cordova插件开发

1. 环境安装
2. Nodejs安装

直接去官网下载安装

1. cordova安装

在docs窗口下

npm install -g cordova

1. 创建一个cordova项目

docs下切换到你要创建项目的目录

cordova create <project path> <pages> <project name>

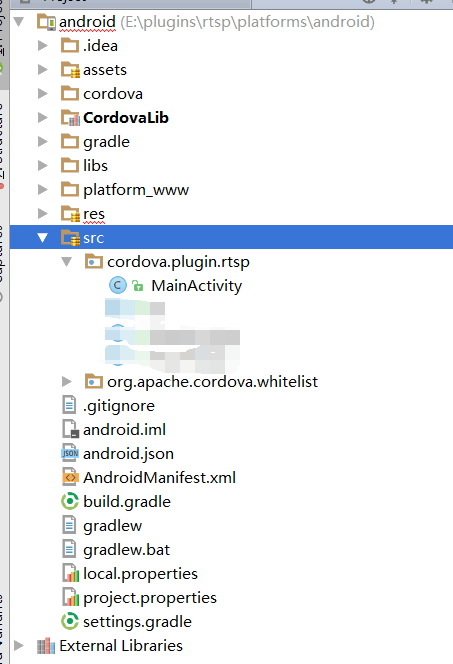
cordova create rtsp cordova.plugin.rtsp rtsp

1. 添加平台

cordova platform add android/ios

1. 编写我们要调用的android代码

用android studio打开项目



初始化的时候项目结构如下，有一个MainActivity这个类知识用来测试的，后期我们打包不会用到

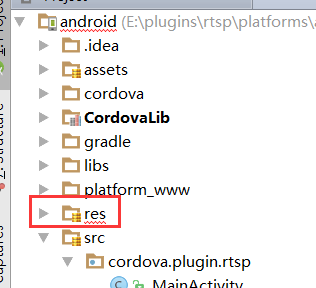
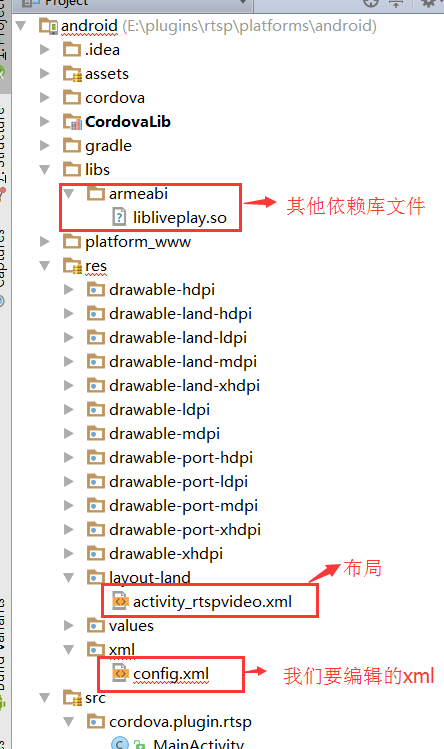
然后编写我们的插件类RtspPlugin.java

**public class** RtspPlugin **extends** CordovaPlugin {  
 **private** CallbackContext **callbackContext**;  
 **public static final** String ***PLAY\_ACTION*** = **"play"**;  
 **public static final** String ***STOP\_ACTION*** = **"stop"**;  
  
 @Override  
 **public boolean** execute(String action, JSONArray args, CallbackContext callbackContext) **throws** JSONException {  
 **try** {  
 **this**.**callbackContext** = callbackContext;  
 *//根据action判断调用方法* **if** (***PLAY\_ACTION***.equals(action)) {  
 String rtspUrl = args.getString(0);  *//通过Intent绑定将要调用的Activity* Intent intent = **new** Intent().setClass(**cordova**.getActivity(), RtspActivity.**class**);  
 *//加入将要传输到activity中的参数* intent.putExtra(**"rtspUrl"**, rtspUrl);  
 *//启动activity* **this**.**cordova**.startActivityForResult(**this**, intent, 0);}  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 **return false**;  
 }  
 **return true**;  
 }  
  
 *//onActivityResult为第二个Activity执行完后的回调接收方法* @Override  
 **public void** onActivityResult(**int** requestCode, **int** resultCode, Intent intent){  
 *// 根据resultCode判断处理结果* **switch** (resultCode) {  
 **case** Activity.***RESULT\_OK***:  
 String message = intent.getStringExtra(**"message"**);  
 *//设置回调success函数* **callbackContext**.success(message);  
 **break**;  
 **default**:  
 **break**;  
 }  
 }  
}

编写要调用的RtspActivity.java

**public class** RtspActivity **extends** Activity {  
 **public** String **rtspUrl**;  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_rtspvideo***);  
  
 *//获取rtspUrl* **rtspUrl** = **this**.getIntent().getStringExtra(**"rtspUrl"**);  
 VideoView videoView = (VideoView)findViewById(R.id.***rtdp\_video***);  
 videoView.setVideoURI(Uri.*parse*(**rtspUrl**));  
 videoView.requestFocus();  
 videoView.start();  
  
