SparkleComm UniApp SDK开发指南

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修订日期** | **修订内容** | **说明** |
| V1.0 | 2019.08.22 | 新增 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[SparkleComm UniApp SDK开发指南 1](#_Toc5342)

[1 初始化环境 3](#_Toc26999)

[2 初始化SDK 6](#_Toc32203)

[3 注销 7](#_Toc10018)

[4 拨打电话 7](#_Toc25112)

[5 来电接听 8](#_Toc3702)

[6 呼叫转移 9](#_Toc11020)

# 初始化环境

本文档是SparkleComm SDK 在uni-app框架下进行音视频通话的使用教程，请先按照如下方式配置开发环境，同时我们也提供了项目的Demo源码。

1. 使用HBuilderX创建uni-app项目，可以参考官网http://dcloud.io/。
2. 把“SparkleComm\_Plugin”这个插件拷贝到新建的uni-app项目的nativeplugins目录下（没有这个目录的话就自己创建）。“SparkleComm\_Plugin”插件在我们的Demo项目的nativeplugins目录里面。
3. 配置uni-app的插件，在HBuilderX工具里面打开项目的manifest.json文件，选择“App原生插件配置——选择本地插件”，在弹出框里选择“SparkleComm”，如下图：



1. 配置权限，在HBuilderX工具里面打开项目的manifest.json文件，选择“App模块权限配置”，在“Android打包权限配置”里面勾选如下权限：

"<uses-feature android:name=\"android.hardware.camera\"/>",

"<uses-feature android:name=\"android.hardware.camera.autofocus\"/>",

"<uses-permission android:name=\"android.permission.ACCESS\_COARSE\_LOCATION\"/>",

"<uses-permission android:name=\"android.permission.ACCESS\_FINE\_LOCATION\"/>",

"<uses-permission android:name=\"android.permission.ACCESS\_NETWORK\_STATE\"/>",

"<uses-permission android:name=\"android.permission.ACCESS\_WIFI\_STATE\"/>",

"<uses-permission android:name=\"android.permission.CALL\_PHONE\"/>",

"<uses-permission android:name=\"android.permission.CAMERA\"/>",

"<uses-permission android:name=\"android.permission.CAPTURE\_AUDIO\_OUTPUT\"/>",

"<uses-permission android:name=\"android.permission.CAPTURE\_SECURE\_VIDEO\_OUTPUT\"/>",

"<uses-permission android:name=\"android.permission.CAPTURE\_VIDEO\_OUTPUT\"/>",

"<uses-permission android:name=\"android.permission.CHANGE\_NETWORK\_STATE\"/>",

"<uses-permission android:name=\"android.permission.CHANGE\_WIFI\_MULTICAST\_STATE\"/>",

