

web sdk概览

- 1. JRTCClient
 - checkSystemRequirements()
 - init(JRTCInitParams)
 - setVideoRecvonly()
 - setVideoParam()
 - JWebrtc.setVideoEncodingParam(resolution: string, fps)
 - setAudioCodecOptions()
 - getVideoTrack()
 - getAudioTrack()
 - getScreenTrack()
 - enterRoom()
 - exitRoom()
 - 错误监听
- 2. JRTCRoom
 - publishVideoStream()
 - publishAudioStream()
 - unPublishStream()
 - subscribeStreams()
 - unsubscribeStreams()
 - pausePublish()
 - resumePublish()
 - pauseSubscribe()
 - resumeSubscribe()
 - sendMessage()
 - changeNickName()
 - enableStreamStat()
 - getFixedAudioConsumers()
 - enableSmallStream()
 - disableSmallStream()
 - subscribeStreamsByInfo()
 - changeStreamType
 - publishDataStream
 - sendChannelData
 - subscribeDataStreams
 - 会控相关接口
 - removePeer
 - muteAudio
 - closeVideo
 - forbidChat
 - unForbidChat
 - customSignal
 - JRTCRoom房间内消息监听
 - UserJoinedRoom
 - UserLeavedRoom
 - NickNameUpdated
 - MessageRecived
 - StreamPublished
 - DataStreamPublished
 - StreamUnpublished
 - StreamPaused
 - StreamResumed
 - ImageRecived
 - 会控相关监听
 - AudioMuted
 - RoomAudioMuted
 - RoomChatUnForbidden
 - SignalCustom

1. JRTCClient

checkSystemRequirements()

const checkResult = JRTCClient.checkSystemRequirements()

参数：无

说明：检查 JRTC H5 SDK 对正在使用的浏览器的适配情况。此方法建议在初始化SDK前调用。

返回：checkResult

参数	说明
isSupport	是否支持当前浏览器, true: 支持 , false: 不支持

message	支持结果描述
result	其它信息

init(JRTCInitParams)

const JWebrtc = JRTCClient.init(JRTCInitParams:{ appId: string })

参数: initParam

返回: JRTCClient实例

JRTCInitParams:

参数	类型	说明
appId	string	appId

setVideoRecvonly()

JWebrtc.setVideoRecvonly(videoRecvonly: boolean)

参数: videoRecvonly: true是, false否, 默认false

说明: 此方法需要在加入房间前调用

有些手机上由于浏览器的版本问题，导致推视频流不支持，但是又想去拉视频流。通过这个方法设置成只接受模式，就可以兼容上述问题。

setVideoParam()

JWebrtc.setVideoParam(videoParams)

说明: 设置分辨率、码率、帧率

参数: videoParams对象

参数	说明
width	宽， 默认 width: 640
height	高， 默认 height: 360
fps	帧率，默认 fps: 25
bitrate	码率，默认 {start: 300, min: 100, max: 500}

JWebrtc.setVideoEncodingParam(resolution: string, fps)

说明: 设置分辨率、帧率

参数: resolution

参数	resolution	必填
resolution	nhd(640*360)、hd(1280*720)、fhd(1920*1080)	Y
fps	可选值15、20、25， 默认25	N

setAudioCodecOptions()

JWebrtc.setSudioCodecOptions(isStereo: Boolean)

参数: isStereo 是否开启立体声: true开启, false关闭

返回: 无

getVideoTrack()

JWebrtc.getVideoTrack()

参数： 无

说明：获取视频track

返回：videoTrack

getAudioTrack()

JWebrtc.getAudioTrack()

参数： 无

说明：获取音频track

返回：audioTrack

getScreenTrack()

JWebrtc.getScreenTrack()

说明： 获取屏幕共享track

参数： 无

返回：screenTrack

enterRoom()

JWebrtc.enterRoom(enterRoomInfo: { appId: string, token: string, userId: string | number, nonce: string | number, timestamp: string | number, userRoomId: number, nickName?: string, subscribeType?: string | number,useVp8?: boolean })

进入房间成功，会返回一个ROOM对象： JRTCRoom,

enterRoomInfo:

参数	必填	说明
appId	Y	应用ID，控制台获取
token	Y	用户生成token，生成方式参见XXXX
userId	Y	用户Id
nonce	Y	令牌随机码，用户生成
timeStamp	Y	令牌过期时间，用户生成
userRoomId	Y	房间Id
nickname	N	昵称
subscribeType	N	大房间模式下，音频订阅模式： 1 固定订阅 2 普通订阅 。默认为 1
useVp8	N	是否开启vp8环境，默认false
recordFileName	N	录制文件名
liveStreamName	N	转推直播流名

