

实时音视频 小程序端集成 产品文档



腾讯云

【版权声明】

©2013-2018 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

小程序端集成

准备工作

Demo部署

webrtc-room 标签

常见问题

小程序端集成 准备工作

最近更新时间：2018-05-17 21:44:39

开通小程序类目与推拉流标签【重要】

出于政策和合规的考虑，微信暂时没有放开所有小程序对 <live-pusher> 和 <live-player> 标签的支持：

- 个人账号和企业账号的小程序暂时只开放如下表格中的类目：

主类目	子类目
【社交】	直播
【教育】	在线教育
【医疗】	互联网医院，公立医院
【政务民生】	所有二级类目
【金融】	基金、信托、保险、银行、证券/期货、非金融机构自营小额贷款、征信业务、消费金融

- 符合类目要求的小程序，需要在小程序管理后台的“设置 - 接口设置”中自助开通该组件权限，如下图所示：



Demo部署

最近更新时间：2018-12-27 11:52:16

Demo 体验

升级微信到最新版本，发现页卡 -> 小程序 -> 搜索“腾讯视频云”，即可打开小程序 Demo：

再次确认您已经完成相关接口的开通

打开 [微信公众平台](#) 注册并登录小程序，在小程序管理后台的“**设置 - 接口设置**”中自助开通该组件权限，如下图所示：

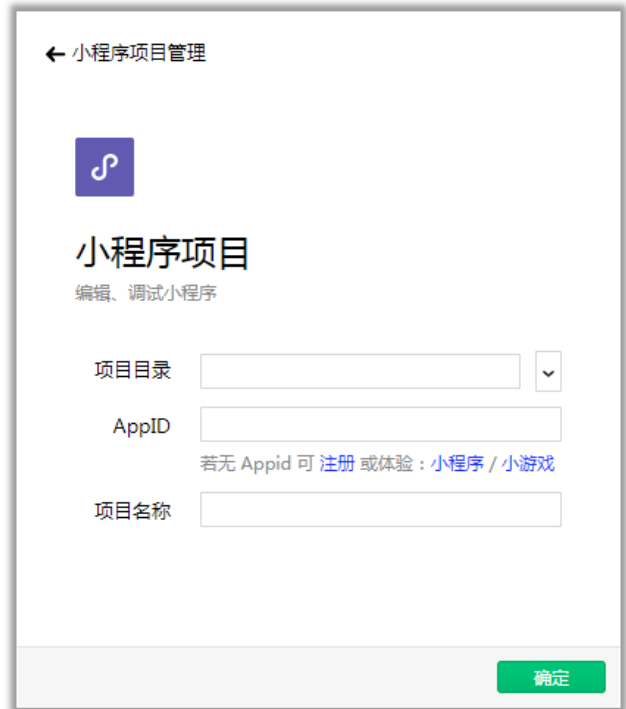


⚠ 注意：

如果以上设置都正确，但小程序依然不能正常工作，可能是微信内部的缓存没更新，请删除小程序并重启微信后，再进行尝试。

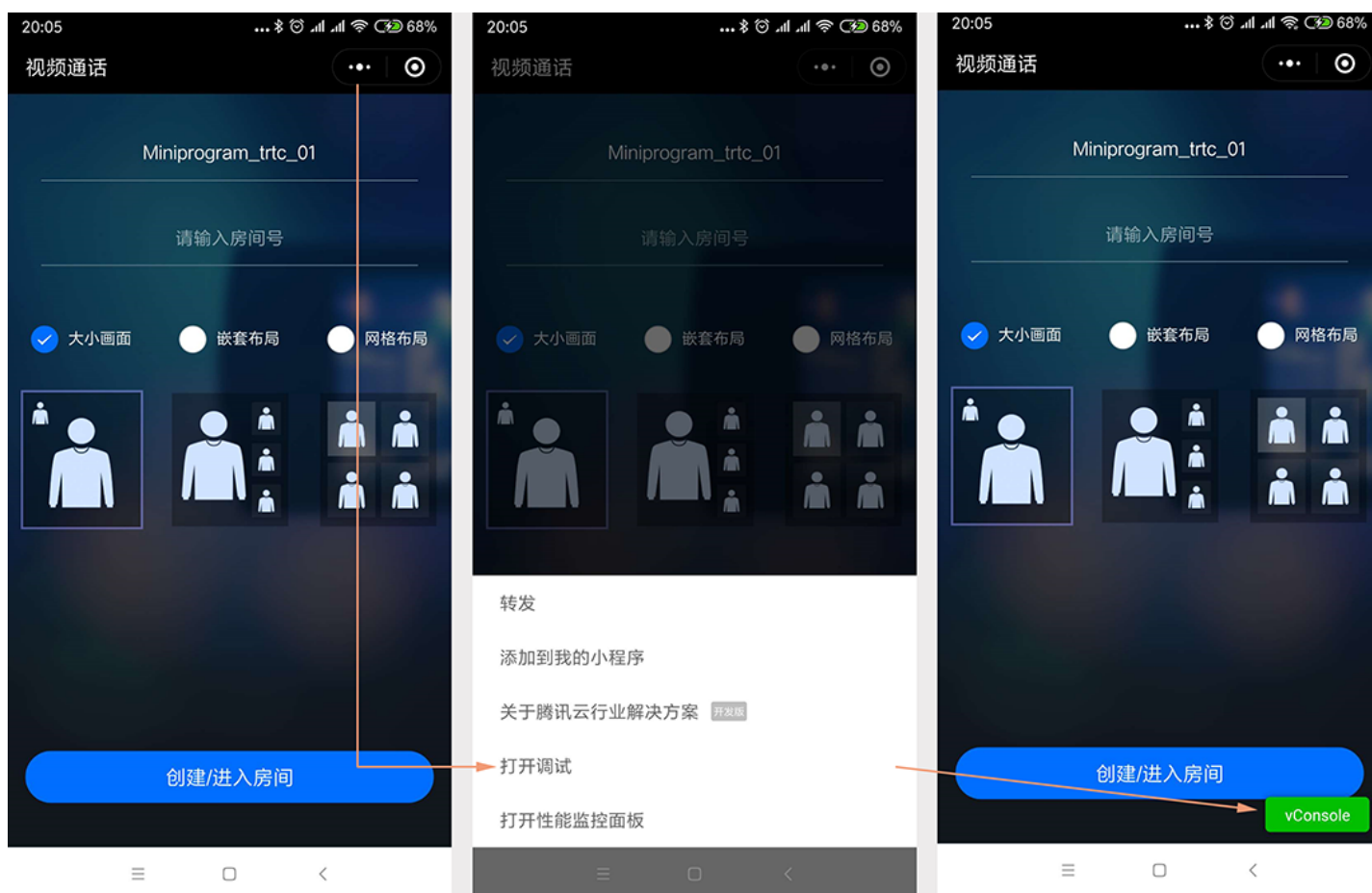
安装微信小程序开发工具

下载并安装最新版本的 [微信开发者工具](#)，使用小程序绑定的微信号扫码登录开发者工具。