 ImageView videoBack = (ImageView) findViewById(R.id.***video\_back***);  
 videoBack.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 String message = **rtspUrl**;  
 Intent intent = **new** Intent();  
 intent.putExtra(**"message"**, message);  
 setResult(Activity.***RESULT\_OK***, intent);  
 *//关闭该activity，把返回值传回到cordovaPlugin插件* finish();  
 }  
 });  
 }  
}

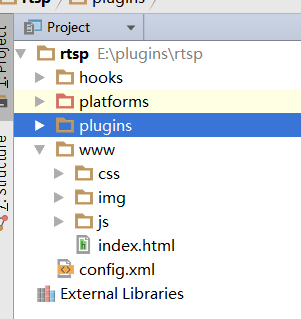
到这里我们的核心代码就写完了，去配置res/xml/config.xml

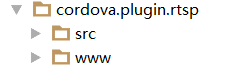
*<?***xml version='1.0' encoding='utf-8'***?>*<**widget id="com.cordova.plugin.rtsp" version="1.0.0" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/ns/1.0"**>  
*<!-- 该APP名称 -->*<**name**>rtsp</**name**>  
*<!-- APP描述 -->*<**description**>  
 this is a support for rtsp stream video plugin  
</**description**>  
*<!-- 作者信息描述 -->*<**author email="345833303@qq.com"**>  
 XieKong  
</**author**>  
*<!-- 默认启动页面 -->*<**content src="index.html"** />  
*<!-- 指定app可进行通信的域名，\*为所有 -->*<**access origin="\*"** />  
*<!-- App默认只请允许通过手机端直接打开，若想通过网站，SMS等方式调用APP，则需要设置allow-intent配置 -->*<**allow-intent href="http://\*/\*"** />  
<**allow-intent href="https://\*/\*"** />  
<**allow-intent href="tel:\*"** />  
<**allow-intent href="sms:\*"** />  
<**allow-intent href="mailto:\*"** />  
<**allow-intent href="geo:\*"** />  
<**allow-intent href="market:\*"** />  
*<!--设置运行环境中的参数值 -->*<**preference name="loglevel" value="DEBUG"** />  
*<!-- 插件描述 -->*<**feature name="Whitelist"**>  
 <**param name="android-package" value="org.apache.cordova.whitelist.WhitelistPlugin"** />  
 <**param name="onload" value="true"** />  
</**feature**>  
<**feature name="RtspPlgin"**>  
 <**param name="android-package" value="cordova.plugin.rtsp.RtspPlgin"** />  
</**feature**>  
</**widget**>

1. 创建插件

用WebStorm打开项目



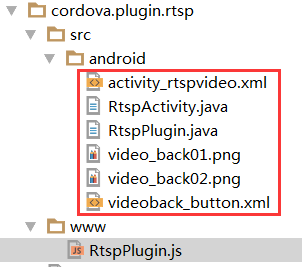
在plugins下面创建包cordova.plugin.rtsp，然后依次在此目录下创建src和www目录



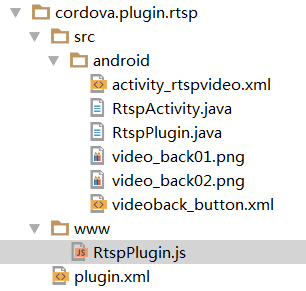
然后在src下面创建andorid目录，然后在www目录下面创建名为RtspPlugin.js

*//****TODO 导入依赖库*var** exec = require(**'cordova/exec'**);  
**var** platform = require(**'cordova/platform'**);  
  
module.**exports** = {  
 *//* ***TODO JS 中调用的 js方法，参数列表可根据业务需求定*** play: **function** (args, success, error) {  
 *//****TODO 参数（回调方法,null,类名，方法名，[参数1，参数2，……]）*** exec(success, error, **"RtspPlugin"**, **"play"**, [args]);  
 },  
 stop: **function** (args, success, error) {  
 *//****TODO 参数（回调方法,null,类名，方法名，[参数1，参数2，……]）*** exec(success, error, **"RtspPlugin"**, **"stop"**, [args]);  
 }  
}

