

北京云枢讯联科技有限公司

# YS SDK For JavaScript

---

V2.0.4

北京云枢讯联科技有限公司研发中心 著

2019/12/03

为避免开发过程中遇到不必要的麻烦，请相关人员在使用 SDK 前仔细阅读此文档。此文档包括术语解释及接口说明等。

目录

目录..... 1

YS SDK(JS)接口文档..... 5

1 集成 Room SDK..... 5

    1.1 环境准备..... 5

    1.2 导入..... 5

2 重要概念..... 5

    2.1 房间..... 5

    2.2 用户..... 5

    2.3 音视频..... 5

3 使用指南..... 6

    3.1 进出房间..... 6

    3.3 用户..... 6

    3.4 音视频..... 6

    3.5 自定义信令..... 7

4 接口说明..... 8

    4.1 全局接口..... 8

        获取 SDK 版本号 (getSdkVersion) ..... 8

    4.2 房间管理类 (Room) ..... 8

        实例化 (Room) ..... 8

        初始化 (init) ..... 8

        取消初始化 (uninit) ..... 9

        检测是否初始化 (checkInit) ..... 9

        加入房间 (joinroom) ..... 9

        加入房间 (joinRoomEx) ..... 10

        离开房间 (leaveroom) ..... 11

        获取房间信息 (getRoomProperties) ..... 11

        获取自己的用户对象 (getMySelf)..... 11

        获取指定用户对象 (getUser) ..... 11

        获取房间中所有用户对象 (getUsers) ..... 11

        获取本地设备管理类对象 (getDeviceMgr) ..... 12

        发布本地视频 (publishVideo) ..... 12

        停止发布本地视频 (unpublishVideo) ..... 12

        发布本地音频 (publishAudio) ..... 13

        停止发布本地音频 (unpublishAudio) ..... 13

        播放视频 (playVideo) ..... 13

        停止播放视频 (unplayVideo) ..... 14

        播放音频 (playAudio) ..... 14

        停止播放音频 (unplayAudio) ..... 14

        设置播放音量 (setRemoteAudioVolume) ..... 15

        获取音量级别 (getAudioVolume) ..... 15

        注册音量监听器 (registerAudioVolumeListener) ..... 16

移除音量监听器 (unregisterAudioVolumeListener)	17
设置日志级别 (setLogIsDebug)	17
注册日志监听器 (registerLogListener)	18
发送聊天消息 (sendMessage)	18
发布自定义消息 (pubMsg)	19
删除自定义消息 (delMsg)	20
改变用户属性 (changeUserProperty)	21
改变指定角色的用户属性 (changeUserPropertyByRole)	22
批量改变用户属性 (batchChangeUserProperty)	23
将用户请出房间 (evictUser)	23
设置是否自动处理设备改变 (插拔) 事件 (setAutoProcessDeviceChangeEvent)	24
获取视频属性 (getVideoProfile)	24
设置视频属性 (setVideoProfile)	24
请求服务器列表 (requestServerList)	25
切换服务器 (switchServer)	25
切换房间音视频状态 (switchOnlyAudioRoom)	26
服务器录制 (start/stopServerRecord)	26
播放共享的媒体文件 (play/unplayRemoteMedia)	27
获取房间的用户列表 (getRoomUsers)	27
获取房间的用户总数 (getRoomUserNum)	28
共享媒体文件 (start/stop/pauseShareMedia ...)	29
获取文件列表 (getFileList)	30
获取文件信息 (getFileinfo)	30
设置本地视频的镜像 (setLocalVideoMirror)	30
上传文件 (uploadRoomFile)	31
删除文件 (deleteRoomFile)	31
获取内网 IPv4 (getIp)	32
获取指定人选的网络状态 (getStreamStats)	32
<b>4.3 事件分发器 (EventDispatcher)</b>	<b>33</b>
实例化 (EventDispatcher)	33
分发事件 (dispatchEvent)	33
添加事件监听者 (addEventListener)	33
移除事件监听者 (removeEventListener)	34
移除所有事件监听者 (removeAllEventListener)	34
<b>4.4 日志类(Logger)</b>	<b>34</b>
debug 日志 (Logger.debug)	34
trace 日志 (Logger.trace)	35
info 日志 (Logger.info)	35
warning 日志 (Logger.warning)	35
error 日志 (Logger.error)	35
<b>4.5 本地设备管理类 (DeviceMgr)</b>	<b>35</b>
枚举系统设备 (getDevices)	35
枚举视频设备 (getCameras)	36
枚举音频输入设备 (getMicrophones)	37

枚举音频输出设备 (getSpeakers).....	37
设置系统设备 (setDevices) .....	37
设置摄像头设备 (setCamera) .....	38
设置麦克风设备 (setMicrophone) .....	38
设置扬声器设备 (setSpeaker) .....	38
检测是否支持设置扬声器 (checkIsSupportSetSpeaker) .....	39
摄像头检测 (start/stopVideoTest) .....	39
麦克风检测 (start/stopMicrophoneTest) .....	40
扬声器检测 (start/stopSpeakerTest) .....	41
网络质量检测 (start/stopNetworkTest) .....	42
注册设备监听器 (registerDeviceChangeListener) .....	42
移除设备监听器.....	43
4.6 用户类 (RoomUser) .....	43
id 属性.....	43
nickname 属性.....	43
role 属性.....	43
hasaudio 属性 【注：不可修改此属性】 .....	43
hasvideo 属性 【注：不可修改此属性】 .....	43
publishstate 属性 【注：不可修改此属性】 .....	44
ys_maincamera 属性 【注：不可修改此属性，该属性只有在多视频模式下才有】 .....	44
ys_ip 属性 【注：不可修改此属性】 .....	44
ys_area 属性 【注：不可修改此属性】 .....	44
ys_carrier 属性 【注：不可修改此属性】 .....	44
5 SDK 派发的事件.....	44
加入房间成功 (YS.EVENT_TYPE.roomConnected) .....	44
加入失败 (YS.EVENT_TYPE.roomConnectFail) .....	44
房间连接中断 (YS.EVENT_TYPE.roomDisconnected) .....	44
离开房间 (YS.EVENT_TYPE.roomLeaveRoom) .....	45
用户加入房间 (YS.EVENT_TYPE.roomParticipantJoin) .....	45
用户离开房间 (YS.EVENT_TYPE.roomParticipantLeave) .....	45
用户被请出房间 (YS.EVENT_TYPE.roomParticipantEvicted) .....	45
收到聊天消息 (YS.EVENT_TYPE.roomTextMessage) .....	45
自定义消息发布 (YS.EVENT_TYPE.roomPubmsg) .....	46
自定义消息删除 (YS.EVENT_TYPE.roomDelmsg) .....	46
用户属性改变 (YS.EVENT_TYPE.roomUserPropertyChanged) .....	46
用户视频发布状态改变 (YS.EVENT_TYPE.roomUserVideoStateChanged) .....	46
用户音频发布状态改变 (YS.EVENT_TYPE.roomUserAudioStateChanged) .....	47
服务器地址更新 (YS.EVENT_TYPE.roomServerAddressUpdate) .....	47
用户共享媒体文件状态改变 (YS.EVENT_TYPE.roomUserMediaStateChanged) .....	48
房间音视频状态切换 (YS.EVENT_TYPE.roomAudioVideoStateSwitched) .....	48
媒体设备不可用 (YS.EVENT_TYPE.deviceNotAvailable) .....	48
媒体设备成功/失败 (YS.EVENT_TYPE.deviceSuccess/YS.EVENT_TYPE.deviceFailure) ..	49
房间模式改变 (YS.EVENT_TYPE.roomModeChanged) .....	49
用户共享媒体文件属性更新 (YS.EVENT_TYPE.roomUserMediaAttributesUpdate) .....	49

错误消息通知 (YS.EVENT_TYPE.roomErrorNotice) .....	50
INFO 消息通知 (YS.EVENT_TYPE.roomInfoNotice) .....	50
警告消息通知 (YS.EVENT_TYPE.roomWarnNotice) .....	50
用户网络状态改变 (YS.EVENT_TYPE.roomUserNetworkStateChanged) .....	51
文件列表 (YS.EVENT_TYPE.roomFiles) .....	52
添加文件 (YS.EVENT_TYPE.roomAddFile) .....	52
删除文件 (YS.EVENT_TYPE.roomDeleteFile) .....	53
更新文件 (YS.EVENT_TYPE.roomUpdateFile) .....	53
视频首帧 (YS.EVENT_TYPE.roomFirstVideoFrame) .....	53
音频首帧 (YS.EVENT_TYPE.roomFirstAudioFrame) .....	54
主摄像头改变 (YS.EVENT_TYPE.roomUserMainCameraChanged) .....	54
6 SDK 中的常量 .....	54
6.1 发送信息 .....	54
6.2 打印日志 .....	55
6.3 音视频发布状态 .....	55
6.4 服务器录制类型 .....	55
6.5 设备不可用原因 .....	56
6.6 视频显示模式 .....	57
6.7 房间模式 .....	57
6.8 房间角色 .....	57
6.9 网络状态质量 .....	57
6.10 媒体类型 .....	58
7 状态码参考 .....	58
7.1 错误码参考 .....	58
7.2 错误消息通知错误码参考 .....	61
7.3 INFO 消息通知状态码参考 .....	62
7.4 警告消息通知警告码参考 .....	62
8 补充说明 .....	62
8.1 聊天消息转发规则说明 .....	62
8.2 大房间（大规模房间）模式说明 .....	63
8.3 多视频模式说明 .....	63
9 FAQ .....	63

# YS SDK(JS)接口文档

## 1 集成 Room SDK

在您阅读此文档时，我们假定您已经具备了基础的前端应用开发经验，并能够理解相关基础概念。

### 1.1 环境准备

您需要一部可以连接到互联网的电脑，并且配置好前端开发环境以及本地服务器。

### 1.2 导入

向您的项目引入 `yssdk.js` 资源文件

## 2 重要概念

### 2.1 房间

使用本 SDK 之前，您需要使用 `webapi` 或访问管理系统页面来创建房间。使用获得的房间号即可与同房间用户进行音视频和其他数据的交换了。

如果不想创建房间，可以使用 `joinRoomEx` 方法加入自定义房间

### 2.2 用户

使用网页、客户端或手机 `app` 皆可进入房间，每一个接入房间的客户端实例，我们称为一个用户。

### 2.3 音视频

每个具有可用的音视频采集设备的用户都可以单独发布自己的音频数据，也可以同时发布音视频数据。

## 3 使用指南

### 3.1 进出房间

调用 `joinroom` 或者 `joinRoomEx` 接口加入房间。成功加入房间后，将收到事件 `YS.EVENT_TYPE.roomConnected`（加入房间成功事件），同时房间内的其他用户将会收到事件 `YS.EVENT_TYPE.roomParticipantJoin`（有用户加入房间事件）。

当需要离开房间时，您需要调用 `leaveroom` 接口。离开房间成功后，您将收到事件 `YS.EVENT_TYPE.roomLeaveRoom`（离开房间事件），同时房间内其他用户将收到事件 `YS.EVENT_TYPE.roomParticipantLeave`（有用户离开房间）。

当遇到网络闪断等异常情况时，您也会收到事件 `YS.EVENT_TYPE.roomDisconnected`（房间连接中断事件）。在这种情况下，您需要做好项目的相关清除工作。