获取 Demo 源码并调试

- step1 : 访问 [SDK + Demo](#) , 获取小程序 Demo 源码。
- step2 : 打开安装的微信开发者工具 , 单击【小程序项目】按钮。
- step3 : 输入小程序 AppID , 项目目录选择上一步下载下来的代码目录 (**注意** : 目录请选择**根目录** , 根目录包含有 `project.config.json` 文件) , 填写自己的小程序的 AppID , 单击【确定】创建小程序项目。
- step4 : 再次单击【确定】进入开发者工具。
- step5 : 请使用手机进行测试 , 直接扫描开发者工具预览生成的二维码进入。
- step6 : **开启调试模式** , 体验和调试内部功能。开启调试可以跳过把这些域名加入小程序白名单的工作。



webrtc-room组件会访问以下地址

<webrtc-room> 组件内部需要访问如下地址：

将以下域名在 [微信公众平台](#) ->开发->开发设置->服务器域名配置进行配置：

域名	说明
https://official.opensso.tencent-cloud.com	WebRTC音视频鉴权服务域名[1]
https://yun.tim.qq.com	WebRTC音视频鉴权服务域名[2]
https://cloud.tencent.com	推流域名
https://webim.tim.qq.com	IM域名

webrtc-room 标签

最近更新时间：2018-12-27 11:53:10

标签说明

<webrtc-room> 标签是基于 <live-pusher> 和 <live-player> 实现的用于 WebRTC 互通的自定义组件。如果您希望直接使用 <live-pusher> 和 <live-player> 标签完成对接，或者想要了解 <webrtc-room> 的内部原理，可以参考 [DOC](#)。

