

互动直播

IE 浏览器集成

产品文档



腾讯云

【版权声明】

©2013-2019 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

- IE 浏览器集成
 - 下载代码
 - 音视频接口
 - 互动消息和上麦接口

IE 浏览器集成

下载代码

最近更新时间：2019-05-16 10:11:27

互动直播 SDK for IE，下文称为 iLiveSDK(IE)，是互动直播在 IE 平台上的 SDK。通过 ActiveX 实现了在 IE 上进行互动直播、上麦和基础 IM 的能力。

支持的浏览器

32 位 IE9，32 位 IE10，IE11。

下载 SDK 和 Demo

Demo 工程是基于 iLiveSDK(IE) 开发的互动直播应用。应用名为随心播，可以与其他终端的随心播互通。

您可以在这里 [在线体验](#) 其效果，也可以在 [GitHub](#) 下载 iLiveSDK(IE) 及其 Demo。其中包含了 iLiveSDK(IE)、接口文档、JS 接口文件和接口调用示例等，具体如下。

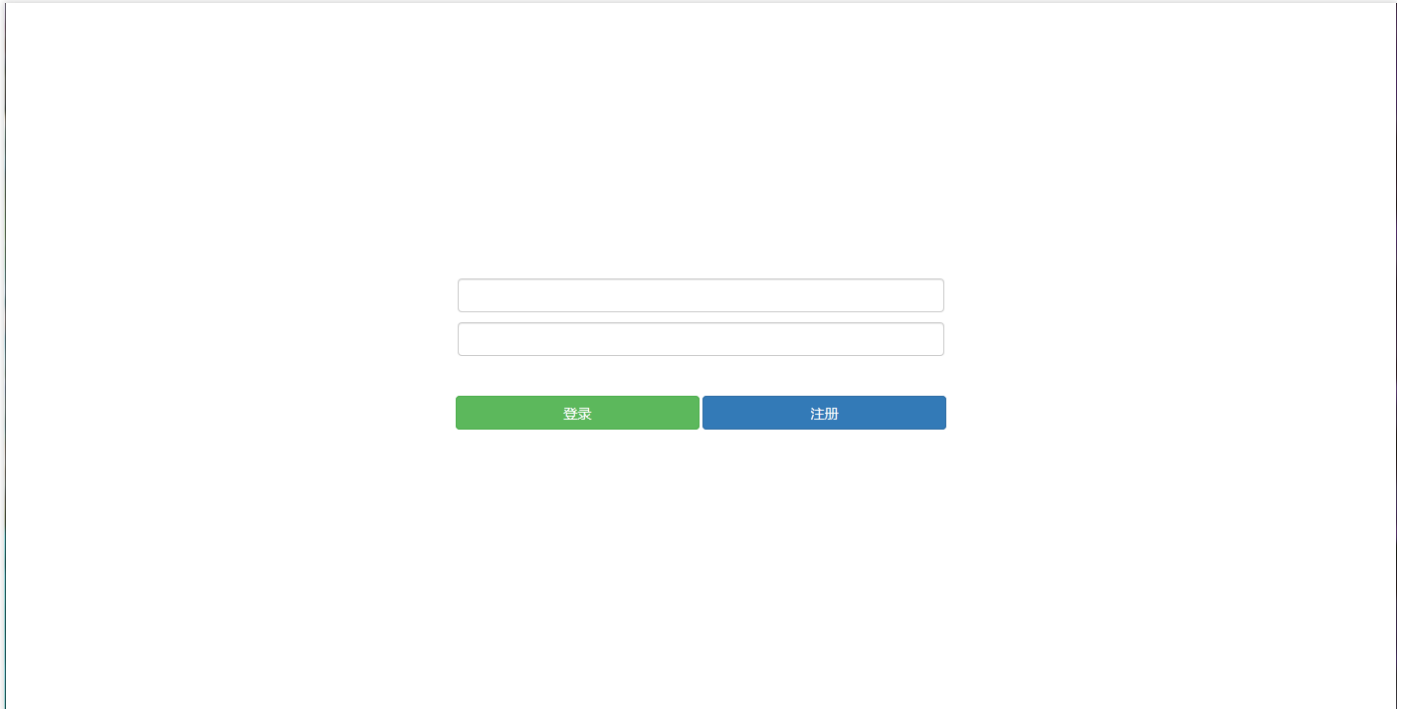
文件	说明
/suixinbo	Demo 随心播工程，可以用 IE 打开 index.html
iLiveSDK/iLiveSDK.cab	互动直播组件，业务层不会直接与此文件交互
iLiveSDK/iLiveSDK.js	互动直播接口，封装了 iLiveSDK.cab 的接口供业务层调用
/doc	相关文档
/tools	开发者工具