### 3.3 用户

每一个连接到房间内的客户端，被称为一个用户。用户的身份由 `userId` 来唯一标识。当您在调用 `joinroom` 接口时，如果传入一个 `userId`，则本地用户的 `userId` 即为您设定的 `Id`。如果没有传，则 SDK 会自动生成一个 `guid` 作为您的 `userId`。但请注意，无论是哪种情况，系统不允许同一个房间内存在 `userId` 相等的两个用户。所以当您使用一个房间内已经存在的 `userId` 进入房间时，服务器会把之前使用这个 `id` 的用户请出房间。

除了 `userId` 外，每个用户还有一些其他属性（预定义的和自定义的），您可以使用 `changeUserProperty` 接口来修改任一用户的任一属性，并且可以指定把这项修改通知给哪个（些）用户。随后，您指定的用户就会收到一个事件 `YS.EVENT_TYPE.roomUserPropertyChanged`（用户属性改变），告知某个用户的某些属性被修改了。

### 3.4 音视频

您可以调用 `publishVideo` 和 `publishAudio` 接口来发布自己的音频或视频。一旦用户开始发布音、视频数据，房间内的所有用户都会收到事件和 `YS.EVENT_TYPE.roomUserAudioStateChanged`。要注意的是，您不需要发布本地用户的视频，都可以随时调用 `playVideo` 观看自己的视频图像。如果您不想观看某个用户的视频，可以通过调用 `unplayVideo` 结束观看。

您可以调用 `unpublishVideo` 和 `unpublishAudio` 接口来取消发布自己的音频或视频。一旦用户将自己的音视频都取消发布，房间内的所有用户都会收到事件

`YS.EVENT_TYPE.roomUserVideoStateChanged` 和

`YS.EVENT_TYPE.roomUserAudioStateChanged`，这时您需要调用 `unplayVideo` 和 `unplayAudio` 取消观看别人的视频和收听别人的音频。

### 3.5 自定义信令

利用自定义信令接口，您可以完成您设计的各种业务流程，创造各种激动人心的用户体验。我们把各种信令都抽象成了消息，和用户类似，消息有自己的 `id` (`id`) 和昵称 (`name`)，昵称可以和其他消息重名，但 `id` 必须在房间内唯一，`id` 相同的消息，后来的会替换之前的。除了这两个属性，消息还有这些属性：

**told:** 谁会接收到该消息的通知。可能取值请见 6.1 ‘发送信息’

**data:** 数据（可存放各种数据结构，包括数字、字符串、Json）

**save:** 是否保存。当为真时，服务器会在房间内保存该条消息，新进入的用户会收到该条消息。如果连续发两条 `id` 相同且需要保存的消息，服务器只会保存后一条。；当为假时，服务器在转发完这条消息后，即将此消息抛弃

具体流程如下：

调用 `pubMsg` 接口发布一条消息，则所有指定用户将收到事件 `YS.EVENT_TYPE.roomPubmsg`（`YS.EVENT_TYPE.roomPubmsg` 消息事件）。

如果这条消息是指定保存的，则之后新进入房间的用户会收到事件 `YS.EVENT_TYPE.roomConnected`(房间连接成功事件)的数据中带有当前服务器的所有保存的且未删除的消息数组。在发布消息之后，可以再次调用 `pubMsg` 接口，发布一条相同 `id` 的消息来覆盖它，则指定用户会再次收到事件 `YS.EVENT_TYPE.roomPubmsg`（`YS.EVENT_TYPE.roomPubmsg` 消息发布事件）。也可以调用 `delMsg` 接口来删除此消息，此时指定用户会收到事件 `YS.EVENT_TYPE.roomDelmsg`（`YS.EVENT_TYPE.roomDelmsg` 消息删除事件）。



## 4 接口说明

### 4.1 全局接口

获取 SDK 版本号 (getSdkVersion)

YS.getSdkVersion()

该方法获取 SDK 版本号，返回值为 String 类型。

### 4.2 房间管理类 (Room)

实例化 (Room)

Room()

该方法实例化一个房间管理类对象。房间中所有操作，几乎都由该对象控制。

示例代码：

```
var room = YS.Room();
```

初始化 (init)

init(authKey, onSuccess, onFailure, isCheckDevice, options)

该方法用于初始化 Room 对象

**注：**在 init 成功后才能调用 room 的其它方法。

**建议：**建议在 init 调用前建议先完成设备检测，因为调用 init 方法后会从设备列表中寻找一个设备（这个设备是否可用无法确保）

名称	类型	描述
authKey	String	请填写正确的企业 Key，形如：82AUScqguvqXzhUh
onSuccess	Function	成功后的回调
onFailure	Function	[非必选项]失败后的回调
isCheckDevice	Boolean	[非必选项]是否检测设备的可用性，默认 true
options	Object	[非必选项]初始化配置项，包含以下配置：

		<p><b>autoSubscribeAV:</b>表示当房间中有用户发布音视频后，是否由 SDK 自动订阅该用户的音视频数据，Boolean 类型，缺省为 <b>false</b>。</p> <p><b>isGetFileList:</b>是否获取文件列表，缺省为 <b>false</b>。</p> <p><b>ys_multistream:</b>是否开启多视频模式，类型为布尔类型，<b>true</b> 开启，<b>false</b> 不开启，缺省为 <b>false</b>。</p> <p><b>ys_max_reconnect_count:</b>发布音视频的最大重连次数，默认无限重连，<b>hasCluster:</b>是否有集群，缺省 <b>true</b></p> <p><b>【注：如果需要设置无限重连请不要配置此属性】</b></p>
--	--	---

## 取消初始化（uninit）

**uninit(onSuccess,onFailure)**

该方法用于重置 Room 对象到未初始化状态

**注：**调用该方法后，Room 对象的相关状态将被重置，设备的使用权限也会被释放，想要调用任何 room 方法前请重新调用 **init** 方法

名称	类型	描述
onSuccess	Function	成功后的回调
onFailure	Function	[非必选项]失败后的回调

## 检测是否初始化（checkInit）

**checkInit()**

该方法用于检测房间的初始化状态，返回布尔类型，**true** 表示已经初始化，**false** 表示没有初始化

## 加入房间（joinroom）

**joinroom(host, port, nickName, userId, roomJsonOptions, userJsonOptions)**

调用该方法加入已创建的房间。此方法应为 Room 类中首先被调用的方法。

名称	类型	描述
host	String	web 服务器域名（或 ip）

port	String	web 服务器端口号
nickName	String	用户在房间中的昵称
userId	String	用户在房间中的 ID，必须唯一。若为空，SDK 将自动生成
roomJsonOptions	Object	加入房间所需的基本参数。 <b>serial</b> : 房间 ID，String 类型； <b>password</b> : 房间密码，String 类型，不同用户角色对应不同密码
userJsonOptions	Object	[非必选项]用户自定义属性

示例代码：

```
var roomOpt = {
    serial : '888888',
    password : '1234'
};
room.joinroom('api.roadofcloud.com', '80', 'tester', "", roomOpt);
```

加入房间（joinRoomEx）

joinRoomEx(roomId, nickName, userId, userJsonOptions)

调用该方法加入已创建的房间。此方法应为 Room 类中首先被调用的方法。

名称	类型	描述
roomId	String	自定义房间号
nickName	String	用户在房间中的昵称
userId	String	用户在房间中的 ID，必须唯一。若为空，SDK 将自动生成
userJsonOptions	Object	[非必选项]用户自定义属性

示例代码：

```
var userJsonOptions= {
    avatar: 'https://www.baidu.com'
};
```

room.joinRoomEx('testRoomId', 'nickname', 'my\_userid', userJsonOptions);

## 离开房间 (leaveroom)

leaveroom()

使用该方法离开房间。离开房间后，所有音视频通话都被终止。

## 获取房间信息 (getRoomProperties)

getRoomProperties()

该方法获取房间信息。此信息中包括预约房间时的所有配置。

示例代码：

```
var prop = room.getRoomProperties();
```

```
var serial = prop.serial;
```

## 获取自己的用户对象 (getMySelf)

getMySelf()

该方法获取自己的用户对象。

注：如果没有 **joinroom** 成功（即没有收到 **YS.EVENT\_TYPE.roomConnected** 事件），则返回空对象

示例代码：

```
var self = room.getMySelf();
```

```
var myId = self.id;
```

## 获取指定用户对象 (getUser)

getUser(userId)

该方法获取指定 **RoomUser** 对象

注：该接口在大房间（即大规模房间）下使用只能获取到已经上台的用户（即发布音频/视频人的用户）和特殊用户（如：老师/助教），因此在大房间模式下，建议只使用该接口来操作上台的用户

名称	类型	描述
userId	String	用户 ID

## 获取房间中所有用户对象 (getUsers)

## getUsers()

该方法返回房间中所有用户对象,返回值类型为 **Map**。

注：该接口在大房间（即大规模房间）下使用只能获取到已经上台的用户（即发布音频/视频人的用户）和特殊用户（如：老师/助教），因此在大房间模式下，不建议使用该接口

## 获取本地设备管理类对象（getDeviceMgr）

### getDeviceMgr（）

该方法返回设备管理类对象

注：**DeviceMgr** 中的方法可由 **YS.DeviceMgr** 直接调用，并非一定要通过此接口获得 **DeviceMgr** 对象才能使用

示例代码：

```
var devMgr = room.getDeviceMgr();
```

## 发布本地视频（publishVideo）

### publishVideo(videoinputDeviceId)

该方法将自己的视频发布出去，房间中的人会收到 **YS.EVENT\_TYPE.roomUserVideoStateChanged** 事件，然后可通过调用 **playVideo** 方法观看发布者的视频。

注：如果是多视频模式调用该方法时需要传入设备的 ID。

名称	类型	描述
videoinputDeviceId	String	[多视频模式必选]视频设备 ID

## 停止发布本地视频（unpublishVideo）

### unpublishVideo(videoinputDeviceId)

该方法停止发布自己的视频，房间中的人会收到 **YS.EVENT\_TYPE.roomUserVideoStateChanged** 事件，然后可调用 **unplayVideo** 方法关闭播放器。

注：如果是多视频模式调用该方法时需要传入设备的 ID。

名称	类型	描述
videoinputDeviceId	String	[多视频模式必选]视频设备 ID

发布本地音频 (publishAudio)

publishAudio()

该方法将自己的音频发布出去，房间中的人会收到 `YS.EVENT_TYPE.roomUserAudioStateChanged` 事件，然后可通过调用 `playAudio` 方法观收听布者的音频。

停止发布本地音频 (unpublishAudio)

unpublishAudio()

该方法停止发布自己的音频，房间中的人会收到 `YS.EVENT_TYPE.roomUserAudioStateChanged` 事件，然后可调用 `unplayAudio` 方法关闭播放器。

播放视频 (playVideo)

playVideo(userId, elementId, options, onFailure , videoinputDeviceId )

该方法用于播放指定用户的视频。若播放其他用户视频必须在收到 `YS.EVENT_TYPE.roomUserVideoStateChanged` 事件后且视频为发布状态方可调用；若播放本地视频，则可直接调用。

名称	类型	描述
userId	String	指定的用户 ID。若为”，则播放自己视频
elementId	String	dom 元素 ID。该 dom 元素用于盛放播放器
options	Object	[非必选项]播放配置项。包含属性如下：loader 属性（Boolean 类型），用于表示未加载到视频数据时是否显示‘加载中’样式，缺省 true；mode 属性（Integer 类型），用于表示视频显示模式（是否裁减），可能取值请见 6.6 ，缺省 <code>YS_VIDEO_MODE.ASPECT_RATIO_CONTAIN</code>
onFailure	Function	[非必选项]失败回调
videoinputDeviceId	String	[多视频模式必选]视频设备 ID

