表示调用成功）。**

**配置config.xml**

打开res/xml/config.xml文件，原本内容如下：



Whitelist是基础项目自带的一个插件，同样的，我们在widget节点下也添加一个feature。

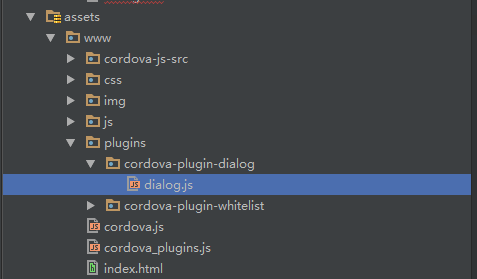
feature name可以随便取，param value填的是具体类的位置。

**web配置并调用**

**1、配置**

打开asserts/www/cordova\_plugins.js文件,注册插件。

然后在assets/www下创建dialog.js文件



内容如下：

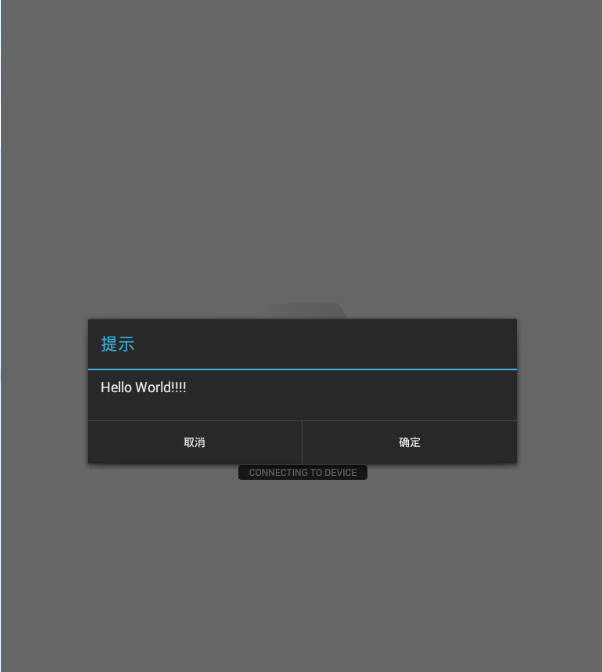
cordova.define 的第一个参数就是cordova\_plugins.js里面定义的id，最主要的是exec方法，参数说明：   
参数1：成功回调function   
参数2：失败回调function   
参数3：feature name，与config.xml中注册的一致   
参数4：调用java类时的action   
参数5：要传递的参数，json格式

**2、调用**

配置完上面的，只剩下在合适的地方调用它了。   
打开assets/www/js/index.js   
找到onDeviceReady方法，然后调用

这里直接调用alertDialog，就是在cordova\_plugins.js下定义的clobbers名称,show是在js申明的function   
index.js#onDeviceReady

直接在Android Studio运行，效果图如下：



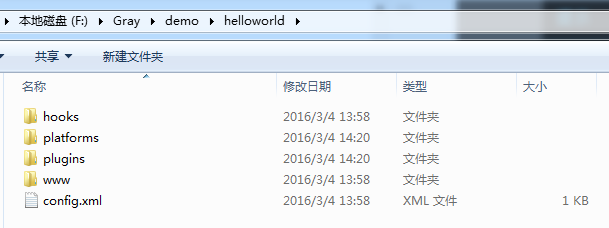
这个对话框就是Android 系统的AlertDialog。当我们点击确定时，就会回调js success function，打印出日志。

http://img.blog.csdn.net/20160307134255732

**不要执行cordova build命令！！！**

你会有这么个疑问，编译项目不是用cordova build命令的吗？很好，网上很多说，在项目中，代码都已经写好了，但是一执行该命令，不但没有生效，反而之前写的插件代码都没了。其实，如果你知道整个项目编译过程，那么问题就迎刃而解了。

这里简单的说下，看下图的目录结构



plugins是存放插件的文件夹，而www是存放H5项目的，也就是Android项目下的asserts/www下的文件，当执行cordova build时，它会根据config.xml配置来编译项目，然后会将plugins和www下的文件应用到platforms文件夹下的各个平台，这就是“一次编译，多平台运行”。

但是，至始至终，我们编写的代码都是在Android平台下的，而上面提到的两个文件夹我们都没有动过，也就是说，所有配置都没有动过，都是初始状态，所以执行build后也就没有效果。

**总结**

通过上面的实例，已经实现了简单的自定义插件。