Demo 运行及其注意事项

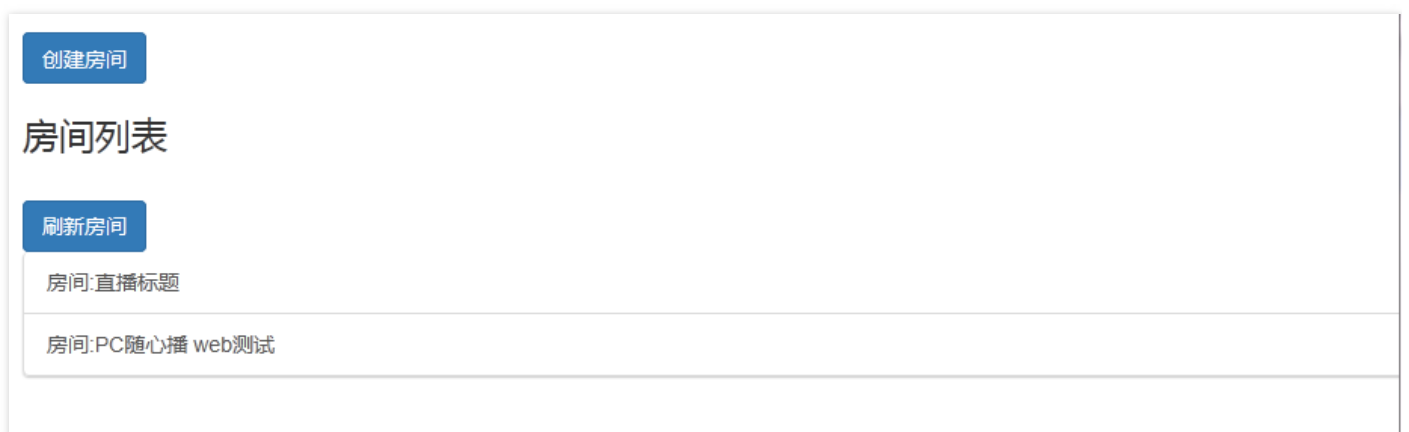
运行 Demo 可以对 iLiveSDK(IE) 具备的能力有直观的印象。您可以参照如下步骤运行 Demo。

- 把随心播代码中的 appid 和 accountType 修改成开发者自己的。即在 demo.js 中找到 OnInit 方法，找到 `sdk = new iLiveSDK(1400027849, 11656, "iLiveSDKCom")` 语句，用自己的 SDKAppID 和 accountType 替换前两个参数。如何获取这两个参数，可以参考 [快速参数配置](#)。