"<uses-permission android:name=\"android.permission.CHANGE\_WIFI\_STATE\"/>",

"<uses-permission android:name=\"android.permission.INTERNET\"/>",

"<uses-permission android:name=\"android.permission.LOCATION\_HARDWARE\"/>",

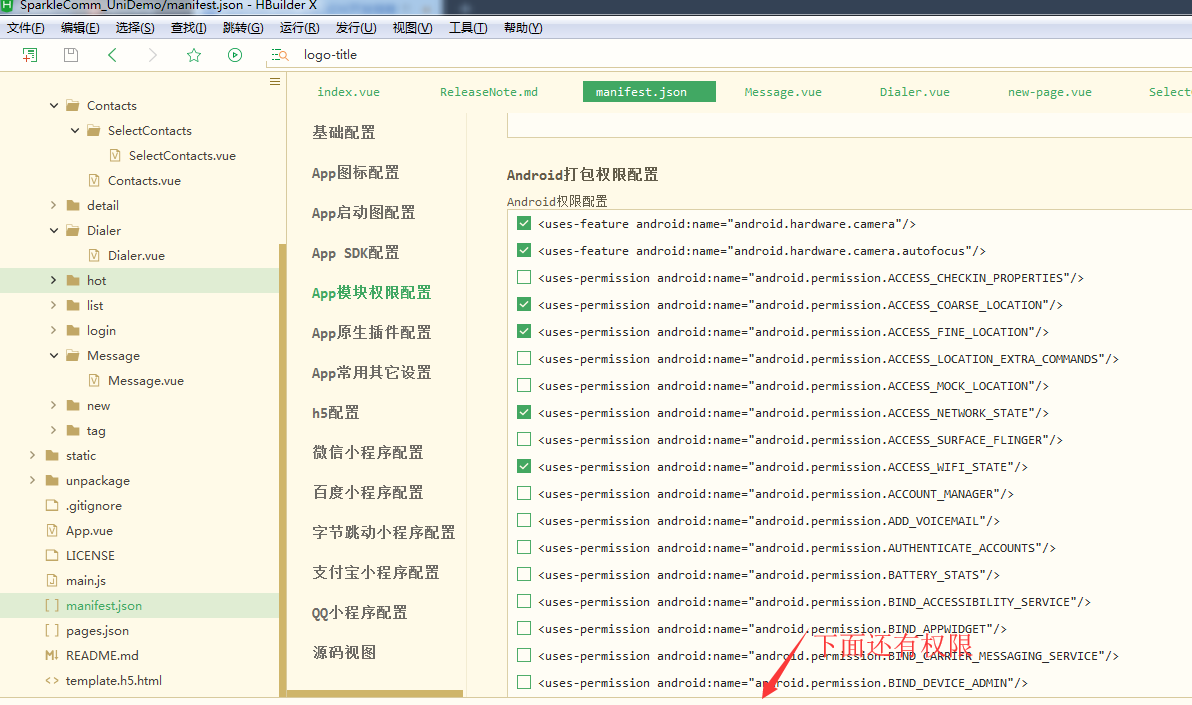
"<uses-permission android:name=\"android.permission.MODIFY\_AUDIO\_SETTINGS\"/>",

"<uses-permission android:name=\"android.permission.READ\_CONTACTS\"/>",

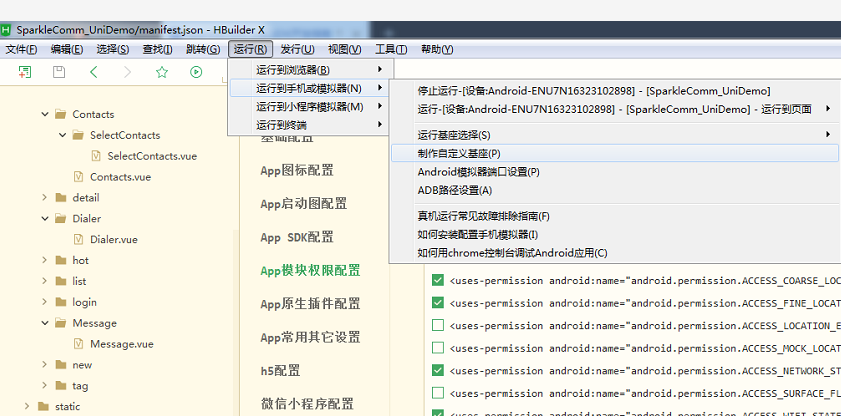
"<uses-permission android:name=\"android.permission.RECORD\_AUDIO\"/>",

"<uses-permission android:name=\"android.permission.WRITE\_CONTACTS\"/>",

"<uses-permission android:name=\"android.permission.WRITE\_EXTERNAL\_STORAGE\"/>"