版本要求

- 微信 6.6.6 版本开始支持。

效果演示

- **PC 端**

用 Chrome 浏览器打开 [体验页面](#) 可以体验桌面版 WebRTC 的效果。

- **Android 端**

用 Android 手机 [下载 App](#) 或者扫码下载安装 App 即可体验 Android 的效果。



- **iOS 端**

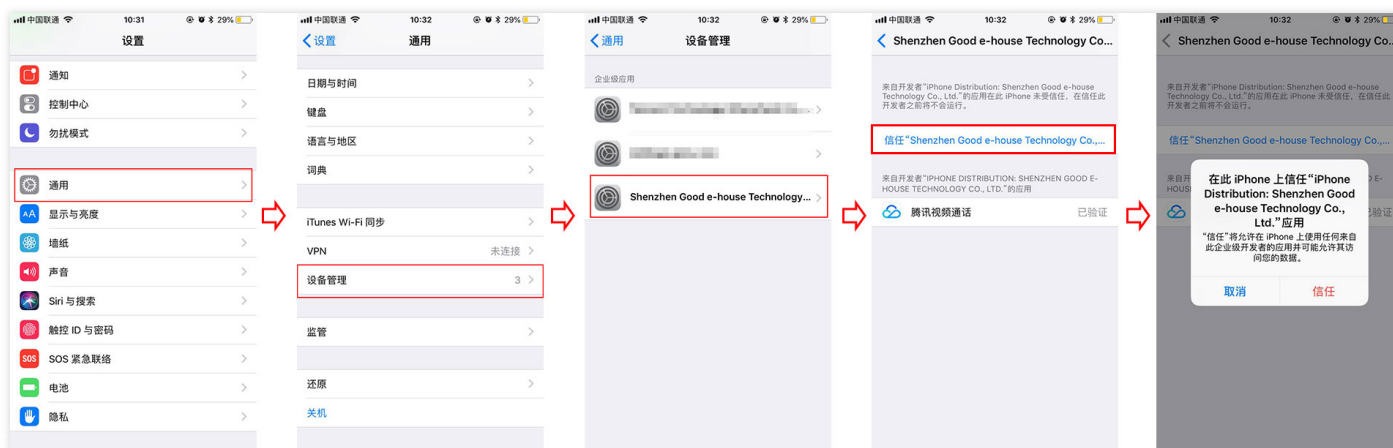
用 iOS 手机 扫码后通过 Safari 打开并安装 App 即可体验 iOS 的效果。