示例代码：

```
room.playVideo('user-id-123', 'element-id-123', {}, function(err) {
```

```
YS.Logger.error('play video failed. error: ' + err);
});
```

### 停止播放视频（unplayVideo）

unplayVideo(userId, onFailure, videoinputDeviceId)

该方法停止播放指定用户的视频。

名称	类型	描述
userId	String	指定的用户 ID。若为”，则停止播放自己视频
onFailure	Function	[非必选项]失败回调
videoinputDeviceId	String	[多视频模式必选]视频设备 ID

示例代码：

```
room.unplayVideo('user-id-123', function(err) {
    YS.Logger.error('stop playing video failed. error: ' + err);
});
```

### 播放音频（playAudio）

playAudio(userId, onFailure)

该方法用于播放指定用户的音频。请在收到 `YS.EVENT_TYPE.roomUserAudioStateChanged` 事件后且音频为发布状态时调用该方法。

**注：用户不能播放自己的音频**

名称	类型	描述
userId	String	指定的用户 ID
onFailure	Function	[非必选项]失败回调

示例代码：

```
room.playAudio('user-id-123', function(err) {
    YS.Logger.error('play audio failed. error: ' + err);
});
```

### 停止播放音频（unplayAudio）

**unplayAudio(userId, onFailure)**

该方法停止播放指定用户的音频。

名称	类型	描述
userId	String	指定的用户 ID
onFailure	Function	[非必选项]失败回调

示例代码：

```
room.unplayAudio('user-id-123', {}, function(err) {  
    YS.Logger.error('stop playing audio failed. error: ' + err);  
});
```

**设置播放音量（setRemoteAudioVolume）**

**setRemoteAudioVolume(volume, userId, onFailure, type)**

该方法设置指定用户的音频播放音量。此方法在调用 **playAudio** 方法后方能生效。

名称	类型	描述
userId	String	指定的用户 ID
volume	Integer	播放音量值。取值区间[0,100]，初始值为 100
onFailure	Function	[非必选项]失败回调
type	String	[非必选项]需要操作媒体类型： YS.MEDIA_TYPE.DEVICE_AV， YS.MEDIA_TYPE.FILE， YS.MEDIA_TYPE.SCREEN， YS.MEDIA_TYPE.MEDIA， (缺省为 YS.MEDIA_TYPE.DEVICE_AV)

**获取音量级别（getAudioVolume）**

**getAudioVolume(userId, onVolume, onFailure, type)**

该方法获取指定用户的当前时刻音频采集音量（即麦克风输入音量），如果您想表示本地或远端音量的变化，我们建议您使用 **registerAudioVolumeListener** 接口来实时获取。



名称	类型	描述
userId	String	指定的用户 ID
onVolume	Function	获取音量回调。参数为当前时刻音量值。取值区间 [0,100]
onFailure	Function	[非必选项]失败回调
type	String	[非必选项]需要操作媒体类型： YS.MEDIA_TYPE.DEVICE_AV， YS.MEDIA_TYPE.FILE， YS.MEDIA_TYPE.SCREEN， YS.MEDIA_TYPE.MEDIA， (缺省为 YS.MEDIA_TYPE.DEVICE_AV)

示例代码：

```
room.getAudio(room.getMySelf().id, function(vol) {
    YS.Logger.debug('current volume: ' + vol);
});
```

注册音量监听器（registerAudioVolumeListener）

registerAudioVolumeListener(userId, msInterval, onVolume, onFailure, type)

该方法用于获取指定用户的音量变化，无需用户自己使用 `setTimeout` 或 `setInterval` 定时调用 `getAudioVolume` 方法，是 `getAudioVolume` 的升级方法。

名称	类型	描述
userId	String	指定的用户 ID。若为”，则监听自己的音量变化
msInterval	Integer	毫秒级时间间隔。最小值为 50
onVolume	Function	获取音量回调。取值区间[0,100]
onFailure	Function	[非必选项]失败回调
type	String    已有常量	[非必选项]需要操作媒体类型： YS.MEDIA_TYPE.DEVICE_AV，

		YS.MEDIA_TYPE.FILE, YS.MEDIA_TYPE.SCREEN, YS.MEDIA_TYPE.MEDIA, (缺省为 YS.MEDIA_TYPE.DEVICE_AV)
--	--	---

示例代码：

```
room.registerAudioVolumeListener (room.getMySelf().id, 80, function(vol, userId) {
    YS.Logger.debug('user: ' + userId + '. volume: ' + vol);
});
```

移除音量监听器（unregisterAudioVolumeListener）

unregisterAudioVolumeListener (userId, onFailure, type)

该方法移除指定用户的音量监听器。

名称	类型	描述
userId	String	指定的用户 ID。若为”，则移除自己的音量监听器
onFailure	Function	[非必选项]失败回调
type	String	[非必选项]需要操作媒体类型： YS.MEDIA_TYPE.DEVICE_AV， YS.MEDIA_TYPE.FILE， YS.MEDIA_TYPE.SCREEN， YS.MEDIA_TYPE.MEDIA， (缺省为 YS.MEDIA_TYPE.DEVICE_AV)

设置日志级别（setLogIsDebug）

setLogIsDebug(isDevelopment, logLevel, isFormatLog)

该方法设置日志级别。

名称	类型	描述
isDevelopment	Boolean	[非必选项]是否是开发日志，缺省为 false

logLevel	Integer	[非必选项]日志级别，可能取值请见 6.2。【注：当 logLevel 为 undefined 时,日志级别由 isDevelopment 参数决定,isDevelopment 为 true,日志级别为 debug, isDevelopment 为 false,日志级别为 info;当 logLevel 有定义时，日志级别由 logLevel 值决定。 】
isFormatLog	Boolean	[非必选项]是否格式化日志，缺省为 false【注：如果为 true，则将数组、对象等格式化为字符串输出】

示例代码：

```
room. setLogIsDebug(false, YS.LOG_INFO,false);
```

注册日志监听器（registerLogListener）

registerLogListener(logListener)

该方法用于监听打印的日志信息。

名称	类型	描述
logListener	Function	日志监听器函数,第 1 个参数 args 表示日志信息的数组，第 2 个参数 logLevel 表示日志级别（logLevel 的可能取值请见 6.2）

示例代码：

```
room. registerLogListener(function(args,logLevel){
    var logArr = ['ys sdk print log(log level is '+logLevel+')-->'];
    logArr = logArr.concat(args);
    console.log.apply(console, logArr);
});
```

比如：YS.Logger.info( 'loginfo\_01' , ' loginfo\_02' )则 args 为[ 'loginfo\_01' , ' loginfo\_02' ],logLevel 为 YS.LOG\_INFO

发送聊天消息（sendMessage）

sendMessage(textMessage, told , extendJson)

该方法发送聊天消息给指定用户

名称	类型	描述
textMessage	String	文本消息内容
told	String	[非必选项]消息接收者，缺省为所有人。可能取值请见 6.1 。接收者会收到 YS.EVENT_TYPE.roomTextMessage 事件
extendJson	Object	[非必选项]用户可自定义的扩展数据

示例代码：

```
var extenJson = {  
    textfont : 'Arial',  
    fontsize : 14  
};  
room.sendMessage('hello world', YS.MSG_TO_ALLUSER, extenJson);
```

发布自定义消息（pubMsg）

pubMsg（params）

该方法发布用户自定义消息。

parmas 参数包含如下属性

名称	类型	描述
name	String	消息名
id	String	消息 ID
told	String	消息接收者。可能取值请见 6.1。接收者会收到 YS.EVENT_TYPE.roomPubmsg 事件
data	Object	消息携带的数据信息
save	Boolean	[非必选项]消息是否保存到服务器，缺省为 <b>false</b> 。若不保存到服务器，晚于发布消息时刻进入房间的用户不会收到此消息，只有发布消息时刻前进入房间的用户可收到此消息；若保存到服务器，在该消息未被删

		<p>除的情况下，任何时刻进入房间的用户都会收到此消息。</p> <p>注：未调用删除消息接口（delMsg）情况下，若房间中所有用户已退出，此消息会被自动删除</p>
associatedMsgID	String	<p>[非必选项]关联的消息 ID。假设此消息关联了一条 msgId 为 ‘test-msg’ 的消息，那么当 ‘test-msg’ 消息被删除后，此消息也被删除</p>
associatedUserID	String	<p>[非必选项]关联的用户 ID。假设此消息关联了一个 id 为 ‘user-id-123’ 的用户，那么当 ‘user-id-123’ 用户离开房间后，此消息将被删除</p> <p>注：若关联了用户 ID，则该用户离开后，则该消息的接收者（即：发送 pubMsg 时指定的 toID）会收到 YS.EVENT_TYPE.roomDelmsg 事件</p>

示例代码：

```
var msgData = {
    textmsg : 'every body will receive the hello world message',
    textfont : 'Arial',
    fontsize : 14
};

var params = {
    msgName : 'sendTextMsg',
    msgId : 'sendTextMsg',
    toID : YS.MSG_TO_ALLEXCEPTSENDER,
    data : msgData,
    save : true,
};

room.pubMsg(params);
```

删除自定义消息（delMsg）

## delMsg(params)

该方法用于删除自定义消息

params 参数包含如下属性

名称	类型	描述
name	String	消息名
id	String	消息 ID。 注：删除消息的 ID 必须与 pubMsg 方法发布的消息 ID 对应，否则会失败
toID	String	消息接收者。可能取值请见 6.1。接收者会收到 YS.EVENT_TYPE.roomDelmsg 事件
data	Object	消息携带的数据信息 注：不建议携带数据

示例代码：

```
var
```

```
var params = {  
    msgName : 'sendTextMsg',  
    msgId : 'sendTextMsg',  
    toID : YS.MSG_TO_ALLEXCEPTSENDER,  
    data : {}  
};
```

```
room.delMsg(params);
```

## 改变用户属性（changeUserProperty）

### changeUserProperty(userId, told, properties)

该方法改变或添加某用户属性，并将属性变化通知给房间中的指定用户。

注：在大房间模式下，事件派发 YS.EVENT\_TYPE.roomUserPropertyChanged（用户属性改变）有限制，详细请见 YS.EVENT\_TYPE.roomUserPropertyChanged 事件派发

名称	类型	描述
----	----	----

userId	String	被改变属性的用户 ID
told	String	被通知的用户。可能取值请见 6.1。被通知的用户会收到 <b>YS.EVENT_TYPE.roomUserPropertyChanged</b> 事件
properties	Object	用户属性集合。

示例代码：

```
var self = room.getMySelf();
var prop = {};
if (self.nickname === 'tester') {
    prop.nickname : 'tester-abc';
}
if (!self. hasOwnProperty('age')) {
    prop.age = 18;
}
room.changeUserProperty(self.id, YS.MSG_TO_ALLUSER, prop);
```

改变指定角色的用户属性（changeUserPropertyByRole）

changeUserPropertyByRole(roles, told, properties)

该方法改变或添加某些角色的用户属性，并将属性变化通知给房间中的指定用户。

注：在大房间模式下，事件派发 **YS.EVENT\_TYPE.roomUserPropertyChanged**（用户属性改变）有限制，详细请见 **YS.EVENT\_TYPE.roomUserPropertyChanged** 事件派发。