1. 制作SDK调试基座，当制作了自定义基座后，就可以不用在线打包调试SDK了，制作方法为，点击菜单“运行——运行到手机或模拟器——制作自定义基座”，如下图。

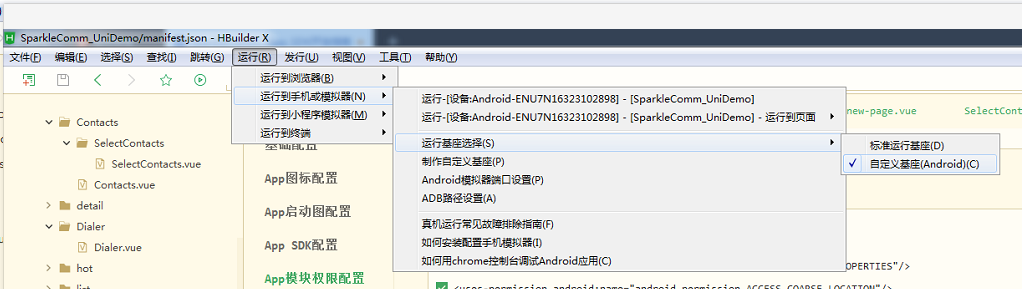




只选择Android，目前不支持iOS，Android包名修改为自己的包名，然后点击“打包”。等待云端打包完成，完成后会显示如下信息：



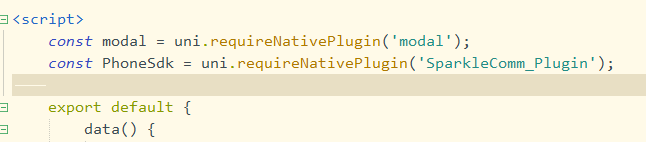
然后点击菜单“运行——运行到手机或模拟器——运行基座选择——自定义基座(Android)”，如下图，



经过以上步骤，就完成了SDK开发环境的准备，接下来讲解如何调用SDK的API。

# 初始化SDK

在uni-app的vue页面（通常为主页面）里面的<script>脚本的开始定义如下语句，引入SDK插件



const modal = uni.requireNativePlugin('modal');

const PhoneSdk = uni.requireNativePlugin('SparkleComm\_Plugin');

然后在onLoad()事件里进行SDK的初始化和注册；只需调用一次即可，范例如下。

注意SDK的相关事件将异步回调，通过PhoneSdk.InitSdk接口的result返回。

onLoad() {

this.InitSdk(); // 初始化SDK

this.RegisterToSipServer(); //注册服务器

},

// 在methods里面实现InitSdk和RegisterToSipServer函数

InitSdk() {// 初始化SDK

console.log('InitSdk')

PhoneSdk.InitSdk(result => { // VOIP相关的事件将通过result回调

const msg = JSON.stringify(result);

console.log("VOIP Event:" + msg);

switch (result.type) {

case 'onSipConnected': // SIP服务器注册成功

console.log("VOIP注册成功");

this.state = "在线"; // 更新界面状态

break;

case 'onSipDisconnected': // SIP注销

console.log("VOIP已经注销");

this.state = "离线"; // 更新界面状态

break;

case 'onCallTrans': // 呼叫转移，通话中进行呼叫转移时，会回调到此处，后续会讲到

console.log("VOIP呼叫转移");

uni.navigateTo({

url: "/SelectContacts?CallID="+result.CallID+"&"+"AccountID="+result.AccountID

})

break;

}

});

},

RegisterToSipServer() { // 注册SIP服务器

PhoneSdk.RegisterToSipServer({

serverAddr: "192.168.0.247", // SIP服务器地址

port: 5060, // 服务器端口

userID: "1019", // 用户名

userPWD: "123456", // 密码

accountID: 0, // 必须填0

domain: "192.168.0.247", // SIP服务器域名

authWithDomain: false, // 填false

idUri: "", // 填空

useIMS3GPP: false, // 填false

extraHeads: "" // 填空

}, result => {

const msg = JSON.stringify(result);

console.log("RegisterToSipServer:result=" + msg);

});

}

# 注销

UnRegister() { // 注销SIP服务器

PhoneSdk.UnRegister({

accountID: 0

}, result => {

const msg = JSON.stringify(result);

console.log("UnRegister:result=" + msg);

});