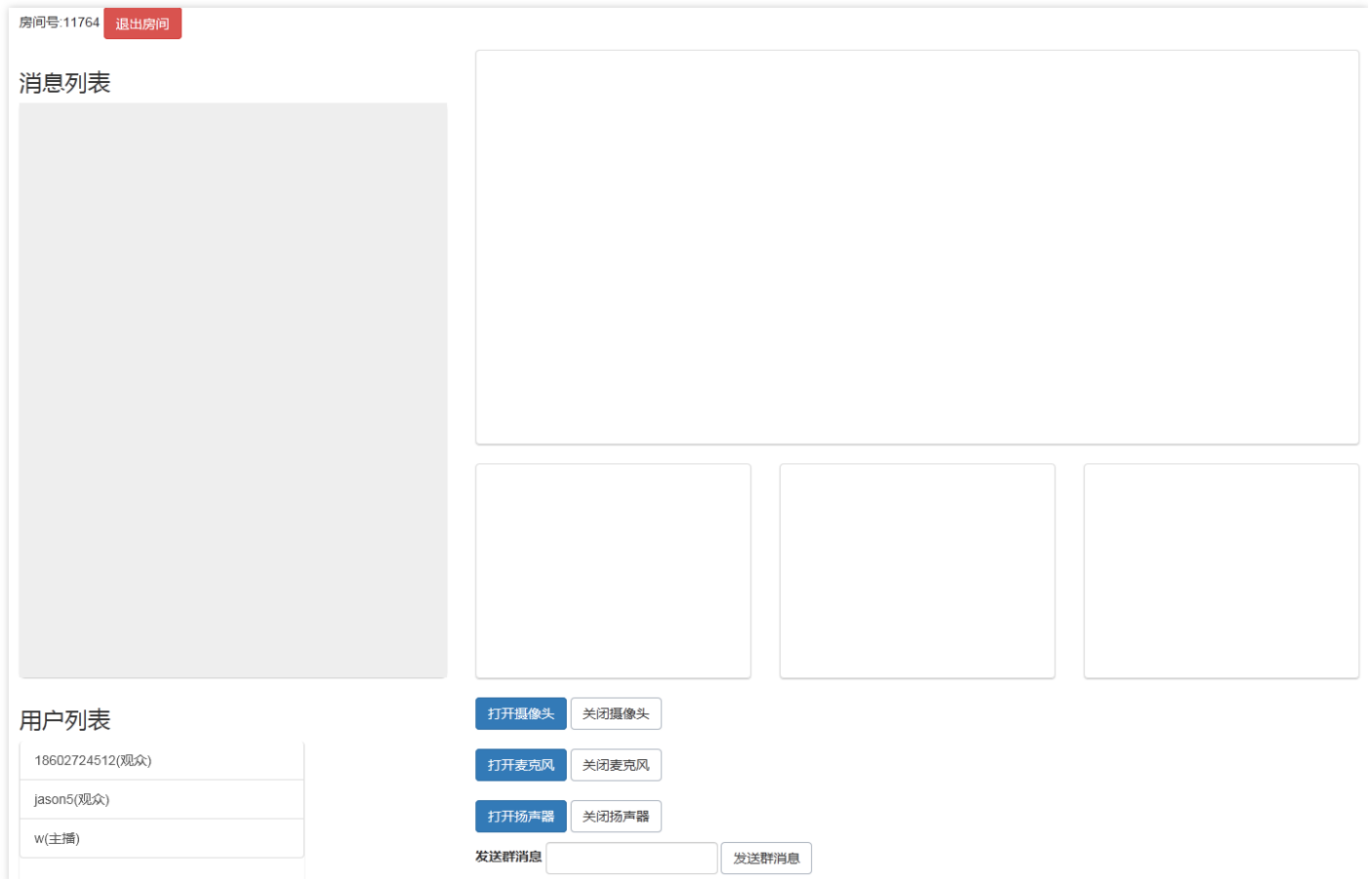
- 将 [随心播后台代码](#) 部署到自己服务器上，并按照文档修改后台的密钥。
- 用 IE 打开 index.html，并允许 activeX 控件，可以看到注册和登录界面。



- 登录成功后可以看到当前正在直播的房间列表（房间是 iLiveSDK 的概念，后文详述）。您也可以自己创建一个直播。



- 进入一个房间，可以看到直播的视频、群消息，当前房间成员等。您可以打开摄像头进行直播，给其他房间成员发消息等。



将 iLiveSDK(IE)集成入自己的工程

集成 iLiveSDK(IE) 仅需使用 JavaScript。将 SDK 引入工程的步骤如下。

- 将 cab 文件和 iLiveSDK.js 放入电脑任意位置
- 在 HTML 页面中添加

```
<object id="iLiveSDK" classid="CLSID:54E71417-216D-47A2-9224-C991A099C531" codebase="路径/iLiveSDK.cab#version=版本号"></object>
```

- 在 HTML 页面中添加 JS 接口文件

```
<script type="text/javascript" src="路径/iLiveSDK.js"></script>
```