名称	类型	描述
roles	Array	被改变属性的角色列表，可能取值请见 6.8
told	String	被通知的用户。可能取值请见 6.1。被通知的用户会收到 <b>YS.EVENT_TYPE.roomUserPropertyChanged</b> 事件
properties	Object	用户属性集合。

示例代码：

```
var roles=[YS.ROOM_ROLE.STUDENT, YS.ROOM_ROLE.ASSISTANT] ;

var prop = {
    age:20
};

room.changeUserPropertyByRole(roles, YS.MSG_TO_ALLUSER, prop);
```

### 批量改变用户属性（batchChangeUserProperty）

batchChangeUserProperty(ids, told, properties)

该方法批量改变或添加某用户属性，并将属性变化通知给房间中的指定用户。

注：在大房间模式下，事件派发 **YS.EVENT\_TYPE.roomUserPropertyChanged**（用户属性改变）有限制，详细请见 **YS.EVENT\_TYPE.roomUserPropertyChanged** 事件派发。

名称	类型	描述
ids	Array	被改变属性的用户 ID 列表
told	String	被通知的用户。可能取值请见 6.1。被通知的用户会收到 YS.EVENT_TYPE.roomUserPropertyChanged 事件
properties	Object	用户属性集合。

示例代码：

```
var ids=[ user1.id, user2.id, user3.id ] ;

var prop = {
    age:20
};

room.batchChangeUserProperty(ids, YS.MSG_TO_ALLUSER, prop);
```

### 将用户请出房间（evictUser）

evictUser(userId, causeJson)

该方法用于将用户请出房间

名称	类型	描述
----	----	----



userId	String	被请出房间的用户 ID。该用户会收到 YS.EVENT_TYPE.roomParticipantEvicted 事件
causeJson	Object	请出房间的原因。用户自定义

```
var cause = {
    reason : 'never want to see you',
    say : 'good-bye'
};
room.evictUser('user-id-123', cause);
```

## 设置是否自动处理设备改变（插拔）事件（setAutoProcessDeviceChangeEvent）

### setAutoProcessDeviceChangeEvent(autoProcess)

该方法设置 SDK 是否自动处理设备改变（插拔）事件。若设定为自动处理，在单视频模式下，设备插拔后 SDK 会自动选择设备重新发布本地音视频（若当前正在发布音视频），在多视频模式下会自动将移除的设备取消发布（即：主动执行 `unpublishVideo`，但是不主动 `unplayVideo`）。若设定为非自动处理，则设备拔出后，视频画面会停留在最后一帧。不主动指定为自动处理的情况下，SDK 不会自动处理。

名称	类型	描述
autoProcess	Boolean	指定是否自动处理设备改变（插拔）事件

备注：在多视频模式下，如果移除的是主摄像头设备，则 SDK 会自动找视频设备列表的第一个设备自动切换主摄像头。

## 获取视频属性（getVideoProfile）

### getVideoProfile(videoinputDeviceId)

该方法获取当前视频属性

名称	类型	描述
videoinputDeviceId	String	视频设备 ID

返回值为最终获取结果！

## 设置视频属性（setVideoProfile）

### setVideoProfile(profile, onFailure,videoinputDeviceId)

该方法设置视频属性。注：设置的分辨率及帧率不可超过预约房间时设定的最大分辨率的乘积（width\*height）及最大帧率，若分辨率超出，则根据比例缩放当前设置的分辨率，若帧率超出，则设置为预约房间时的最大分辨率。

名称	类型	描述
profile	Object	视频属性。包含如下属性，width：水平分辨率（Integer）；height：垂直分辨率（Integer）；maxfps：最大帧率（Integer）
onFailure	Function	[非必选项]失败回调
videoinputDeviceId	String	[多视频模式必选]视频设备 ID

示例代码：

```
var profile = {  
    width : 80,  
    height : 60,  
    maxfps : 10  
};  
_room.setVideoProfile(profile);
```

### 请求服务器列表（requestServerList）

#### requestServerList(host,port,onSuccess,onFailure)

该方法用于获取可用服务器信息，用户可根据自己实际位置选择不同服务器，以提升房间会话体验。

名称	类型	描述
host	String	服务器域名
port	Int	服务器端口
onSuccess	Function	成功回调，回调参数包含请求所得服务器列表
onFailure	Function	[非必选项]失败回调

### 切换服务器（switchServer）

## switchServer(serverName)

该方法用于将当前使用的服务器切换为 **serverName** 指定的服务器。由于服务器地理位置不同，可能造成使用不同服务器时，在房间中的会话体验不同，请用户根据自己的情况，合理选择。

名称	类型	描述
serverName	String	服务器名称。通过 requestServerList 接口获得

## 切换房间音视频状态（switchOnlyAudioRoom）

### switchOnlyAudioRoom(isSwitch)

该方法可切换房间中所有用户的音视频状态。可将音视频并存的房间变成只有音频的房间，此时用户均无法发布和观看视频，只能发布和收听音频；也可将只有音频的房间变成音视频可并存的房间，此时用户可以发布音频视频以及观看收听音视频。切换后，所有用户均会收到 YS.EVENT\_TYPE.roomAudioVideoStateSwitched 事件。

名称	类型	描述
isSwitch	Boolean	是否将房间切换为只有音频的房间。 <b>true</b> : 切换为只有音频的房间； <b>false</b> : 切换为音视频可并存的房间

## 服务器录制（start/stopServerRecord）

### 1、startServerRecord(spec)

该方法开始服务器端录制。参数 **spec** 是可选参数，包含如下属性

名称	类型	描述
recordType	String	[可选]录制类型。可能取值请见 6.4（服务器录制类型）。缺省为 YS.REC_TYPE_VIDEOLIST
convert	Integer	[可选]是否转换。0: 不转换（mkv 格式）；1: 转为 webm 格式；2: 转为 mp4 格式。缺省为 0 <b>注：该属性只在 recordType 为 YS.REC_TYPE_VIDEOLIST 时生效</b>
layout	Integer	recordType 为 YS.REC_TYPE_MIXVIDEO 时视频排列方式。0: 横向排列；1: 纵向排列。缺省为 0

		注：该属性值只在 <b>recordType</b> 为 <b>YS.REC_TYPE_MIXVIDEO</b> 时生效
<b>recordChat</b>	<b>Boolean</b>	[可选]是否录制聊天消息，缺省为 <b>true</b>

注：连续调用两次 **startServerRecord**（即没调用 **stopServerRecord** 就再次调用 **startServerRecord**）将以第 1 次调用为准

## 2、stopServerRecord()

该方法停止服务器端录制。

## 播放共享的媒体文件（play/unplayRemoteMedia）

### 1、playRemoteMedia(userId, elementId, options, onFailure)

该方法播放房间中共享的媒体文件。在收到 **YS.EVENT\_TYPE.roomUserMediaStateChanged** 事件后，若开始共享媒体文件，则调用此方法观看

名称	类型	描述
<b>userId</b>	<b>String</b>	指定的用户 ID
<b>elementId</b>	<b>String</b>	Dom 元素 ID。该 Dom 元素用于盛放播放器，一般应为 Div
<b>options</b>	<b>Object</b>	[非必选项]播放配置项。包含属性如下： <b>loader</b> 属性（ <b>Boolean</b> 类型），用于表示未加载到视频数据时是否显示‘加载中’样式，缺省 <b>true</b>
<b>onFailure</b>	<b>Function</b>	[非必选项]失败回调

### 2、unplayRemoteMedia(userId, onFailure)

该方法停止播放共享的媒体文件。停止共享媒体文件后需要调用此方法释放播放器资源。

名称	类型	描述
<b>userId</b>	<b>String</b>	指定的用户 ID
<b>onFailure</b>	<b>Function</b>	[非必选项]失败回调

## 获取房间的用户列表（getRoomUsers）

getRoomUsers(onSuccess, start, max ,roles , onFailure, search, order)

该方法获取用户列表。

注：该接口建议只在大房间（即大规模房间）下使用，非大房间模式下建议使用 **getUsers** 接口

名称	类型	描述
onSuccess	Function	成功回调,回调参数返回满足条件用户列表数组和总用户数【注：总人数受 <b>roles</b> 和 <b>search</b> 影响，原理同接口 <b>getRoomUserNum</b> 一样】
start	Int	[非必选项]开始位置，缺省为 0
max	Int	[非必选项]最大获取个数,缺省为 200
roles	Array	[非必选项]角色列表，缺省为所有角色，角色取值请见 6.8
onFailure	Function	[非必选项]失败回调
search	String	[非必选项]昵称检索的内容，缺省为”【注：此处只对于用户昵称进行搜索且为模糊搜索】
order	Array	[非必选项]排序规则，最多 3 条规则，缺省根据加入的时间进行升序（即：order 为 <code>[[{ts:'asc'}]]</code> ）。 例如：先根据角色进行升序，其次根据加入时间进行升序，则 order 为 <code>[[{"role":'asc'},{"ts":'asc'}]</code> 注意事项如下： 1、最多 3 条排序规则，超过的规则将不起效 2、规则在前面的优先起效，即 order 数组下标越小排序优先级越高 3、升序使用 'asc',降序使用 'desc'

获取房间的用户总数（getRoomUserNum）

getRoomUserNum(onSuccess, roles , onFailure, search)

该方法获取用户总数。

注：该接口建议只在大房间（即大规模房间）下使用，非大房间模式下建议使用 **getUsers** 接口取到列

表后根据业务需求统计总数

名称	类型	描述
onSuccess	Function	成功回调
roles	Array	[非必选项]角色列表，缺省为所有角色，角色取值请见 6.8
onFailure	Function	[非必选项]失败回调
search	String	[非必选项]昵称检索的内容，缺省为”【注：此处只对于用户昵称进行搜索且为模糊搜索】

### 共享媒体文件（start/stop/pauseShareMedia ...）

共享媒体文件是指将已上传到服务器的媒体文件（音频文件或视频文件）共享给房间内的人，共享成功（或停止）以后，房间内的人会收到 **YS.EVENT\_TYPE.roomUserMediaStateChanged** 事件，然后通过调用 **playRemoteMedia** 或 **unplayRemoteMedia** 来观看或停止观看共享的媒体文件。

注：共享的媒体文件必须是已经上传服务器的，不能共享非服务器上的媒体文件，因为该媒体文件是由服务器进行共享，因此自己收到 **YS.EVENT\_TYPE.roomUserMediaStateChanged** 后也需要调用 **playRemoteMedia** 或 **unplayRemoteMedia**

#### 1、startShareMedia(url, isVideo, onFailure)

该方法开始共享媒体文件。

名称	类型	描述
url	String	共享的媒体文件的绝对路径
isVideo	Boolean	[非必选项]是否是视频文件，缺省为 <b>false</b>
onFailure	Function	[非必选项]失败回调

#### 2、stopShareMedia(onFailure)

该方法停止共享媒体文件

名称	类型	描述
onFailure	Function	[非必选项]失败回调

#### 3、pauseShareMedia(bPause, onFailure)

该方法暂停共享媒体文件

名称	类型	描述
bPause	Boolean	是否暂停（或恢复）共享媒体文件。 <b>true</b> 表示暂停共享， <b>false</b> 表示继续共享
onFailure	Function	[非必选项]失败回调

#### 4、seekMedia(positionPercent, onFailure)

该方法控制共享媒体文件的进度，使媒体文件跳跃到指定的时间点继续播放。

名称	类型	描述
positionPercent	Number	指定进度的百分比，取值范围为[0,1]。例如文件总时长为 5000 毫秒，想要跳跃到 2000 毫秒处继续播放，则 positionPercent 值为 0.4，即（2000/5000）
onFailure	Function	[非必选项]失败回调