},

可以随时调用UnRegister()函数注销，注销成功后会通过 “ PhoneSdk.InitSdk(result =>”进行回调，如上面初始化化SDK时所示，回调的类型为“onSipDisconnected”；可以再次调用RegisterToSipServer进行重新注册，注册成功后的回调类型为“onSipConnected”。

# 拨打电话

PhoneSdk.MakeCall({

number: number, // 对方号码

video: video, // 是否视频通话

accountID: 0 // 账户ID，必须为0

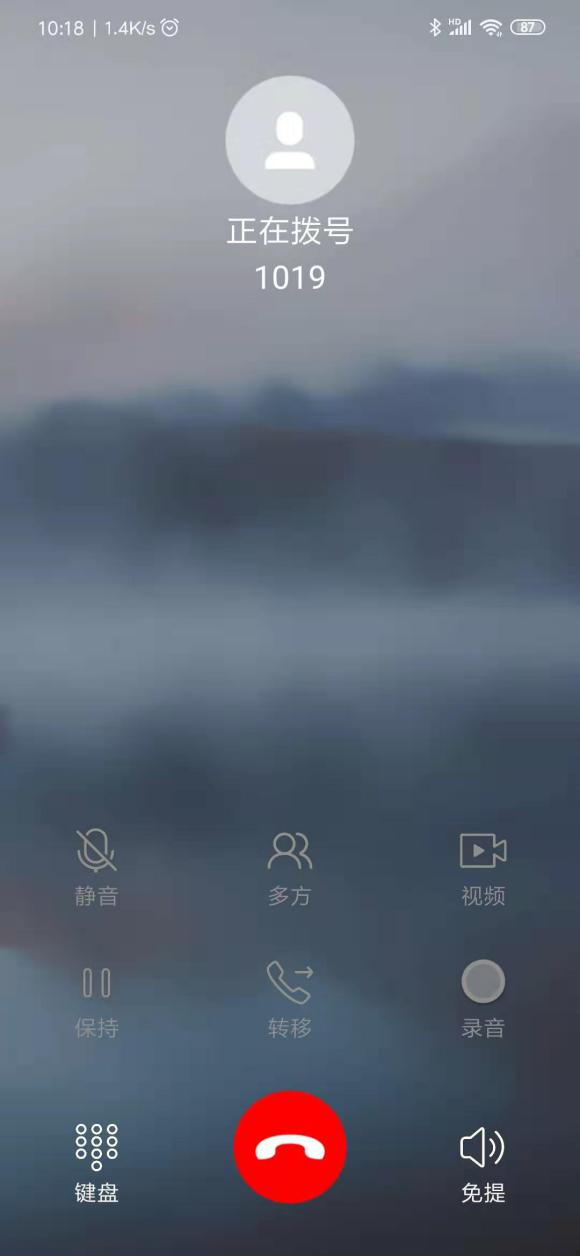
}, result => {

const msg = JSON.stringify(result);

console.log("MakeCall:result=" + msg);

});

将自动弹出通话界面，如下：

**

# 来电接听

来电时，自动打开接听界面，如下：



# 呼叫转移

在通话界面点击“转移”按钮时，将触发回调类型“onCallTrans”，参数为result.CallID和result.AccountID。收到此消息时，通话界面会自动隐藏，然后需要通过uni-app导航到一个选择联系人的界面，选择对应的联系人，然后调用呼叫转移接口，如下：

PhoneSdk.CallTransfer({

addr: number, // 对方号码

callID: CallID, // 回调参数里的result.CallID

accountID: AccountID // 回调参数里的result.AccountID

}, result => {

const msg = JSON.stringify(result);

console.log("CallTransfer:result=" + msg);

});

若用户没有选择联系人，取消了呼叫转移，则需要重新打开当前的通话界面，需要调用如下接口函数：

PhoneSdk.ShowInCall({

}, result => {

const msg = JSON.stringify(result);

console.log("CallTransfer:result=" + msg);

});