ROOM:

参数	说明
roomId	房间ID
peers	当前房间内成员信息 peers:[{"userId":"12","peerId":"129532","nickName":"AAs"}, {"userId":"13","peerId":"129533","nickName":"AAs"}]
streamInfos	当前房间内已发布流信息 streamInfos:[{"peerId":"129533","streamId":"12.129533.2.1.480","kind":"video"}, {"peerId":"129533","streamId":"12.129533.1.1.1","kind":"audio"}]
roomTemplate	1小房间， 2代表大房间

exitRoom()

JWebrtc.exitRoom()

说明：退出房间

参数： 无

返回：无

错误监听

JWebrtc.on('onError', (err) => {})

说明：错误事件，返回 { errorCode: XXX, message: XXXX}

错误信息参考：

类型	errorCode	message	说明
鉴权错误	12000	err.msg 'Authentication failure.'	鉴权错误
网络错误	11000	network_error	网络已断开
音频错误	10001	audio device not found	未找到音频设备
	10003	audio device not allowed	浏览器禁用音频设备
	10005	audio device not readable	系统禁用音频设备
视频错误	10002	video device not found	未找到视频设备
	10004	video device not allowed	浏览器禁用视频设备
	10006	video device not readable	系统禁用视频设备
屏幕共享	10010	err.message "unknown screenshare error"	屏幕共享未知错误
	10011	screenshare not allowed	屏幕共享被禁用
	10012	screenshare ended	屏幕共享已取消
其它错误	10007	"constraints " + e.constraint + " error"	OverConstrainedError无法满足要求错误
	1e4	e.message : "device unknown error"	未知错误
	。 。 。	。 。 。	。 。 。

2. JRTCRoom

调用JWebrtc.enterRoom 进入房间成功，会返回一个ROOM对象： JRTCRoom, 可通过JRTCRoom来进行房间内操作

- 1. 可直接通过调用JRTCRoom = await JWebrtc.enterRoom(XXXX)加入房间成功后的返回值来获取
- 2. 也可在加入房间成功后，调用JRTCRoom = JWebrtc.getRoomObj() 来获取

publishVideoStream()

const { track, streamId } = await JRTCRoom.publishVideoStream(videoTrack, videoType)

说明：发布视频流

参数： videoTrack, videoType

返回： track及streamId

参数	说明
videoType	默认1, 1 前置摄像头 2 后置摄像头 3 屏幕共享 4文件视频

publishAudioStream()

```
const { track, streamId } = await JRTCRoom.publishAudioStream(audioTrack)
```

说明：发布音频

参数： audioTrack

返回： track及streamId

unPublishStream()

```
JRTCRoom.unPublishStream(streamId)
```

说明：取消发布

参数： streamId

返回： 无

subscribeStreams()

```
JRTCRoom.subscribeStreams(streamIds: string[])
```

说明：订阅流，订阅成功如果产生新的消费者需手动监听streamSubscribed

参数： streamIds: string[]

返回： 无

监听： streamSubscribed，返回peerInfo， peerInfo信息如下

```
JRTCRoom.on('StreamSubscribed', ({ streamInfo}) => {  
  
    console.log(streamInfo)  
})
```

参数	说明
userId	userId
nickName	昵称
audioTrack	audioTrack
videoTrack	videoTrack

unSubscribeStreams()

```
JRTCRoom.unSubscribeStreams(streamIds: string[])
```

说明：取消订阅

参数： streamIds: string[]

返回： 无

pausePublish()

```
JRTCRoom.pausePublish(streamId)
```

说明：暂停发布

参数： streamId

返回： 无

resumePublish()

```
JRTCRoom.resumePublish(streamId)
```

说明：恢复发布

参数： streamId

返回： 无

pauseSubscribe()

JRTCRoom.pauseSubscribe(streamId)

说明： 暂停订阅

参数： streamId

返回： 无

resumeSubscribe()

JRTCRoom.resumeSubscribe(streamId)

说明： 恢复订阅

参数： streamId

返回： 无

sendMessage()

JRTCRoom.sendMessage({msg: string, userId?: string})

说明： 发送消息，可发送给指定人，可发送消息到房间

参数： msg: 消息体, userId: 指定userId

返回： 无

changeNickName()

JRTCRoom.changeNickName(nickName: string)

说明： 修改昵称

参数： nickName

返回： 无

监听： NickNameUpdate, 返回 {userRoomId: string, userId: string, nickName: string}

enableStreamStat()