• 信任证书

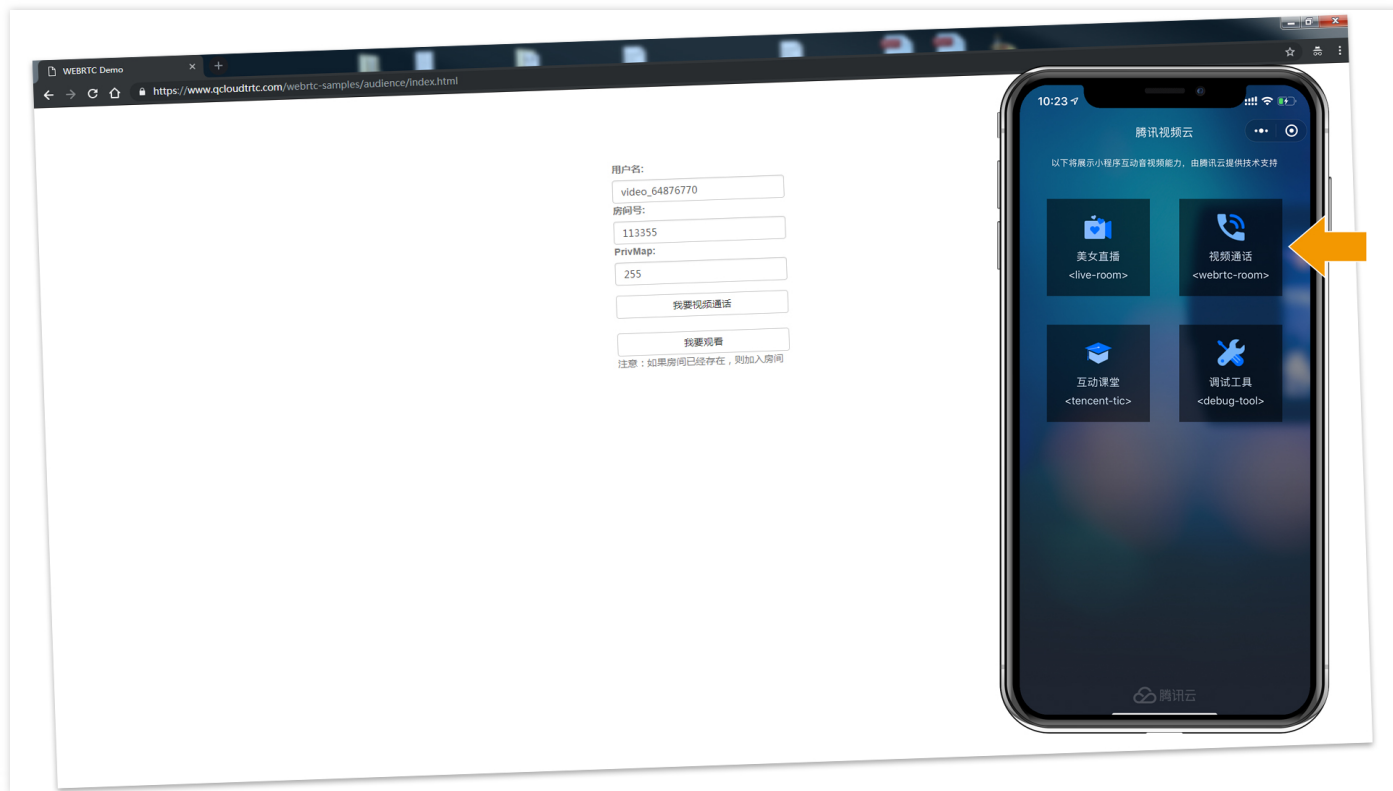
安装 App 后，还需要配置信任证书方可打开体验 (单击打开大图)

• 设置->通用->设备管理->选择证书->信任证书



• 微信端

发现=>小程序=>搜索“腾讯视频云”，单击 WebRTC 功能卡，就可以体验跟桌面版 Chrome 互通的效果了。



对接资料

对接资料	说明	github地址
小程序源码	包含<webrtc-room>的组件源码以及 demo 源码	前往
Web 端源码	基于 WebrtcAPI 实现的 Chrome 版 WebRTC 接入源码	前往

标签详解

属性定义

属性	类型	默认值	说明
template	String	'float'	必要，标识组件使用的界面模版，demo中内置bigsmall，float，grid三种布局（用户如果需要自定义界面，请看 界面定制 ）
sdkAppID	String		必要，开通实时音视频服务创建应用后分配的 sdkAppID

属性	类型	默认值	说明
userID	String		必要，用户 ID
userSig	String		必要，身份签名，相当于登录密码的作用
roomID	Number		必要，房间号
privateMapKey	String		可选，房间权限 key，相当于进入指定房间 roomID 的钥匙，如果trtc控制台开启了权限密钥，则需要传 privateMapKey (trtc控制台 ->选择您的应用->账号信息)
beauty	Number	0	可选，美颜指数，取值 0 - 9，数值越大效果越明显
whiteness	String	0	可选，美白指数，取值 0 - 9，数值越大效果越明显
muted	Boolean	false	可选，true 静音 false 不静音
debug	Boolean	false	可选，true 打印推流 debug 信息 fales 不打印推流 debug 信息
bindRoomEvent	Function		必要，监听 <webrtc-room> 组件返回的事件
enableIM	Boolean	false	可选，是否启用IM
bindIMEvent	Function		当IM开启时必要，监听 IM 返回的事件
aspect	String	9:16	可选，宽高比3:4, 9:16
minBitrate	String	200	可选，最小码率，该数值决定了画面最差的清晰度表现
maxBitrate	String	400	可选，最大码率，该数值决定了画面最好的清晰度表现
autoplay	Boolean	false	可选，进入房间后是否自动播放房间中其他的远程画面 true 自动播放 false 不自动播放
enableCamera	Boolean	true	可选，开启\关闭摄像头
pureAudioPushMod	Number		可选，纯音频推流模式，需要旁路直播和录制时需要带上此参数 1 => 本次是纯音频推流,不需要录制mp3文件 2 => 本次是纯音频推流,录制文件为mp3
recordId	Number		可选，自动录制时业务自定义id，将在录制完成后通过 直播录制回调 接口通知业务方，注意：如果小程序与小程序或者小程序与Web端互通，且传了recordId，必须保证web端和小程序传递的值一致