然后把我们的java类和用到的资源全部拷贝到src/android下

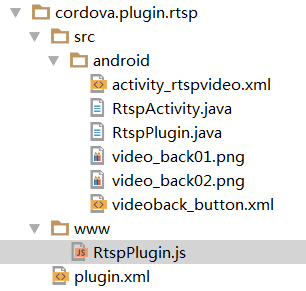


最后创建plugin.xml



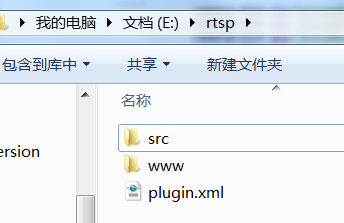
*<?***xml version="1.0" encoding="UTF-8"***?>*<**plugin xmlns="http://apache.org/cordova/ns/plugins/1.0"  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 id="cordova.plugin.rtsp"  
 version="1.0.0"**>  
 *<!--****TODO 域名空间，插件ID,版本号 cordova 将根据 ID生成 plugins 下的目录结构*** *-->  
 <!--****TODO 插件名****-->* <**name**>rtsp</**name**>  
 *<!--****TODO 插件描述****-->* <**description**>  
 this is a support for rtsp stream video plugin  
 </**description**>  
 *<!--****TODO 作者****-->* <**author**>XieKong</**author**>  
 *<!--****TODO 关键字****-->* <**keywords**>rtsp</**keywords**>  
 *<!--****TODO 许可协议****-->* <**license**>Apache 2.0</**license**>  
 <**engines**>  
 *<!--****TODO 支持的引擎及版本号****-->* <**engine name="cordova" version=">=4.0"**/>  
 </**engines**>  
 *<!--* ***TODO android 平台的配置****-->* <**platform name="android"**>  
 <**js-module src="www/RtspPlugin.js" name="RtspPlugin"**>  
 *<!--****TODO 这里是在JS中调该插件的前缀****-->* <**merges target="rtsp"** />  
 </**js-module**>  
 <**config-file target="res/xml/config.xml" parent="/\*"**>  
 <**feature name="RtspPlugin"**>  
 *<!--****TODO 插件的完整类路径****-->* <**param name="android-package" value="cordova.plugin.rtsp.RtspPlugin"**/>  
 </**feature**>  
 </**config-file**>  
 *<!--****TODO Java 本地代码映射，由两部分组成，前面是在插件文件中的路径（源目录），后面是目标目录，就是要把这些文件放到哪个目录下****-->* <**source-file src="src/android/RtspPlugin.java" target-dir="src/cordova/plugin/rtsp"**/>  
 <**source-file src="src/android/RtspActivity.java" target-dir="src/cordova/plugin/rtsp"**/>  
 <**source-file src="src/android/video\_back01.png" target-dir="res/drawable-hdpi"**/>  
 <**source-file src="src/android/video\_back02.png" target-dir="res/drawable-hdpi"**/>  
 <**source-file src="src/android/videoback\_button.xml" target-dir="res/drawable-hdpi"**/>  
 <**source-file src="src/android/activity\_rtspvideo.xml" target-dir="res/layout-land"**/>  
 </**platform**>  
</**plugin**>

至此我们的插件就开发问了，随后可以发布到github，插件包含三部分，src：java代码或者资源文件，www：js调用代码，plugin.xml：插件配置信息



1. 发布插件

把src，www，plugin.xml拷贝到另外单独的一个目录下面，我拷贝到了rtsp目录下面



切换到我们要用此插件的项目目录下面执行

cordova plugin add E:\rtsp（或者github远程仓库）

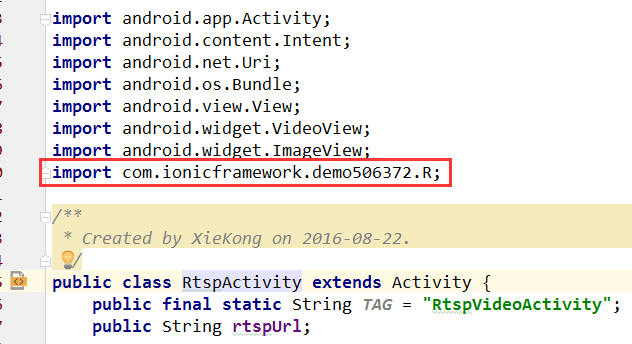
1. 修改以及使用

如果使用过布局文件的activity需要修改R资源导入，还需要把此activity在AndroidManifest.xml里面声明

<**activity  
 android:name="cordova.plugin.rtsp.RtspActivity"  
 android:screenOrientation="landscape"**/>

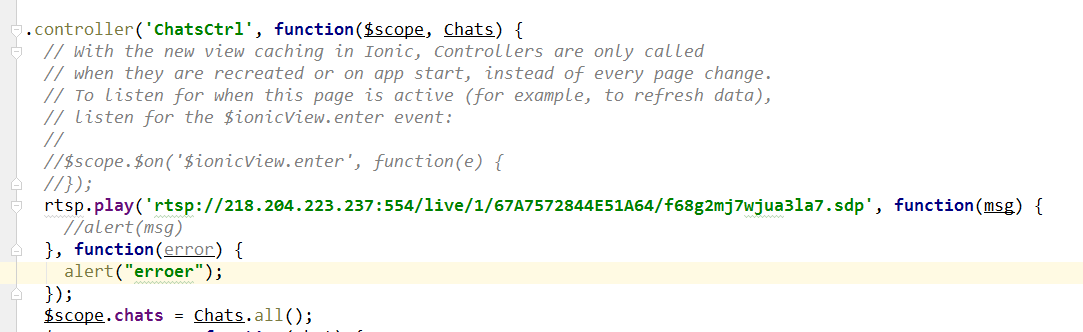
打开此activity导入AndroidManifest.xml里面**package="com.ionicframework.demo506372"**

**import com.ionicframework.demo506372.R**



1. 使用插件

在任意一个controller里面写入我这只做测试



然后docs切换到此项目 就可以去ionic build 和run了