JRTCRoom.enableStreamStat(isStat, intervalSec)

说明： 监控接口

参数：

参数	说明	类型
isStat	是否开启监控	boolean
intervalSec	几秒获取一次监控信息	number

返回： { JRTCNetStats, JRTCLocalStreamStats, JRTCRemoteStreamStats }

参数	说明	类型
JRTCNetStats	基本信息：	{ localIp: XX, serverIp: XX }

JRTCLocalStreamStats	本地流信息	{ userId: XX, streamId: XX, kind: XX, rtt: XX, netLoss: XX, frameWidth: XX, frameHeight: XX, fps: XX }
JRTCRemoteStreamStats	远端流信息	{ userId: XX, streamId: XX, kind: XX, jitterBufferDelay: XX, netLoss: XX, frameWidth: XX, frameHeight: XX, fps: XX }

getFixedAudioConsumers()

JRTCRoom.getFixedAudioConsumers()

说明：如果房间为大房间，可获取大房间固定音频， 可自行订阅(不用单独订阅每个参会人音频)

参数： 无

返回：fixedAudioConsumerList

fixedAudioConsumerList:

参数	说明
streamId	流ID
kind	媒体类型
track	音频track

enableSmallStream()

JRTCRoom.enableSmallStream()

说明：开启大小流模式，默认关闭

参数： 无

disableSmallStream()

JRTCRoom.enableSmallStream()

说明：关闭大小流模式

参数： 无

subscribeStreamsByInfo()

JRTCRoom.subscribeStreamsByInfo(subInfo: subInfo)

说明：订阅大小流，订阅成功后如果产生新的消费者需手动监听streamSubscribed

监听：streamSubscribed

参数： subInfo，说明如下

参数	说明
streamId	流id
streamBitrateType	0：小流， 1：中流， 2：大流

changeStreamType

JRTCRoom.changeStreamType(streamId, streamMode, streamBitrateType)

说明：切换所订阅的大小流

参数：说明如下

参数	说明
streamId	流id
streamMode	固定为2
streamBitrateType	0：小流， 1：中流， 2：大流

publishDataStream

JRTCRoom.publishDataStream()

说明：发布datachannel 数据流(字符串—图片base64或文字)

sendChannelData

JRTCRoom.sendChannelData(sendData: sendData)

说明：发布datachannel数据流

参数说明：sendData

参数	类型	说明
type	string	自定义
data	string	字符串数据

subscribeDataStreams

JRTCRoom.subscribeDataStreams(streamIds: string[])

说明：订阅datachannel 数据流， 可通过DataStreamPublished监听有人发布datachannel数据

参数：流id ， 数组

会控相关接口

removePeer

JRTCRoom.removePeer(userId, appData)

说明：移除指定用户

参数说明：

参数	类型	是否必须	说明
userId	string	Y	移除目标用户ID
appData	Object	N 默认为 {}	自定义参数

muteAudio

JRTCRoom.muteAudio({userId, appData})

说明：广播 房间内全局静音/指定用户静音

参数说明：

参数	类型	是否必须	说明
userId	string	N	目标用户ID，如果传入userId则为指定用户操作，不传则是对房间全局操作
appData	Object	N 默认为 {}	自定义参数

closeVideo

JRTCRoom.closeVideo()

说明：广播 房间内全局关闭视频/关闭指定用户视频

参数说明：同muteAudio

forbidChat

JRTCRoom.forbidChat(appData)

说明：广播房间禁言

参数说明：

参数	类型	是否必须	说明
appData	Object	N 默认为 {}	自定义参数

unForbidChat

JRTCRoom.unForbidChat()

说明：广播房间解除禁言

参数：同forbidChat

customSignal

JRTCRoom.customSignal()

说明：广播用户自定义信令

参数说明：

参数	类型	是否必须	说明
eventName	string	Y	自定义信令名称
userId	string	N	目标用户ID
appData	Object	N 默认为 {}	自定义参数

JRTCRoom房间内消息监听

UserJoinedRoom

```
JRTCRoom.on('UserJoinRoom', (data: {userId: string, nickName: string}) => {})
```

说明：加入房间

UserLeavedRoom

```
JRTCRoom.on('UserLeaveRoom', (data: {userId: string, nickName: string}) => {})
```

说明：离开房间

NickNameUpdated

```
JRTCRoom.on('NickNameUpdate', (data: {userRoomId: string, userId: string, nickName: string}) => {})
```

说明：用户修改昵称

MessageRecived

```
JRTCRoom.on('MessageRecived', (data: {userId: string, nickName: string, msg: string}) => {})
```