属性	类型	默认值	说明
enableCamera	Boolean	true	是否开启摄像头
smallViewLeft	String	'1vw'	小窗口距离大画面左边的距离，只在template设置为bigsmall有效
smallViewTop	String	'1vw'	小窗口距离大画面顶部的距离，只在template设置为bigsmall有效
smallViewWidth	String	'30vw'	小窗口宽度，只在template设置为bigsmall有效
smallViewHeight	String	'40vw'	小窗口高度，只在template设置为bigsmall有效
waitingImg	String		当微信切到后台时的垫片图片
loadingImg	String		画面loading图片

小程序实时音视频与 WebRTC 互通只需要保证两端的 sdkAppID 与 roomID 一致

操作接口

<webrtc-room> 组件包含如下操作接口，您需要先通过 selectComponent 获取 <webrtc-room> 标签的引用，之后就可以进行相应的操作了。

函数名	说明
start()	启动
pause()	暂停
resume()	恢复
stop()	停止
switchCamera()	切换摄像头
sendC2CTextMsg(receiveUser, msg, succ, fail)	发送C2C文本消息
sendC2CCustomMsg(receiveUser, msgObj, succ, fail)	发送C2C自定义消息
sendGroupTextMsg(msg, succ, fail)	发送群组文本消息
sendGroupCustomMsg(msgObj, succ, fail)	发送群组自定义消息

```
var webrtcroom = this.selectComponent("#webrtcroomid")
webrtcroom.pause();
```

sendC2CTextMsg

发送C2C文本消息

参数	类型	是否必填	描述
receiveUser	String	是	接收人的userid
msg	String	是	消息内容
succ	Function	否	发送成功的回调
fail	Function	否	发送失败的回调

sendC2CCustomMsg

发送C2C自定义消息

参数	类型	是否必填	描述
receiveUser	String	是	接收人的userid
msgObj	Object	是	{data: '消息内容', ext: '', desc: ''}
succ	Function	否	发送成功的回调
fail	Function	否	发送失败的回调

sendGroupTextMsg

发送群组文本消息

参数	类型	是否必填	描述
----	----	------	----

参数	类型	是否必填	描述
msg	String	是	消息内容
succ	Function	否	发送成功的回调
fail	Function	否	发送失败的回调

sendGroupCustomMsg

发送群组自定义消息

参数	类型	是否必填	描述
msgObj	Object	是	{data: '消息内容', ext: '', desc: ''}
succ	Function	否	发送成功的回调
fail	Function	否	发送失败的回调

事件通知

<webrtc-room> 标签通过 **onRoomEvent** 返回内部事件，通过 **onIMEvent** 返回 IM 消息事件，事件参数格式如下：

如果enableIM关闭了，则可以忽略onIMEvent

```
"detail": {
  "tag": "事件tag标识，具有唯一性",
  "code": "事件代码",
  "detail": "对应事件的详细参数"
}
```

示例代码

```
// Page.wxml 文件
<webrtc-room id="webrtcroom"
  roomId="{{roomId}}"
  userID="{{userID}}">
```

```
userSig="{{userSig}}"
sdkAppID="{{sdkAppID}}"
privateMapKey="{{privateMapKey}}"
template="1v3"
beauty="{{beauty}}"
muted="{{muted}}"
debug="{{debug}}"
bindRoomEvent="onRoomEvent"
enableIM="{{enableIM}}"
bindIMEvent="onIMEvent">
</webrtc-room>
```