#### 获取文件列表（getFileList）

##### getFileList()

该方法获取文件列表，返回文件信息数组。

#### 获取文件信息（getFileinfo）

##### getFileinfo(fileid)

该方法获取文件信息，返回文件信息。

注：根据文件 **id** 从文件列表中获取文件信息，如果文件列表没有对应的文件，则返回 **undefined**

名称	类型	描述
fileid	Number	文件 id

#### 设置本地视频的镜像（setLocalVideoMirror）

##### setLocalVideoMirror(isMirror, videoinputDeviceId)

该方法设置本地视频是否启用视频镜像。

注：默认房间不启用本地视频镜像

名称	类型	描述
----	----	----

isMirror	Boolean	本地视频是否启用视频镜像
videoinputDeviceId	String	[多视频模式必选]视频设备 ID

### 上传文件 (uploadRoomFile)

uploadRoomFile(filename, filetype, filedata, onSuccess, onProgress, onFailure, options)

该方法用于上传文件。

名称	类型	描述
filename	String	文件名
filetype	String	文件类型
filedata	File	文件数据
onSuccess	Function	成功回调，第一个参数为 <b>code</b> ，表示上传的状态码（状态码为 0），第二个参数 <b>response</b> 表示上传成功服务器返回的数据
onProgress	Function	[非必选项]上传进度回调，第一个参数为 <b>evt</b> ，表示上传进度的 <b>event</b> 对象,第二个参数 <b>progress</b> 表示上传的百分比（0-100）
onFailure	Function	[非必选项]上传失败的回调，第一个参数为 <b>code</b> ，表示上传失败的错误码，详情请参考 7.1
options	Object	[非必选项]配置项，配置项如下： <b>isWritedb</b> :是否写入数据库，缺省为 <b>false</b>

### 删除文件 (deleteRoomFile)

deleteRoomFile(fileid, onSuccess, onFailure)

该方法用于删除文件。

名称	类型	描述
fileid	Integer	文件 id
onSuccess	Function	成功回调，第一个参数为 <b>code</b> ，表删除文件的状态



		码（状态码为 0）,第二个参数 <b>response</b> 表示删除文件成功服务器返回的数据
<b>onFailure</b>	<b>Function</b>	[非必选项]上传失败的回调，第一个参数为 <b>code</b> ，表示删除文件失败的错误码，详情请参考 7.1

### 获取内网 IPv4（getIp）

#### getIp(onSuccess, onFailure)

该方法用于获取内网 IP

名称	类型	描述
<b>onSuccess</b>	<b>Function</b>	成功回调，第一个参数为获取到的 ip 地址
<b>onFailure</b>	<b>Function</b>	[非必选项]获取失败的回调，第一个参数为 <b>error info</b>

### 获取指定人选的网络状态（getStreamStats）

#### getStreamStats（userId,callback）

该方法用于获取指定用户的网络状态

名称	类型	描述
<b>userId</b>	<b>String</b>	指定用户的 id
<b>callback</b>	<b>Function</b>	用于接受回调数据的函数，回调参数包含指定用户网络状态，该回调可能会执行多次

返回数据示例：

```
{
  audio: {audioMute: 0, bitsPerSecond: 41004, fractionLost: 7, jitter: 173, lossRate: 4, ...},
  deviceId: "aceb65e71f4c80cdfca68481c8ed057a6da69e0e0337063c2be7292bf6f331ac",
  timestamp: 1542796210141,
  type:"down",
  userId: "67fc7737-74e4-ce8a-7c19-a498d33651a4",
  video: {NACK: 2, bandwidth: 707476, bitsPerSecond: 256021, fractionLost: 0, googPlisCount: 2, ...},
  peerId: "af1eadaa-c211-4a7f-b829-78f959467816"
}
```

注：peerId 为调用该方法时传入的第一个参数，userId 代表该媒体链路属于的用户，deviceId 只有在

多流模式下才有值

### 4.3 事件分发器（EventDispatcher）

#### 实例化（EventDispatcher）

通过 `YS.EventDispatcher()` 方式实例化 `EventDispatcher` 对象

示例代码：

```
var dispatcher = YS. EventDispatcher();
```

#### 分发事件（dispatchEvent）

`dispatchEvent(event, log)`

该方法将某一事件分发给所有监听者

名称	类型	描述
event	Object	事件描述。其中必须包括 <b>type</b> 属性（ <b>String</b> 类型），表示事件类型。其他属性可自行定义添加。监听者会收到整个 <b>event</b> 对象，按封装格式反解，即可得到事件携带的数据
log	Boolean	[非必选项]是否打印日志（ <b>debug</b> 级别），缺省为 <b>true</b>

示例代码：

```
var event = {  
    type : 'dispatch-test',  
    data : 'this is event data'  
};  
dispatcher. dispatchEvent(event);
```

#### 添加事件监听者（addEventListener）

`addEventListener(eventType, listener)`

该方法将某一类型事件的监听者（回调函数）添加到事件分发器中，当事件触发后，事件分发器会通知（回调）该监听者

名称	类型	描述
eventType	String	事件类型。对应 <code>dispatchEvent</code> 方法分发的事件中的 <code>type</code> 属性
listener	Function	事件监听者。分发事件后，会触发该函数

示例代码：

```
dispatcher.addEventListener('dispatch-test', function(event) {
    YS.Logger.debug(event.data);
});
```

移除事件监听者（`removeEventListener`）

`removeEventListener (eventType, listener)`

该方法将某一类型事件的监听者移除，移除后，分发事件不会再通知该监听者

名称	类型	描述
eventType	String	事件类型。对应 <code>dispatchEvent</code> 方法分发的事件中的 <code>type</code> 属性
listener	Function	事件监听者。移除后，不会触发该函数

移除所有事件监听者（`removeAllEventListener`）

`removeAllEventListener (eventTypeArr)`

该方法移除指定事件的所有监听者

名称	类型	描述
eventTypeArr	Array	事件类型数组。

## 4.4 日志类(Logger)

debug 日志（`Logger.debug`）

该方法打印 `debug` 级别日志，与 `console.debug` 使用方式相同

示例代码：

YS.Logger.debug('this is debug log');

#### trace 日志 (Logger. trace)

该方法打印 **trace** 级别日志，与 **console.trace** 使用方式相同

示例代码：

YS.Logger.trace('this is trace log');

#### info 日志 (Logger. info)

该方法打印 **info** 级别日志，与 **console.info** 使用方式相同

示例代码：

YS.Logger.info ('this is info log');

#### warning 日志 (Logger. warning)

该方法打印 **warning** 级别日志，与 **console.warning** 使用方式相同

示例代码：

YS.Logger.warning ('this is warning log');

#### error 日志 (Logger. error)

该方法打印 **error** 级别日志，与 **console.error** 使用方式相同

示例代码：

YS.Logger.error ('this is error log');

## 4.5 本地设备管理类 (DeviceMgr)

### 枚举系统设备 (getDevices)

getDevices(callback);

该方法枚举系统中的摄像头、麦克风及扬声器设备。

名称	类型	描述
callback	Function	方法调用成功或失败回调

该方法如果调用成功，会返回包含系统设备信息的对象 **devices** 及系统当前正在使用的设备信息对象

useDevices。如果调用失败，devices 及 useDevices 对象中设备信息为空。

devices 对象中包含摄像头（videoinput）、麦克风（audioinput）和扬声器（audiooutput）信息对象，其中每个设备对象都包含 3 个属性：

属性	类型	描述
deviceId	String	该设备所独有的设备 ID
label	String	用于区分设备的标识。如果用户没有打开摄像头和麦克风的权限，label 会以特定名称加索引方式命名
kind	String	设备类型，根据选用的音视频设备，有 videoinput、audioinput 和 audiooutput 三种类型

userDevices 对象中包含系统正在使用的摄像头（videoinput）、麦克风（audioinput）和扬声器（audiooutput）的设备 ID。

示例代码：

```
YS.DeviceMgr.getDevices(function (deviceInfo) {  
    var devices = deviceInfo.devices;  
    var usingDevices = deviceInfo.useDevices;  
  
    var audioInputDevArr = devices.audioinput;  
    var audioInputDevCount = audioInputDevArr.length;  
    var audioInputDevId = audioInputDevCount[0].deviceId;  
  
    var usingAudioInputId = usingDevices.audioinput;  
}
```

枚举视频设备（getCameras）

```
getCameras(callback);
```

该方法枚举系统中的摄像头。

名称	类型	描述
Callback([])	Function	方法调用成功或失败回调

该方法如果调用成功，会返回摄像头列表

## 枚举音频输入设备（getMicrophones）

`getMicrophones(callback);`

该方法枚举系统中的摄像头。

名称	类型	描述
Callback([])	Function	方法调用成功或失败回调

该方法如果调用成功，会返回麦克风列表

## 枚举音频输出设备（getSpeakers）

`getSpeakers(callback);`

该方法枚举系统中的摄像头。

名称	类型	描述
Callback([])	Function	方法调用成功或失败回调

该方法如果调用成功，会返回扬声器列表

## 设置系统设备（setDevices）

`setDevices(deviceIdMap, onFailure)`

该方法设置系统使用的摄像头、麦克风及扬声器设备。

名称	类型	描述
deviceIdMap	Object	包含设备 ID 的对象。设备 ID 通过 <code>getDevices</code> 方法获取
onFailure	Function	[非必选项]方法调用失败回调

示例代码：

```
function setLocalDevices(videoInputId, audioInputId, audioOutputId) {  
    var deviceIdMap = {  
        videoinput : videoInputId,  
        audioinput : audioInputId,  
        audiooutput : audioOutputId
```

```

    });
    YS.DeviceMgr.setDevices(deviceIdMap, function() {
        console.log("set devices failed");
    });
}

```

注：setDevices 设置的 videoinput 在多视频模式下为设置主摄像头,设置主摄像头就是将音频关联到主摄像头上,摄像头在多视频模式下不存在切换问题,因此该接口切换的 videoinput 是切换主摄像头(即：切换音频关联的摄像头)

## 设置摄像头设备（setCamera）

setCamera(deviceId, onFailure)

该方法设置系统使用的摄像头设备。

名称	类型	描述
deviceId	String	设备 ID
onFailure	Function	[非必选项]方法调用失败回调

注：多视频模式下为设置主摄像头,设置主摄像头就是将音频关联到主摄像头上,摄像头在多视频模式下不存在切换问题,因此该接口是切换主摄像头（即：切换音频关联的摄像头）

## 设置麦克风设备（setMicrophone）

setMicrophone(deviceId, onFailure)

该方法设置系统使用的麦克风设备。

名称	类型	描述
deviceId	String	设备 ID
onFailure	Function	[非必选项]方法调用失败回调

## 设置扬声器设备（setSpeaker）

setSpeaker(deviceId, onFailure)

该方法设置系统使用的扬声器设备。

名称	类型	描述
deviceId	String	设备 ID
onFailure	Function	[非必选项]方法调用失败回调

## 检测是否支持设置扬声器（checkIsSupportSetSpeaker）

### checkIsSupportSetSpeaker()

该方法检测是否支持设置扬声器，如果返回值为 **true** 则支持，为 **false** 则不支持。

## 摄像头检测（start/stopVideoTest）

### 1、startVideoTest(deviceId, elementId, options, onSuccess, onFailure)

该方法开始检测摄像头是否可用。

注：若当前有摄像头正在检测，调用 **startVideoTest** 会自动停止正在进行的检测操作，但若要停止最后一次摄像头检测，必须调用 **stopVideoTest** 方法。