说明：有成员发送消息时

参数	说明
userId	发送者id
nickName	发送者昵称
msg	消息体

StreamPublished

```
JRTCRoom.on('StreamPublished', (streamsInfo: streamInfo[]) => {  
    let streamIds = streamInfos.map(streamInfo => streamInfo.streamId);  
    console.log('StreamPublished', streamIds)  
    JRTCRoom.subscribeStreams(streamIds) // 可选择订阅  
})
```

说明：有新流发布

streamsInfo:,已发布流数组 streamInfo[]: streamInfo信息如下:

参数	说明
userId	用户Id
streamId	流ID
kind	流类型
streamName	流名称

DataStreamPublished

```
JRTCRoom.on('DataStreamPublished', (streamsInfo: streamInfo[]) => {  
    let streamIds = streamInfos.map(streamInfo => streamInfo.streamId);  
    console.log('subscribeDataStreams', streamIds)  
    JRTCRoom.subscribeDataStreams(streamIds) // 可选择订阅datachannel文本流  
})
```

说明：有新datachannel文本流发布

streamsInfo:,已发布流数组 streamInfo[]: streamInfo信息如下:

StreamUnpublished

```
JRTCRoom.on('StreamUnpublished', (streamInfo: StreamInfo) => {  
    console.log('StreamUnpulished', streamInfo) // 返回取消发布的流信息  
})
```

说明：当有成员取消音频或视频发布时

StreamInfo:

参数	说明
userId	用户Id
streamId	流ID
kind	流类型

StreamPaused

```
JRTCRoom.on('StreamPaused', (streamInfo: StreamInfo) => {
    console.log('StreamPaused', streamInfo) // // 返回目标消费信息
})
```

说明：当有成员暂停音频或视频发布时，返回StreamInfo信息如下

参数	说明
userRoomId	房间ID
userId	暂停流发布 用户Id
streamId	流ID
kind	流类型

StreamResumed

```
JRTCRoom.on('StreamResumed', (streamInfo: StreamInfo) => {
    console.log('StreamResumed', streamInfo) // 返回目标消费信息
})
```

说明：当有成员恢复音频或视频发布时，返回StreamInfo信息如下

参数	说明
userRoomId	房间ID
userId	暂停流发布 用户Id
streamId	流ID
kind	流类型

ImageRecived

```
JRTCRoom.on('ImageRecived', (imageData: ImageData) => {
    console.log('ImageRecived', ImageData) //
})
```

说明：当有成员发送图片信息时，返回ImageData信息如下

参数	说明
type	类型自定义
data	图片数据

说明：命名后期需规范

会控相关监听

UserRemoved

```
JRTCRoom.on('UserRemoved', (data: {userId: string, nickName: string}) => {})
```

说明：某个用户被移出房间

返回参数 data：

参数	说明
userRoomId	userRoomId
userId	发送者 用户Id
nickName	发送者 用户昵称
appData	用户自定义参数

AudioMuted

```
JRTCRoom.on('AudioMuted', (data: {userId: string, nickName: string}) => {})
```

说明：某个用户被静音

返回参数 data：

参数	说明
userRoomId	userRoomId
userId	发送者 用户Id
nickName	发送者 用户昵称
appData	用户自定义参数

RoomAudioMuted

```
JRTCRoom.on('RoomAudioMuted', (data: {userId: string, nickName: string}) => {})
```

说明：房间被静音

返回参数 data： 同上

VideoClosed

```
JRTCRoom.on('VideoClosed', (data: {userId: string, nickName: string}) => {})
```

说明：某个用户视频被关闭

返回参数 data： 同上

RoomVideoClosed

```
JRTCRoom.on('RoomVideoClosed', (data: {userId: string, nickName: string}) => {})
```

说明：房间被关闭所有视频

返回参数 data： 同上

RoomChatForbidden

```
JRTCRoom.on('RoomChatForbidden', (data: {userId: string, nickName: string}) => {})
```

说明：房间禁止所有文字聊天

返回参数 data： 同上

RoomChatUnForbidden

```
JRTCRoom.on('RoomChatForbiddenlosed', (data: {userId: string, nickName: string}) => {})
```

说明：房间取消禁言

返回参数 data： 同上

SignalCustom

```
JRTCRoom.on('SignalCustom', (data: {userId: string, nickName: string, eventName}) => {})
```

说明：用户自定义信令

返回参数data:

参数	说明
userRoomId	userRoomId
userId	发送者 用户Id
nickName	发送者 用户昵称
appData	用户自定义参数，
eventName	自定义信令名称