// Page.js 文件

```
Page({
  data: {
    //...
    roomId: "",
    userID: "",
    userSig: "",
    sdkAppID: "",
    beauty: 3,
    muted: false,
    debug: false,
    enableIM: false
  },
  onRoomEvent: function(e){
    switch(e.detail.tag){
      case 'error': {
        //发生错误
        var code = e.detail.code;
        var detail = e.detail.detail;
        break;
      }
    }
  },
  onIMEvent: function(e){
    switch(e.detail.tag){
      case 'big_group_msg_notify':
        //收到群组消息
        console.debug( e.detail.detail )
        break;
      case 'login_event':
        //登录事件通知
        console.debug( e.detail.detail )
    }
  }
})
```

```
break;
case 'connection_event':
//连接状态事件
console.debug( e.detail.detail )
break;
case 'join_group_event':
//进群事件通知
console.debug( e.detail.detail )
break;
},

onLoad: function (options) {
self.setData({
  userID: self.data.userID,
  userSig: self.data.userSig,
  sdkAppID: self.data.sdkAppID,
  roomID: self.data.roomID,
  privateMapKey: res.data.privateMapKey
}, function() {
var webrtcroomCom = this.selectComponent('#webrtcroom');
if (webrtcroomCom) {
webrtcroomCom.start();
}
})
},

})
```

使用指引

请确认已经参照 [Demo部署](#) 开通了相关服务和并正确的完成了配置。

step1: 下载自定义组件源码

<webrtc-room> 并非微信小程序原生提供的标签，而是一个自定义组件，所以您需要额外的代码来支持这个标签。单击 [小程序源码](#) 下载源码包。

step2: 在工程中引入组件

- 在 page 目录下的 json 配置文件内引用组件，这一步是必须的，因为 <webrtc-room> 并非原生标签。


```
"usingComponents": {
  "webrtc-room": "/pages/webrtc-room/webrtc-room"
}
```

- 在 page 目录下的 wxml 文件中使用标签

```
<webrtc-room id="webrtcroomid"
  roomId="{{roomId}}"
  userID="{{userID}}"
  userSig="{{userSig}}"
  sdkAppID="{{sdkAppID}}"
  privateMapKey="{{privateMapKey}}"
  template="float"
  beauty="{{beauty}}"
  muted="{{muted}}"
  debug="{{debug}}"
  bindRoomEvent="onRoomEvent"
  enableIM="{{enableIM}}"
  bindIMEvent="onIMEvent">
</webrtc-room>
```

step3: 获取 key 信息

按照如下表格获取关键的 key 信息，这是使用腾讯云互通直播服务所必须的几个信息：

KEY	示例	作用	获取方案
sdkAppID	1400087915	用于计费和业务区分	step1 中获取
userID	xiaoming	用户名	可以由您的服务器指定，或者使用小程序的openid
userSig	加密字符串	相当于 userid 对应的登录密码	由您的服务器签发 (PHP / JAVA)
roomId	12345	房间号	可以由您的服务器指定
privateMapKey	加密字符串	当开启权限密钥时需要传。进房票据：相当于是进入 roomId 的钥匙	由您的服务器签发 (PHP / JAVA)

下载 [sign_src.zip](#) 可以获得服务端签发 userSig 和 privateMapKey 的计算代码 (生成 userSig 和 privateMapKey 的签名算法是 **ECDSA-SHA256**)。

step4: 进入房间

```
self.setData({
  userID: userID,
  userSig: userSig,
  sdkAppID: sdkAppID,
  roomID: roomID,
  privateMapKey: privateMapKey
}, function() {
  var webrtcroomCom = this.selectComponent('#webrtcroomid');
  if (webrtcroomCom) {
    webrtcroomCom.start();
  }
})
```

界面定制

- 创建界面模版

//第一步：新建 /pages/templates/mytemplate 文件夹，并创建 mytemplate.wxml 和 mytemplate.wxss 文件