名称	类型	描述
deviceId	String	要检测设备的 ID。设备 ID 通过 <b>getDevices</b> 方法获取
elementId	String	Dom 元素 ID。该 Dom 元素用于盛放摄像头对应的视频窗口，一般应为 Div
options	Object	[非必选项]播放配置项。包含属性如下： <b>mirror</b> 属性（Boolean 类型），用于表示是否以镜面方式显示视频，缺省 <b>false</b> ； <b>loader</b> 属性（Boolean 类型），用于表示未加载到视频数据时是否显示‘加载中’样式，缺省 <b>true</b> ； <b>mode</b> 属性（Integer 类型），用于表示视频显示模式（是否裁减），可能取值请见 6.6，缺省 <b>YS_VIDEO_MODE.ASPECT_RATIO_CONTAIN</b>
onSuccess	Function	[非必选项]【新增】视频 <b>canplay</b> 之后的回调函数
onFailure	Function	[非必选项]方法调用/视频失败回调

示例代码：

```
function startVideoTest(videoInputId) {
```



```

var container = document.createElement('div');

container.id = 'container';

document.body.appendChild(container);

YS.DeviceMgr.startVideoTest(videoInputId, container.id);
}

```

### 3、stopVideoTest()

名称	类型	描述
deviceId	String	[非必选项]要关闭检测设备的 ID 对应开始检测时的设备 ID，如不传默认关闭所有指定 deviceId 调用 startVideoTest 的检测。

该方法停止摄像头检测。

注：为避免引起设备占用问题，等一系列进入教室内可能出现的设备问题，需及时关闭检测。

### 麦克风检测（start/stopMicrophoneTest）

#### 1、startMicrophoneTest(deviceId, elementId, onSuccess, onFailure)

该方法开始检测麦克风是否可用。

注：若当前有麦克风正在检测，调用 startMicrophoneTest 会自动停止正在进行的检测操作，但若要停止最后一次麦克风检测，必须调用 stopMicrophoneTest 方法。

名称	类型	描述
deviceId	String	要检测设备的 ID。设备 ID 通过 getDevices 方法获取
elementId	String	Dom 元素 ID。该 Dom 元素用于盛放播放麦克风声音的播放器，一般应为 Div
onSuccess	Function	[非必选项]方法调用成功回调，携带麦克风当前采集到的音量值
onFailure	Function	[非必选项]方法调用失败回调

示例代码：

```

function startMicrophoneTest(audioInputId) {
    var container = document.createElement('div');

```

```

        container.id = 'container';

        document.body.appendChild(container);

        var onSuccess = function(vol) {
            console.log("microphone volume: ", vol);
        }

        var onFailure = function(err) {
            console.log("microphone test failed. ", err);
        }

        YS.DeviceMgr.startMicrophoneTest(audioInputId, container.id, onSuccess, onFailure);
    }

```

## 2、stopMicrophoneTest()

该方法停止麦克风检测。

## 扬声器检测（start/stopSpeakerTest）

### 1、startSpeakerTest(deviceId, audioUrl, onFailure)

该方法开始检测扬声器是否可用。

注：若当前有扬声器正在检测，调用 **startSpeakerTest** 会自动停止正在进行的检测操作，但若若要停止最后一次麦克风检测，必须调用 **stopSpeakerTest** 方法。

名称	类型	描述
deviceId	String	要检测设备的 ID。设备 ID 通过 <b>getDevices</b> 方法获取
audioUrl	String	声音文件的绝对地址，建议使用 <b>wav</b> 格式的文件
onFailure	Function	[非必选项]方法调用失败回调

示例代码：

```

YS.DeviceMgr.startSpeakerTest(audioInputId, audioUrl, function(err){
    console.log("speaker test failed. ", err);
});

```

### 2、stopSpeakerTest()

该方法停止扬声器检测。

网络质量检测（start/stopNetworkTest）

1、startNetworkTest(onSuccess, onFailure)

该方法开始网络质量检测。

注：该接口只能用于加入房间(即调用 joinroom 接口)前进行网络质量检测，加入房间后该接口失效，加入房间后的网络质量请通过监听事件 YS.EVENT\_TYPE.roomUserNetworkStateChanged 实现。

名称	类型	描述
onSuccess	Function	成功回调。回调的第一个参数表示网络质量（可能取值请见 6.9），第二个参数表示延时（单位为 ms）
onFailure	Function	[非必选项]方法调用失败回调

示例代码：

```
YS.DeviceMgr.startNetworkTest(function(delayLevel,currentDelay){
    //实现业务逻辑的代码
});
```

2、stopNetworkTest()

该方法停止网络质量检测。

注册设备监听器（registerDeviceChangeListener）

registerDeviceChangeListener(listener)

该方法添加监听设备改变（插拔）的监听器。

注：只能有一个监听器，若重复添加，之前添加的监听器会失效。

名称	类型	描述
listener(changed, type)	Function	回调函数，当设备变化时会回调该函数。回调参数为 {videoinput:[{kind:'videoinput',label:'视频设备 label',deviceId:'视频设备 id'}]}。 注：经测试，极个别设备的行为异常，在未发生插拔设备的操作时，也会不断触发回调函数。

示例代码：

```
YS.DeviceMgr.registerDeviceChangeListener(function(changed, type) {
```

```
    if (type == YS.DEVICE_STATE.DEVICE_ADD)
    {
        else if (type == YS.DEVICE_STATE.DEVICE_REMOVE)
        {
    }
});
```

移除设备监听器

```
YS.DeviceMgr.registerDeviceChangeListener(null);
```

## 4.6 用户类 (RoomUser)

用户类没有预定义方法，只有如下预定义属性。用户可通过调用 `changeUserProperty` 方法向此类中添加属性。若只想添加本地属性，不想告诉房间中其他人，可直接为 `RoomUser` 对象添加属性并赋值，无需调用 `changeUserProperty` 方法。

### id 属性

用户在房间中的唯一标识。`String` 类型。可在调用 `joinroom` 接口时指定，也可由 SDK 自动生成。同一房间中只能存在 id 相同的一个用户。【注：不可修改此属性】

### nickname 属性

用户在房间中的昵称。`String` 类型。在 `joinroom` 时指定。

### role 属性

用户角色。`Integer` 类型。此属性由 `joinroom` 时传递的房间密码决定。不同的房间密码对应不同用户角色。

### hasaudio 属性 【注：不可修改此属性】

用户是否有音频设备。`Boolean` 类型。此属性初始值根据系统是否有音频设备决定。

### hasvideo 属性 【注：不可修改此属性】

用户是否有视频设备。`Boolean` 类型。此属性初始值根据系统是否有视频设备决定。

### **publishstate 属性 【注：不可修改此属性】**

用户音视频发布状态。Integer 类型。具体可能取值请见 6.3（音视频发布状态）。当用户发布音视频或取消发布音视频，此属性会在 SDK 中自动改变，用户不要主动修改此属性，请仅作为发布状态判断依据即可。

### **ys\_maincamera 属性 【注：不可修改此属性，该属性只有在多视频模式下才有】**

主摄像头设备 id。String 类型。此属性值根据接口 setDevices 的 videoinput 指定。

### **ys\_ip 属性 【注：不可修改此属性】**

用户的 ip 地址。String 类型。此属性值由 SDK 获取用户的 ip。

### **ys\_area 属性 【注：不可修改此属性】**

用户的所属区域。String 类型。此属性值由 SDK 获取用户的所属区域。

### **ys\_carrier 属性 【注：不可修改此属性】**

用户的所属运营商。String 类型。此属性值由 SDK 获取用户的所属运营商。

## **5 SDK 派发的事件**

### **加入房间成功（YS.EVENT\_TYPE.roomConnected）**

返回的数据：{type: YS.EVENT\_TYPE.roomConnected, message:msgArray}

msgArray: 房间当前存在的信令（pubMsg 发布的消息）数组， Array

### **加入失败（YS.EVENT\_TYPE.roomConnectFail）**

返回的数据：{type:YS.EVENT\_TYPE.roomConnectFail,message:{source:source, error:result } }

source:加入房间操作进行到哪一步

error: 错误原因，状态码可能值请见 7.1(错误码参考)

### **房间连接中断（YS.EVENT\_TYPE.roomDisconnected）**

返回的数据: {type:YS.EVENT\_TYPE.roomDisconnected}

## 离开房间 (YS.EVENT\_TYPE.roomLeaveRoom)

返回的数据: {type:YS.EVENT\_TYPE.roomLeaveRoom}

## 用户加入房间 (YS.EVENT\_TYPE.roomParticipantJoin)

返回的数据: {type:YS.EVENT\_TYPE.roomParticipantJoin, user:user, userid:userid}

user:用户对象 , Json

userid: 用户 id , String

## 用户离开房间 (YS.EVENT\_TYPE.roomParticipantLeave)

返回的数据: {type:YS.EVENT\_TYPE.roomParticipantLeave, user:user, userid:userid}

user:用户对象 , Json

userid: 用户 id , String

## 用户被请出房间 (YS.EVENT\_TYPE.roomParticipantEvicted)

返回的数据: {type:YS.EVENT\_TYPE.roomParticipantEvicted ,

user:user, userid:userid , message:evictedCauseJson}

user:用户对象 , Json

userid: 用户 id , String

evictedCauseJson: 被请出房间的原因 (调用 evictUser 携带的数据) , 如果是服务器主动踢的则没有 evictedCauseJson (也就是 undefined) , Json

## 收到聊天消息 (YS.EVENT\_TYPE.roomTextMessage)

返回的数据: {type:YS.EVENT\_TYPE.roomTextMessage, userid:userid,nickname:nickname,role:role, message:textData}

userid: 用户 id , String

nickname:用户昵称,String

role:用户角色, 可能取值请见 6.8

textData:聊天数据, Json

## 自定义消息发布 (YS.EVENT\_TYPE.roomPubmsg)

返回的数据: {type:YS.EVENT\_TYPE.roomPubmsg, message:pubmsgData}

pubmsgData:pubMsg 数据, Json

## 自定义消息删除 (YS.EVENT\_TYPE.roomDelmsg)

返回的数据: {type:YS.EVENT\_TYPE.roomDelmsg, message:delmsgData}

delmsgData:delMsg 数据, Json

## 用户属性改变 (YS.EVENT\_TYPE.roomUserPropertyChanged)

返回的数据: {type:YS.EVENT\_TYPE.roomUserPropertyChanged, user:user, userId:userId, message:changePropertiesData, fromID:fromID}

user:用户对象, Json

userId: 用户 id, String

changePropertiesData :改变的用户属性, Json

fromID:改变用户属性的调用者用户 id, String

注: 在大房间模式下, 如果 **changeUserProperty** 的 **told** 为 **YS.MSG\_TO\_ALLUSER** 或者 **YS.MSG\_TO\_ALLEXCEPTSENDER**, 那么没有上台的用户的用户属性改变只转发给调用者 (即调用 **changeUserProperty** 者)、被操作者 (即被改变属性的用户), 如果是上台的用户 (即发布音频/视频的用户) 的用户属性被改变则根据 **told** 转发给指定用户。