- 调用 iLiveSDK 中的接口实现业务需求

SDK 日志位置

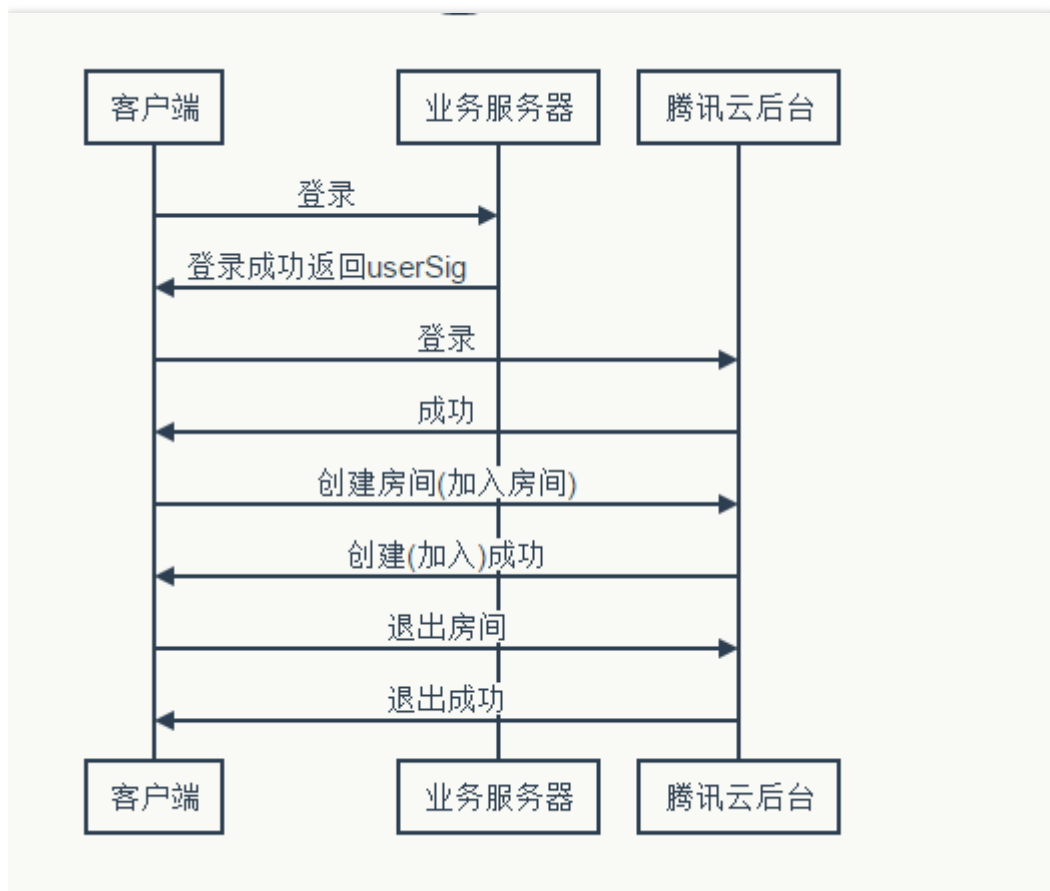
日志地址：`%appdata%\Tencent\iLiveSDK\txsdklog`（在开始菜单运行中执行）。

音视频接口

最近更新时间：2019-05-16 10:00:41

概述

iLiveSDK 中的音视频通讯能力被抽象为房间这个概念。在同一个房间内的成员可以看到房间内其他成员的音视频，每个房间同时最多可以有四路视频。iLiveSDK(IE)提供了房间管理，音视频设备管理等功能，方便用户建立互动视频。具体的直播业务流程如下：



注：接口详细参数请参阅 [iLiveSDK_Help](#)。

创建对象

创建 iLiveSDK 对象供后续使用。


```
//iLiveSDK 是 cab 组件的 ID  
var sdk = new ILiveSDK(sdkappid, accounttype, "iLiveSDK");
```

初始化

使用其他各项功能前必须将 iLiveSDK 初始化。

```
sdk.init(function () {  
  alert("init succ");  
}, function (errMsg) {  
  alert("错误码:" + errMsg.code + " 错误信息:" + errMsg.desc);  
});
```

登录

用户在登录后才能使用消息通讯，互动视频等功能。登录需要填写用户 ID 和签名。其中签名是由[腾讯登录服务](#)提供的。

```
sdk.login(id, sig, function () {  
  alert("login succ");  
}, function (errMsg) {  
  alert("错误码:" + errMsg.code + " 错误信息:" + errMsg.desc);  
});
```

音视频权限管理

业务层需重点关注房间成员的音视频权限。 只有主播或者需要上麦的观众才能拥有音视频上行的权限。在进入或者创建房间时需要填写正确的权限（详见下文进入房间）。另外成员允许在房间内改变自己的权限。

您可以在腾讯云[控制台配置](#)自身业务需要的角色及其权限，腾讯云服务器会根据房间成员不同的权限分配不同的[接入机](#)。错误配置权限可能导致不必要的带宽支出和观众异常上行数据等问题。

进入/创建房间

创建房间可以设置角色权限，房间号，房间号不能重复。

```
sdk.createRoom(roomid, "LiveMaster", function () {
  alert("create room succ");
}, function (errMsg) {
  alert("错误码:" + errMsg.code + " 错误信息:" + errMsg.desc);
});
```

加入房间可以填写角色权限，房间号。

**** 特别注意:** 普通观众加入房间时，应该配置authBits无上行权限，仅观看权限。否则会 and 主播一样走**核心机房DC**，产生高额费用。 ******

```
sdk.joinRoom(roomid, authBits, "Guest", function () {
  alert("join room succ");
}, function (errMsg) {
  alert("错误码:" + errMsg.code + " 错误信息:" + errMsg.desc);
});
```

设备管理

打开摄像头

在房间中拥有上行权限的成员打开摄像头，其他房间成员就可以获取视频画面数据。没有权限的成员的**上行数据**会被服务器丢弃。在打开摄像头前可以先调用接口获取当前的摄像头列表。

```
var szRet = sdk.getCameraList();
if (szRet.code != 0) {
  alert("获取摄像头列表出错; 错误码:" + szRet.code);
  return;
}
var nRet = sdk.openCamera(szRet.cameras[0].id);
if (nRet != 0) {
  alert("打开摄像头失败; 错误码:" + nRet);
}
```

