1,在你的RN项目中依赖voicesdk,见sometest项目中package.json文件

"voicesdk": "^1.0.2"

node\_modle依赖下载完毕之后，你的项目中的android目录最好使用Android studio打开然后同步一下

2，Android端配置：

你的项目名/android/app/src/main/java/com/你的项目名 的AndroidManifest.xml增加权限，复制位置详见sometest此.xml文件内容

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

<!-- 语音识别需要，部分为可选，详见讯飞文档 -->

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

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

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

<!--读取网络信息状态 -->

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

<!--获取当前wifi状态 -->

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

<!--允许程序改变网络连接状态 -->

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

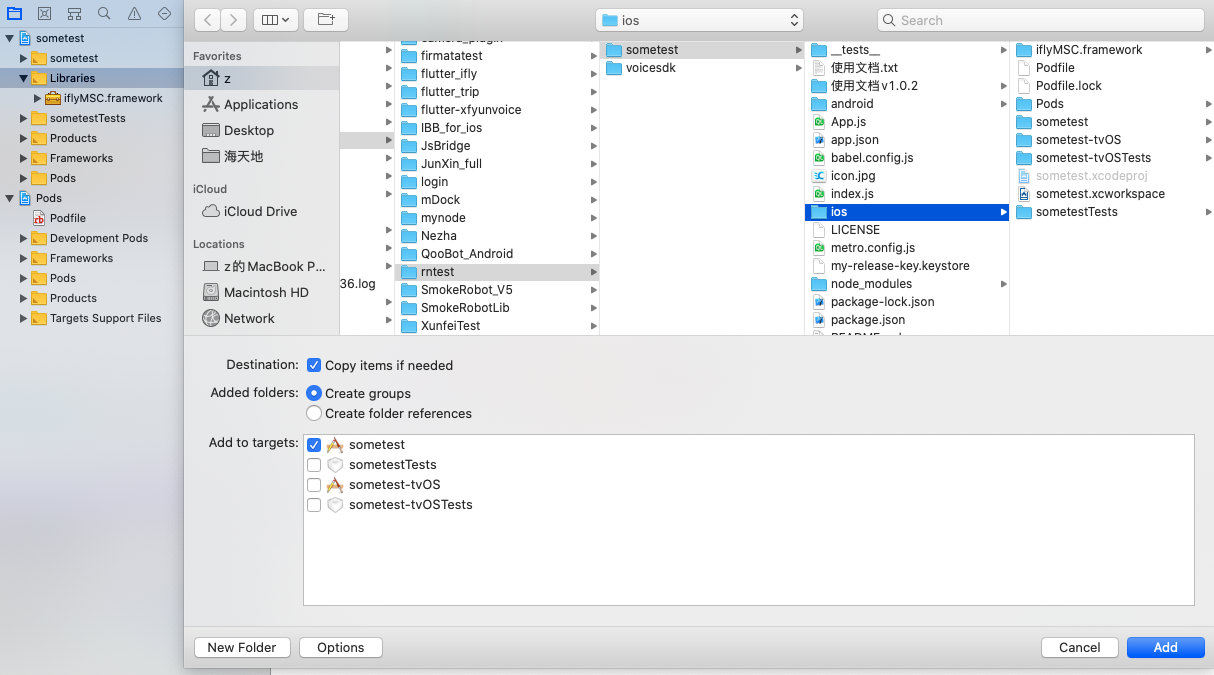
<!--读取手机信息权限 -->

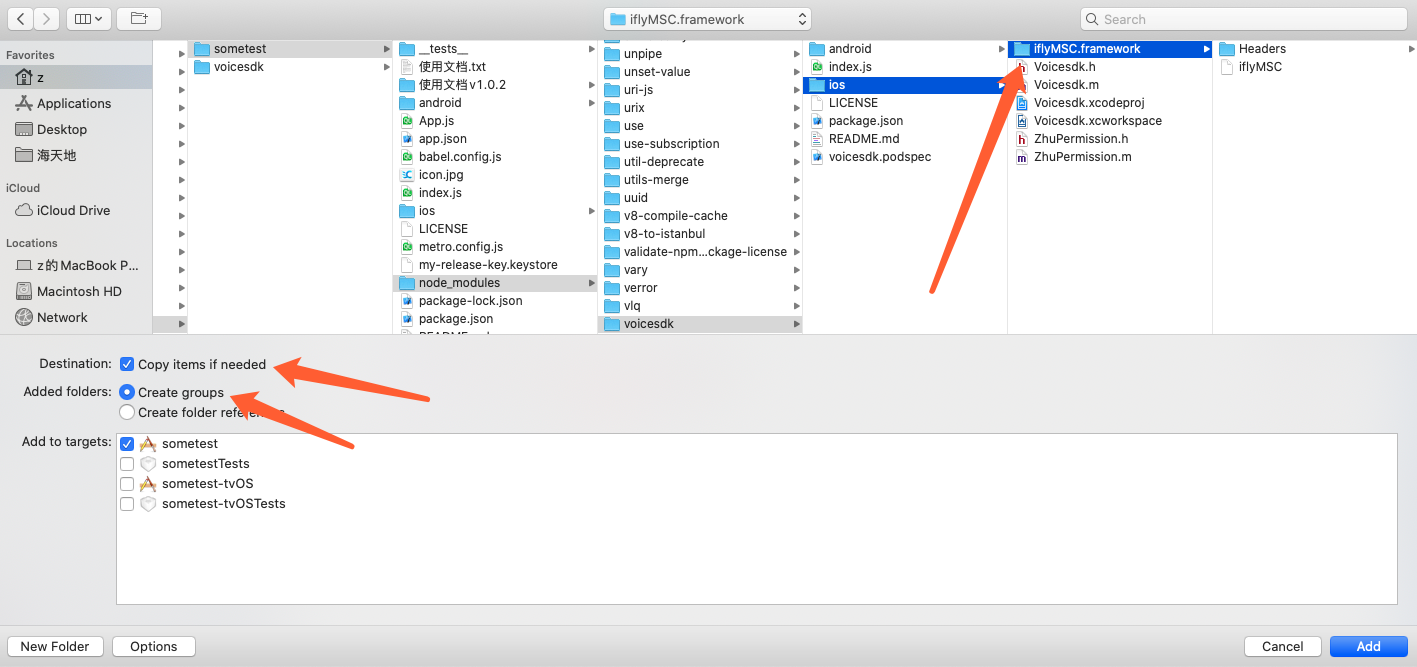
<uses-permission android:name="android.permission.READ\_PHONE\_STATE"/>

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

3，iOS端配置：

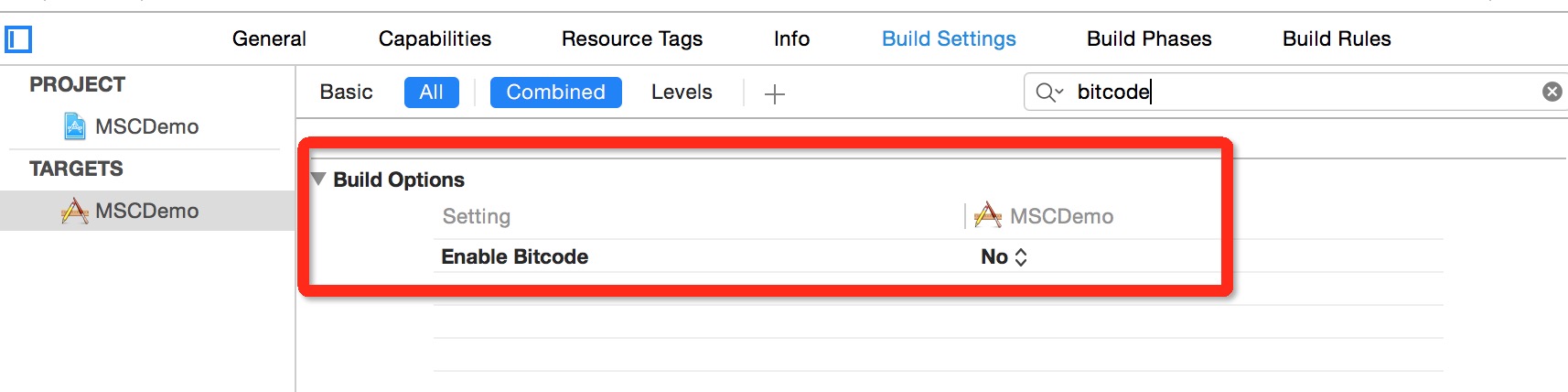
I.在项目的libraries文件加右击添加”add files to “sometest””