## 用户视频发布状态改变 (YS.EVENT\_TYPE.roomUserVideoStateChanged)

返回的数据: {type:YS.EVENT\_TYPE.roomUserVideoStateChanged, message:data}

message: 视频发布状态信息

包括如下属性：

**userId**: 用户 ID

**published**: 发布或停止发布，Boolean，true 表发布，false 表停止发布

**deviceId**: 视频设备 id, String 类型 【注：只有多视频模式下发布的视频才有 **deviceId**】

## 用户音频发布状态改变 (YS.EVENT\_TYPE.roomUserAudioStateChanged)

返回的数据: {type:YS.EVENT\_TYPE.roomUserAudioStateChanged, message:data}

**message**: 视频发布状态信息

包括如下属性：

**userId**: 用户 ID

**published**: 发布或停止发布，Boolean，true 表发布，false 表停止发布

## 服务器地址更新 (YS.EVENT\_TYPE.roomServerAddressUpdate)

返回的数据: {

type:YS.EVENT\_TYPE.roomServerAddressUpdate,

message:{

web\_protocol:web\_protocol,

web\_host:web\_host ,

web\_port:web\_port ,

doc\_protocol:doc\_protocol,

doc\_host:doc\_host ,

doc\_port:doc\_port ,

backup\_doc\_protocol:backup\_doc\_protocol,

backup\_doc\_host\_list:backup\_doc\_host\_list ,

backup\_doc\_port:backup\_doc\_port ,

}

}

**web\_protocol**: web 请求的协议



**web\_host** : web 请求的域名

**web\_port** : web 请求的端口

**doc\_protocol**: 文档加载的协议

**doc\_host** : 文档加载的域名

**doc\_port** : 文档加载的端口

**backup\_doc\_protocol**: 备份文档加载的协议

**backup\_doc\_host\_list** : 备份文档加载的域名列表, 类型为 Array

**backup\_doc\_port** : 备份文档加载的端口

## 用户共享媒体文件状态改变 (YS.EVENT\_TYPE.roomUserMediaStateChanged)

返回的数据: {type:YS.EVENT\_TYPE.roomUserMediaStateChanged, message:data}

**message**: 共享媒体文件状态信息

包括如下属性:

**userId**: 用户 ID

**published**: 开始共享或停止共享, Boolean, true 表开始共享, false 表停止共享

**attributes**: 媒体文件属性。包括媒体文件宽高、时长、文件名、是否暂停、是否包含音视频等。

## 房间音视频状态切换 (YS.EVENT\_TYPE.roomAudioVideoStateSwitched)

返回的数据: {type:YS.EVENT\_TYPE.roomAudioVideoStateSwitched, message:data}

**message**: 共享文件状态信息

包括如下属性:

**fromId**: 用户 ID。表名哪个用户切换了房间音视频状态

**onlyAudio**: 是否切换成只有音频的房间, Boolean, true 表只有音频, false 表音视频

## 媒体设备不可用 (YS.EVENT\_TYPE.deviceNotAvailable)

真正使用设备时(如播放本地视频或发布本地音视频)会触发此事件

返回的数据: {type:YS.EVENT\_TYPE.deviceNotAvailable, code: errorcode, message:data}

**code**: 设备不可用原因, 可能取值请见 6.5

**message:** 不可用设备信息，包含以下属性

```
message={  
  kind:'audioinput'或者'videoinput', //audioinput: 音频输入设备, videoinput: 视频输入设备  
  deviceId: 设备 id  
}
```

## 媒体设备成功/失败

(YS.EVENT\_TYPE.deviceSuccess/YS.EVENT\_TYPE.deviceFailure)

room 对象初始化(init)以后, 会触发此类事件。若有可用设备, 则触发 YS.EVENT\_TYPE.deviceSuccess 事件; 若所有设备均不可用, 则触发 YS.EVENT\_TYPE.deviceFailure 事件。

YS.EVENT\_TYPE.deviceSuccess 返回的数据: {type: YS.EVENT\_TYPE.deviceSuccess, message: {kind: kind}}

YS.EVENT\_TYPE.deviceFailure 返回的数据: {type: YS.EVENT\_TYPE.deviceFailure, message: {kind: 'audioinput'}, code: code }

失败可能的错误码只有 DEVICE\_ERROR\_UnknownError, DEVICE\_ERROR\_NotFoundError , DEVICE\_ERROR\_NotAllowedError。

## 房间模式改变 (YS.EVENT\_TYPE.roomModeChanged)

返回的数据:{ type:YS.EVENT\_TYPE.roomModeChanged, message:{roomMode:roomMode} }

roomMode:房间模式, 可能的取值请见 6.7

注: 如果是大房间 (即大规模房间) 模式, 也就是 roomMode 为 YS.ROOM\_MODE.BIG\_ROOM 时, 某些接口或者事件将发生变动, 详情请见 8.2

## 用户共享媒体文件属性更新

(YS.EVENT\_TYPE.roomUserMediaAttributesUpdate)

返回的数据: {type:YS.EVENT\_TYPE.roomUserMediaAttributesUpdate, message:data}

message: 共享媒体文件属性更新信息

包括如下属性：

**userId:** 用户 ID

**attributes:** 媒体文件属性。包括媒体文件宽高、时长、文件名、是否暂停、是否包含音视频等。

**updateAttributes:** 更新的媒体文件属性

## 错误消息通知 (YS.EVENT\_TYPE.roomErrorNotice)

返回的数据：{

type:YS.EVENT\_TYPE.roomErrorNotice,

message: {

code: 状态码, //详情请见 7.2 (错误消息通知错误码参考)

info: 补充信息, 类型为 Object //用于对相关状态码的补充说明

}

}

## INFO 消息通知 (YS.EVENT\_TYPE.roomInfoNotice)

返回的数据：{

type:YS.EVENT\_TYPE.roomInfoNotice,

message: {

code: 状态码, //详情请见 7.3 (INFO 消息通知状态码参考)

info: 补充信息, 类型为 Object //用于对相关状态码的补充说明

}

}

## 警告消息通知 (YS.EVENT\_TYPE.roomWarnNotice)

返回的数据：{

type:YS.EVENT\_TYPE.roomWarnNotice,

message: {

code: 状态码, //详情请见 7.4 (警告消息通知警告码参考)

```

    info: 补充信息，类型为 Object //用于对相关状态码的补充说明
  }
}

```

## 用户网络状态改变 (YS.EVENT\_TYPE.roomUserNetworkStateChanged)

返回的数据: {type:YS.EVENT\_TYPE.roomUserNetworkStateChanged, message:data}

message: 网络状态信息

包括如下属性:

userId: 用户 ID

deviceId: 视频设备 id, String 类型 【注: 只有多视频模式下发布的视频才有 deviceId】

networkStatus: 网络状态数据, Json

注: networkStatus 包含如下数据:

```

{
  video:{ //视频数据统计
    bitsPerSecond: 带宽 bps ,
    packetsLost: 丢包数 ,
    totalPackets: 总包数 ,
    packetsLostRate: 视频丢包率(%),
    currentDelay: 延迟 ,
    frameRate: 帧率 ,
    frameWidth: 视频宽 ,
    frameHeight: 视频高 ,
    timestamp: 时间戳 ,
    netquality: 网络质量 (可能取值请见 6.9) ,
  },
  audio:{
    bitsPerSecond: 带宽 bps ,
    packetsLost: 丢包数,
  }
}

```

```

        totalPackets:总包数 ,
        packetsLostRate:音频丢包率(%),
        currentDelay:延迟 ,
        jitter:抖动 ,
        timestamp:时间戳 ,
        netquality:网络质量（可能取值请见 6.9）,
    }
}

```

## 文件列表（YS.EVENT\_TYPE.roomFiles）

返回的数据：{type:YS.EVENT\_TYPE.roomFiles, message:fileArray}

message: 文件信息数组

文件信息包含如下内容

```

{
    "fileid"文件 id,Int 类型,
    "companyid": 企业 id, Int 类型,
    "filename": 文件名字, Int 类型,
    "uploaduserid": 上传者 id,String 类型 ,
    "uploadusername": 上传者用户名称,String 类型,
    "downloadpath": 文件下载路径, String 类型 ,
    "swfpath": 文件的查看路径, String 类型,
    "filetype":文件类型, String 类型,
    "pagenum":总页数, 该字段只对普通的静态文件有效, 对于动态 PPT 和 H5 课件无效,Int 类型
    "filecategory":文件种类, 0:课堂 , 1: 系统, Int 类型
    "fileprop": 文件分类, 0: 普通文档 , 1-2: 动态 ppt(1-旧版, 2-新版) , 3: h5 文档, Int 类型
    "type": 是否是默认文档,/0: 非默认文档 1: 默认文档, Int 类型
}

```

## 添加文件（YS.EVENT\_TYPE.roomAddFile）

返回的数据: {type:YS.EVENT\_TYPE.roomAddFile, message:data}

**message:** 关于添加的文件的描述,

包括如下属性:

**fileid:**添加的文件 id,Int 类型

**fromID:**上传者的用户 id,String

## 删除文件 (YS.EVENT\_TYPE.roomDeleteFile)

返回的数据: {type:YS.EVENT\_TYPE.roomDeleteFile, message:data}

**message:** 关于删除的文件的描述,

包括如下属性:

**fileid:**添加的文件 id,Int 类型

**fromID:**删除者的用户 id,String

**fileinfo:**文件信息

## 更新文件 (YS.EVENT\_TYPE.roomUpdateFile)

返回的数据: {type:YS.EVENT\_TYPE.roomUpdateFile, message:data}

**message:** 关于更新的文件的描述,

包括如下属性:

**fileid:**更新的文件 id,Int 类型

**fromID:**更新者的用户 id,String

**updateFileinfo:**更新的文件信息

## 视频首帧 (YS.EVENT\_TYPE.roomFirstVideoFrame)

返回的数据: {type:YS.EVENT\_TYPE.roomFirstVideoFrame, message:data}

**message:** 视频首帧数据信息,

包括如下属性:

**userId:**用户 ID,String 类型

**mediaType:**媒体类型,String 类型, 详见 6.10

width:视频宽度, Integer 类型

Height:视频高度,Integer 类型

deviceId:视频设备 id,String 类型【注：只有多视频模式下发布的视频才有 deviceId】

注：如果没有 joinroom 成功（即：没有收到 YS.EVENT\_TYPE.roomConnected 事件），则 userId 为”

## 音频首帧（YS.EVENT\_TYPE.roomFirstAudioFrame）

返回的数据：{type:YS.EVENT\_TYPE.roomFirstAudioFrame, message:data}

message：音频首帧数据信息，

包括如下属性：

userId:用户 ID,String 类型

mediaType:媒体类型,String 类型，详见 6.10

注：如果没有 joinroom 成功（即：没有收到 YS.EVENT\_TYPE.roomConnected 事件），则 userId 为”

## 主摄像头改变（YS.EVENT\_TYPE.roomUserMainCameraChanged）

返回的数据：{type:YS.EVENT\_TYPE.roomUserMainCameraChanged, message:data}

message：主摄像头改变的信息，

包括如下属性：

userId:用户 ID,String 类型

deviceId:使用的主摄像头 ID,String 类型

注：如果没有 joinroom 成功（即：没有收到 YS.EVENT\_TYPE.roomConnected 事件），则 userId 为”

# 6 SDK 中的常量

## 6.1 发送信息

名称	类型	描述
YS.MSG_TO_ALLUSER	String	发送信息给房间中的所有人
YS.MSG_TO_ALLEXCEPTSENDE	String	发送信息给除自己以外的房间中所有人