关闭摄像头

需要停止视频时，需要关闭摄像头。

```
var nRet = sdk.closeCamera();
```

打开麦克风

```
var nRet = sdk.openMic();
```

关闭麦克风

```
var nRet = sdk.closeMic();
```

视频渲染

收到每个用户的视频数据后，需要使用渲染控件来渲染收到的视频

```
render.setIdentifier( userid ); //渲染控件设置了用户 ID 后,收到此用户的视频数据,将会自动渲染
```

用户的视频数据停止后，需要对渲染控件进行释放。

```
render.freeRender();
```

美颜

设置美颜程度 0~10。

```
sdk.setBeauty(beauty);
```

设置美白程度 0~10。

```
sdk.setWhite(white);
```

销毁美颜滤镜资源。如果开启了美颜功能，在**相机关闭**时，调用即可。

```
sdk.destroyFilter();
```

注：

- SDK 默认关闭美颜。

- 美颜 和 美白程度都设置为 0，SDK 自动关闭美颜功能；其中一项不为 0，SDK 打开美颜功能。
- sdk.destroyFilter 接口建议在关闭相机后调用。

退出房间

退出房间并回收资源。

```
sdk.quitRoom(function () {  
    alert("quit room succ");  
}, function (errMsg) {  
    alert("错误码:" + errMsg.code + " 错误信息:" + errMsg.desc);  
});
```

登出

登录的逆操作。

```
sdk.logout(function () {  
    alert("logout succ");  
}, function (errMsg) {  
    alert("错误码:" + errMsg.code + " 错误信息:" + errMsg.desc);  
});
```

在线状态

用户登录后是有在线状态的，每个账号同时只能在一个终端登录。在其他终端登录时将会把当前终端的登录踢下线，当前终端会收到被踢下线通知。SDK 允许设置踢下线通知的监听，业务层收到通知后可以在上层处理登录态的变化。

```
sdk.setForceOfflineListener(function() {  
    alert("您已被踢下线，请重新登录");  
});
```

互动消息和上麦接口

最近更新时间：2018-06-29 17:54:26

iLiveSDK(IE)提供了消息通讯的功能。基于消息通讯可以实现房间内成员的群消息，两个用户之间的 C2C 消息（不用加好友）。当前 SDK 只支持发文本消息和自定义消息。

发群消息

当前仅支持给当前所在房间发群消息，房间内其他成员都会收到改消息。

```
//ILiveMessageElem, ILiveMessage 的参数说明详见接口文档
var elem = new ILiveMessageElem(0, "message content");
var elems = [];
elems.push(elem);
var message = new ILiveMessage(elems);
sdk.sendGroupMessage(message, function () {
    alert("send message succ");
}, function (errMsg) {
    alert("错误码:" + errMsg.code + " 错误信息:" + errMsg.desc);
});
```

接收群消息

设置群消息监听后可以收到群消息，群消息可以在初始化后设置。

```
sdk.setMessageListener(function (msg) {
    //msg 的定义详见接口文档
})
```

发 C2C 消息

C2C 消息指的是两个用户之间的点对点聊天消息。邀请房间成员上麦等消息可以在业务层通过 C2C 自定义消息实现。具体的实现可以参考 Demo。

```
//ILiveMessageElem, ILiveMessage 的参数说明详见接口文档
var elem = new ILiveMessageElem(0, "message content");
```

```
var elems = [];  
elems.push(elem);  
var message = new ILiveMessage(elems);  
sdk.sendC2CMessage("somebody", message, function () {  
  alert("send message succ");  
}, function (errMsg) {  
  alert("错误码:" + errMsg.code + " 错误信息:" + errMsg.desc);  
});
```

接收 C2C 消息

设置 C2C 消息监听后可以收到 C2C 消息，C2C 消息可以在初始化后设置。

```
sdk.setMessageListener(function (msg) {  
  //msg 的定义详见接口文档  
})
```

连麦互动

上麦行为主要通过修改音视频权限达到。

上麦流程：

【切换角色】 --> 【打开摄像头】 --> 【渲染画面】

下麦流程：

【切换角色】 --> 【关闭摄像头】 --> 【结束渲染】

您可以在控制台 [配置场景和角色](#)，然后调用接口切换角色。

```
//其中 liveMaster 是控制台配置的角色  
sdk.changeRole("LiveMaster");
```