#### **设置Bitcode**

在Xcode 7,8默认开启了Bitcode，而Bitcode 需要工程依赖的所有类库同时支持。MSC SDK暂时还不支持Bitcode，可以先临时关闭。后续MSC SDK支持Bitcode 时，会在讯飞开放平台上进行SDK版本更新，请关注。关闭此设置，只需在Targets - Build Settings 中搜索Bitcode 即可，找到相应选项，设置为NO。



#### **用户隐私权限配置**

iOS 10发布以来，苹果为了用户信息安全，加入隐私权限设置机制，让用户来选择是否允许。 隐私权限配置可在info.plist 新增相关privacy字段，MSC SDK中需要用到的权限主要包括麦克风权限、联系人权限和地理位置权限：

<key>NSMicrophoneUsageDescription</key><string></string><key>NSLocationUsageDescription</key><string></string><key>NSLocationAlwaysUsageDescription</key><string></string><key>NSContactsUsageDescription</key><string></string>

即在Info.plist 中增加下图设置：



如果iOS Xcode运行的时候LOG里边打印的错误是：

Assertion failed: (severity >= 0 && severity < NUM\_SEVERITIES), function SetLogDestination,  
 file /Users/samyao/Git/trace/node\_modules/react-native/third-party/glog-0.3.4/src/logging.cc, line 595.

解决办法：

inline void LogDestination::SetLogDestination(LogSeverity severity,

const char\* base\_filename) {

// assert(severity >= 0 && severity < NUM\_SEVERITIES);

char\* base\_filename\_copy = nullptr; //初始化char\*类型

base\_filename\_copy = const\_cast<char\*>(base\_filename); //const char\*类型转char\*类型

if (severity >= 0 && severity < NUM\_SEVERITIES) {

//

} else {

severity = 0;

base\_filename\_copy = const\_cast<char\*>("glogTest");

}

// Prevent any subtle race conditions by wrapping a mutex lock around

// all this stuff.

MutexLock l(&log\_mutex);

//log\_destination(severity)->fileobject\_.SetBasename(base\_filename);

log\_destination(severity)->fileobject\_.SetBasename(base\_filename\_copy);

}

4,复制sdk.js文件到你的项目中，调用sdk.js

/Users/z/code/rntest/sometest/voice-view.js 这个组件中有调用步骤

I. /\*

第一步，申请原生权限

\*/

import sdk from './sdk.js';

sdk.requestPerssion((errCode) => {

console.log('view中 JS权限 = '+errCode);

if(errCode == 1)sdk.initRecognizer();

});

/\*\*

\* NLP网络请求需要的参数，根据请求需要参数设置

\*/

//sdk.setConfig({user\_id:'number for example 123'});

II./\*

第二步，初始化语音识别引擎,此函数也可以放在willMount的生命周期函数中

\*/

sdk.initRecognizer();

III./\*

第三步，引擎开始识别

\*/

sdk.startRecognizer((jsonObj, jsonString) => {

//this.setState({ enable: true, result: '' + jsonString })

});

IV./\*

第四步，用户主动停止语音识别

\*/

sdk.stopRecognizer();

V./\*

第五步，释放语音引擎，节省手机内存

\*/

componentWillUnmount(){

sdk.release();

}