//第二步：编写 mytemplate.wxml 和 mytemplate.wxss 文件

```
//mytemplate.wxml
<template name='mytemplate' >
  <view class='videoview' >
    <view class="pusher-box">
      <live-pusher
        id="rtcpusher"
        autopush
        mode="RTC"
        url="{{pushURL}}"
        aspect="{{aspect}}"
        min-bitrate="{{minBitrate}}"
        max-bitrate="{{maxBitrate}}"
        audio-quality="high"
        beauty="{{beauty}}"
        muted="{{muted}}"
        waiting-image="https://mc.qcloudimg.com/static/img/
        daeed8616ac5df256c0591c22a65c4d3/pause_publish.jpg"
        background-mute="{{true}}"
        debug="{{debug}}"
```

```
bindstatechange="onPush"
binderror="onError">
<cover-image class='character' src="/pages/Resources/mask.png"></cover-image>
<cover-view class='character' style='padding: 0 5px;'>我</cover-view>
</live-pusher>
</view>
<view class="player-box" wx:for="{{members}}" wx:key="userID">
<view class='poster'>
<cover-image class='set'
src="https://miniprogram-1252463788.file.myqcloud.com/roomset_{{index + 2}}.png">
</cover-image>
</view>
<live-player
id="{{item.userID}}"
autoplay
mode="RTC"
wx:if="{{item.accelerateURL}}"
object-fit="fillCrop"
min-cache="0.1"
max-cache="0.3"
src="{{item.accelerateURL}}"
debug="{{debug}}"
background-mute="{{true}}"
bindstatechange="onPlay">
<cover-view class='loading' wx:if="{{item.loading}}">
<cover-image src="/pages/Resources/loading_image0.png"></cover-image>
</cover-view>
<cover-image class='character' src="/pages/Resources/mask.png"></cover-image>
<cover-view class='character' style='padding: 0 5px;'>{{item.userName}}</cover-view>
</live-player>
</view>
</view>
</template>

//mytemplate.wxss
.view {
background-repeat:no-repeat;
background-size: cover;
width: 100%;
height: 100%;
}
```

- **webrtc-room 组件引入模版**

```
//为 <webrtc-room> 组件中的 webrtcroom.wxml 文件添加自定义模版
<import src='/pages/templates/mytemplate/mytemplate.wxml' />
<view class='component-box' >
<view styles="width:100%;height=100%;" wx:if="{{template=='1v3'}}" >
<template is='mytemplate' data="{{pushURL, aspect,
minBitrate, maxBitrate, beauty, muted, debug, members}}"/>
</view>
</view>
```

```
//为 <webrtc-room> 组件中的 webrtcroom.wxss 文件添加自定义样式
@import "../templates/mytemplate/mytemplate.wxss";
```

Chrome端对接

了解腾讯云官网的 [WebrtcAPI](#) ，可以对接 Chrome 端的 H5 视频通话，因为不是本文档的重点，此处不做赘述。

常见问题

最近更新时间：2018-12-27 11:53:44

1. 运行小程序进入多人音视频看不到画面？

- 请确认使用手机来运行，微信开发者工具内部的模拟器目前还不支持直接运行
- 检查安全域名是否有正确配置
- 请确认小程序基础库版本 `wx.getSystemInfo` 可以查询到该信息，1.7.0 以上的基础库才支持音视频能力。
- 请确认小程序所属的类目，由于监管要求，并非所有类目的小程序都开发了音视频能力，已支持的类目请参考 [DOC](#)。
- 如有更多需求，或希望深度合作，可以提工单或客服电话（400-9100-100）联系我们。

2. live-pusher、live-player标签使用及错误码参考

webrtc-room实际是由live-pusher和live-player封装而成，具体的错误码可以查看live-pusher和live-player。

- [live-pusher&错误码](#)
- [live-player&错误码](#)
- [livePusherContext](#)
- [livePlayerContext](#)

3. 如果需要上线或者部署正式环境怎么办？

域名	说明
https://official.opensso.tencent-cloud.com	WebRTC音视频鉴权服务域名[1]
https://yun.tim.qq.com	WebRTC音视频鉴权服务域名[2]
https://cloud.tencent.com	推流域名
https://webim.tim.qq.com	IM域名

设置

基本设置 **开发设置** 第三方服务 接口设置 开发者工具

开发者ID

开发者ID

AppID(小程序ID)

AppSecret(小程序密钥)

服务器域名

服务器配置	说明
<u>request合法域名</u>	https://yun.tim.qcloud.com