R		
YS.MSG_TO_NONEUSER	String	只发送信息到服务器，不发给任何人
userId	String	某用户 ID。只发送信息给此用户

## 6.2 打印日志

名称	类型	描述
YS.LOG_DEBUG	Integer	debug 级别日志
YS.LOG_TRACE	Integer	trace 级别日志
YS.LOG_INFO	Integer	info 级别日志
YS.LOG_WARNIN G	Integer	warning 级别日志
YS.LOG_ERROR	Integer	error 级别日志
YS.LOG_NONE	Integer	不打印日志

## 6.3 音视频发布状态

名称	类型	描述
YS.PUBLISH_STATE_NONE	Integer	没有发布
YS.PUBLISH_STATE_AUDIOONLY	Integer	只发布了音频
YS.PUBLISH_STATE_VIDEOONLY	Integer	只发布了视频
YS.PUBLISH_STATE_BOTH	Integer	发布了音频和视频

## 6.4 服务器录制类型

名称	类型	描述
YS.REC_TYPE_VIDEOLIST	String	录制房间中所有发布者的音视频，每



		个发布者的音视频都会生成一个录制文件（仅包含自己的音频）。录制结束后调用 web 接口“getuserrocordlist”获取录制文件路径。“getuserrocordlist”接口使用方法请见 web 接口文档中相关说明
YS.REC_TYPE_MP3	String	只录制房间中发布者的音频，录制后的文件为 mp3 格式。使用 web 接口“getrecordlist”接口获取录制文件路径。“getrecordlist”接口使用方法请见 web 接口文档中相关说明
YS.REC_TYPE_MIXVIDEO	String	录制房间中所有发布者的音视频，且所有视频将按一定排列规则混合录制在文件中。此类型的录制，会同时生成 mp3 和 mp4 文件。mp4 文件包含所混合排列后的视频和所有发布者的音频。mp3 文件包含所有发布者的音频。使用 web 接口“getrecordlist”接口获取录制文件路径。“getrecordlist”接口使用方法请见 web 接口文档中相关说明

## 6.5 设备不可用原因

名称	类型	描述
YS_ERR.DEVICE_ERROR_UnknownError	Integer	未知原因
YS_ERR.DEVICE_ERROR_NotFoundError	Integer	未找到设备（设备被禁用）
YS_ERR.DEVICE_ERROR_NotAllowedError	Integer	设备未授权

YS_ERR.DEVICE_ERROR_NotReadableError	Integer	设备占用
YS_ERR.DEVICE_ERROR_OverconstrainedError	Integer	设备不支持

## 6.6 视频显示模式

名称	类型	描述
YS_VIDEO_MODE.ASPECT_RATIO_CONTAIN	Integer	等比例显示在容器内
YS_VIDEO_MODE.ASPECT_RATIO_COVER	Integer	等比例填充容器(可能被裁减)

## 6.7 房间模式

名称	类型	描述
YS.ROOM_MODE.NORMAL_ROOM	String	常规房间模式
YS.ROOM_MODE.BIG_ROOM	String	大房间（大规模房间）模式

## 6.8 房间角色

名称	类型	描述
YS.ROOM_ROLE.TEACHER	Integer	老师（主讲）
YS.ROOM_ROLE.ASSISTANT	Integer	助教
YS.ROOM_ROLE.STUDENT	Integer	学生
YS.ROOM_ROLE.AUDIT	Integer	旁听（直播用户）
YS.ROOM_ROLE.PATROL	Integer	巡检员（巡课）

## 6.9 网络状态质量

名称	类型	描述
----	----	----

YS.NET_QUALITY.YS_NET_QUALITY_EXCELLENT	Integer	优（丢包率<1 或延时<80），值为 1
YS.NET_QUALITY.YS_NET_QUALITY_GOOD	Integer	良（1<=丢包率<5 或 80<=延时<150），值为 2
YS.NET_QUALITY.YS_NET_QUALITY_ACCEPTED	Integer	中（5<=丢包率<10 或 150<=延时<300），值为 3
YS.NET_QUALITY.YS_NET_QUALITY_BAD	Integer	差（10<=丢包率<20 或 300<=延时<500），值为 4
YS.NET_QUALITY.YS_NET_QUALITY_VERYBAD	Integer	极差（丢包率>=20 或 延时>=500），值为 5

## 6.10 媒体类型

名称	类型	描述
YS.MEDIA_TYPE.DEVICE_AV	String	音视频设备类型
YS.MEDIA_TYPE.MEDIA	String	媒体文件类型
YS.MEDIA_TYPE.FILE	String	本地媒体文件类型
YS.MEDIA_TYPE.SCREEN	String	屏幕共享类型

## 7 状态码参考

### 7.1 错误码参考

错误码	描述
3001	服务器过期
3002	公司已经冻结
3003	房间已删除或过期
4001	该公司不存在
4002	用户名或密码错误

4003	房间名称不允许为空
4004	时间格式错误
4005	时间设置有误
4006	老师密码格式错误
4007	房间不存在、房间无效已被删除、房间过期
4008	房间密码错误
4009	学生用户密码格式错误
4010	密码格式错误
4011	房间类型必填
4101	需要验证企业 ID
4102	验证企业 ID 错误
4103	房间数超过最大限制
4105	没有符合的数据
4106	房间文件数量超过数量限制
4107	数据库操作错误
4108	没有权限
4109	Auth 不正确
4110	该房间需要密码，请输入密码
4111	非注册用户或密码不正确，该房间要求必须注册用户才能参会
4200	组 id 必填
4201	组名必填
4202	文件 id 必填
5000	用户名密码必填
5001	房间是否有密码必填
5002	主席密码必填
5003	房间普通密码必填
5004	开始时间，结束时间必填
5005	如果允许旁听用户则应有密码
5006	房间学员密码格式错误

5007	旁听用户密码错误
5008	最大用户数格式错误
5009	最大用户数超过限制
5010	最大旁听用户数错误
5011	最大旁听用户数超过限制
5012	最大视频数格式错误
5013	最大视频数错误
5014	最大音频数格式错误
5015	最大音频数错误
5016	视频分辨率参数格式错误
5017	布局参数格式错误
5018	助教密码必填
5019	巡课密码必填
5020	视频分辨率错误或视频分辨率超过企业限制
5100	需要房间编号
5101	需要用户账号
5102	房间密码不可重复
5200	主席密码错误
5201	旁听密码错误
5203	学员密码
5204	没有密码
5205	昵称或名字必填
5206	数据写入失败
5207	用户类型 0: 主讲(老师 ) 1: 助教 2: 学员 3: 直播用户 4:巡检员
5208	用户已登陆
5300	登录 40 位 key 必填
5301	40 位 key 错误
6000	serverid 不正确
6001	ip 地址不正确

6002	普通用户数超出范围
6003	旁听用户数超出范围
2501	设备不可用错误码
2502	没有找到设备错误码
2504	设备占用错误码
2505	设备无法满足约束配置错误码
2506	设备约束对象为空或者约束都设置为 <b>false</b> 错误码
2509	检测不到 <b>web</b> 共享插件
2526	发布失败
2527	取消发布失败
2528	订阅失败
2529	取消订阅失败
2600	房间链接失败
2601	发布路数超限
2602	超过规定最大次数

## 7.2 错误消息通知错误码参考

错误码	描述
103	参数不正确
111	<b>web</b> 接口请求失败
131	发布失败（SDK 会自动重连）
132	发布超时（SDK 会自动重连）
133	发布的媒体链路超限
134	发布重连次数超过规定的最大重连数【注：最大重连数默认无限次，可以在 <b>joinroom</b> 接口进行配置重连次数】
141	订阅失败（SDK 会自动重连）
142	订阅超时（SDK 会自动重连）

7.3 INFO 消息通知状态码参考

状态码	描述
2001	发布成功
2002	订阅成功

7.4 警告消息通知警告码参考

错误码	描述
1011	切换纯音频房间时，如果发布了视频，SDK 自动取消发布视频
1012	切换音视频房间时，如果发布了音频，SDK 自动发布视频
1013	发布重连次数超过规定的最大重连数，SDK 自动取消发布视频
1014	发布重连次数超过规定的最大重连数，SDK 自动取消发布音频
1015	获取设备流失败，SDK 自动取消发布视频
1016	获取设备流失败，SDK 自动取消发布音频

注：

1) 131、132、133、134、141、142、2001、2002 错误/状态码对应的补充信息如下：

```
{
    deviceId: 设备 id,String 类型【注：只有多视频模式下发布的视频才有 deviceId】
    userId: 用户 id,
    type:媒体类型,String 类型，详见 6.10
}
```

2) 1011、1012、1013、1015 警告码对应的补充信息如下：

```
{
    deviceId: 设备 id,String 类型【注：只有多视频模式下发布的视频才有 deviceId】
}
```

8 补充说明

8.1 聊天消息转发规则说明

聊天消息 1 秒内的数据如果超过 100 条，则丢弃 100 条之后的聊天消息（老师和助教不丢弃，可以收到完整的聊天消息），例如：1 秒内收到 150 条数据，除了老师和助教外，其它人只收到前 100 条数据，下 1s 收到 90 条数据，则所有人都能收到这 90 条数据。

## 8.2 大房间（大规模房间）模式说明

1、大房间模式启用条件：人数超过 100 人【注：人数一旦超过 100 人，则切换为大房间模式，一旦启动大房间模式则不再更改房间模式，即：人数小于 100 人也不会切换回常规模式】

2、大房间模式下不通知用户加入/离开的消息（即

YS.EVENT\_TYPE.roomParticipantJoin/YS.EVENT\_TYPE.roomParticipantLeave 不再派发），因此大房间模式下，所有人的用户列表是不完整的，所以 getUsers 和 getUser 接口的调用需要根据实际需求调用，建议大房间模式下使用 getRoomUsers 和 getRoomUserNum 来获取指定用户和人数。【注：在台上的人一定存在用户列表中，因此可以使用 getUser 接口取到台上的用户】

3、大房间模式下，如果 changeUserProperty 的 told 为 YS.MSG\_TO\_ALLUSER 或者 YS.MSG\_TO\_ALLEXCEPTSENDER，那么没有上台的用户的用户属性改变只转发给调用者（即调用 changeUserProperty 者）、被操作者（即被改变属性的用户），如果是上台的用户（即发布音频/视频的用户）的用户属性被改变则根据 told 转发给指定用户。

3、大房间模式，不建议除了老师和助教外的人发送 told 为 YS.MSG\_TO\_ALLUSER（或者 YS.MSG\_TO\_ALLEXCEPTSENDER）的信令，如答题业务：老师发起答题后，学生进行答题，则学生提交的答题不建议发给所有人，建议只发给老师（助教）。

4、大房间模式，不建议改变台下用户的用户属性，建议只改变台上用户的用户属性

## 8.3 多视频模式说明

1、多视频模式启用条件：init 方法配置了 ys\_multistream 配置项为 true

2、多视频模式功能：能够发布/播放多个摄像头的视频

3、多视频模式和单视频模式的区别：

1）多视频模式可以发布/播放多个摄像头视频

2）setDevices 设置的 videoinput 在多视频模式下为设置主摄像头，设置主摄像头就是将音频关联到主摄像头上，摄像头在多视频模式下不存在切换问题，因此该接口切换的 videoinput 是切换主摄像头（即：切换音频关联的摄像头）

## 9 FAQ

Q:SDK 支持哪些浏览器版本？

A:JS-SDK 目前支持 Webkit 内核浏览器（对应谷歌浏览器 60 及 60 以上版本的 Webkit 内核）；