

527meeting

---

## RTCPeer 接口设计说明书

文档版本：v3.0.8

发布日期：2022年12月1日

## 版权声明

本手册的所有内容，其著作权归属南京特临信息科技有限公司所有，未经本公司许可，不得以任何方式仿制、拷贝、眷抄或转译。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 修改记录

日期	作者	版本	参考版本	备注
2016-06-06	江水	1.0.0		支持音视频功能
2017-04-10	江水	1.0.1		增加服务器录制
2018-05-02	江水	1.0.2		增加本地录制，屏幕分享
2018-06-16	江水	2.0.0		增加 IM 功能
2018-09-19	江水	2.0.1		增加 Java IM 接口，平台对接方案
2020-09-16	江水	2.0.2		增加 chat
2020-09-18	江水	2.0.3		增加会议更新、删除、查询接口
2020-09-27	江水	3.0.0		替换低层，更新到 remax 流媒体版本
2020-10-27	江水	3.0.1		增加翻译功能
2020-12-02	江水	3.0.2		增加获取设备接口，优化 public 方法
2021-06-25	江水	3.0.3		增加服务器录制接口
2022-02-14	江水	3.0.4		增加修改密码和重置密码
2022-03-07	江水	3.0.5		支持视频多码流
2022-08-19	江水	3.0.6		增加本地麦克风监听，会议室人员声音事件
2022-09-29	江水	3.0.7		增加指定使用摄像头和麦克风设备
2022-12-01	江水	3.0.8		增加签名接口，修改认证接口

# 目录

1. 系统概述 .....	1
2. RTCPeer 接口 .....	1
2.1. 创建 RTCPeer 实例 .....	1
2.1.1. 概述 .....	1
2.1.2. 参数 .....	1
2.1.3. 示例 .....	2
2.2. 登录 (rtcPeer.login) .....	2
2.2.1. 概述 .....	2
2.2.2. 参数 .....	2
2.2.3. 返回参数 .....	2
2.2.4. 示例 .....	2
2.3. 登出 (rtcPeer.logout) .....	2
2.3.1. 概述 .....	2
2.3.2. 参数 .....	3
2.3.3. 返回参数 .....	3
2.3.4. 示例 .....	3
2.4. 当前用户 (rtcPeer.getCurrentUser) .....	3
2.4.1. 概述 .....	3
2.4.2. 参数 .....	3
2.4.3. 返回参数 .....	3
2.4.4. 示例 .....	3
2.5. 当前公司 (rtcPeer.getCurrentCompany) .....	3
2.5.1. 概述 .....	3
2.5.2. 参数 .....	4
2.5.3. 返回参数 .....	4
2.5.4. 示例 .....	4
2.6. 查询房间 (rtcPeer.getRoomById) .....	4
2.6.1. 概述 .....	4
2.6.2. 参数 .....	4
2.6.3. 返回参数 .....	4
2.6.4. 示例 .....	4
2.7. 创建房间 (rtcPeer.createRoom) .....	4
2.7.1. 概述 .....	4
2.7.2. 参数 .....	5
2.7.3. 返回参数 .....	5
2.7.4. 示例 .....	5
2.8. 修改房间 (rtcPeer.upadteRoom) .....	6
2.8.1. 概述 .....	6
2.8.2. 参数 .....	6

2.8.3. 返回参数 .....	6
2.8.4. 示例 .....	6
2.9. 删除房间 (rtcPeer.deleteRoom) .....	7
2.9.1. 概述 .....	7
2.9.2. 参数 .....	7
2.9.3. 返回参数 .....	7
2.9.4. 示例 .....	7
2.10. 创建临时房间 (rtcPeer.newTempRoom) .....	7
2.10.1. 概述 .....	7
2.10.2. 参数 .....	7
2.10.3. 返回参数 .....	8
2.10.4. 示例 .....	8
2.11. 修改房间密码 (rtcPeer.modifyRoomPwd) .....	8
2.11.1. 概述 .....	8
2.11.2. 参数 .....	8
2.11.3. 返回参数 .....	9
2.11.4. 示例 .....	9
2.12. 检查房间密码 (rtcPeer.checkRoomPwd) .....	9
2.12.1. 概述 .....	9
2.12.2. 参数 .....	9
2.12.3. 返回参数 .....	9
2.12.4. 示例 .....	9
2.13. 检查房间是否有密码 (rtcPeer.checkRoomHasPwd) .....	10
2.13.1. 概述 .....	10
2.13.2. 参数 .....	10
2.13.3. 返回参数 .....	10
2.13.4. 示例 .....	10
2.14. 添加房间成员 (rtcPeer.addRoomMan) .....	10
2.14.1. 概述 .....	10
2.14.2. 参数 .....	10
2.14.3. 返回参数 .....	10
2.14.4. 示例 .....	11
2.15. 删除房间成员 (rtcPeer.deleteRoomMen) .....	11
2.15.1. 概述 .....	11
2.15.2. 参数 .....	11
2.15.3. 返回参数 .....	11
2.15.4. 示例 .....	11
2.16. 查询房间成员列表 (rtcPeer.queryRoomMen) .....	11
2.16.1. 概述 .....	11
2.16.2. 参数 .....	11
2.16.3. 返回参数 .....	11
2.16.4. 示例 .....	12
2.17. 查询房间列表 (rtcPeer.queryMyRooms) .....	12
2.17.1. 概述 .....	12
2.17.2. 参数 .....	12

2.17.3. 返回参数 .....	12
2.17.4. 示例 .....	12
2.18. 检查房间是否正在使用 (rtcPeer.checkActiveRoom) .....	12
2.18.1. 概述 .....	12
2.18.2. 参数 .....	12
2.18.3. 返回参数 .....	12
2.18.4. 示例 .....	13
2.19. 创建预约会议 (rtcPeer.createAppointment) .....	13
2.19.1. 概述 .....	13
2.19.2. 参数 .....	13
2.19.3. 返回参数 .....	14
2.19.4. 示例 .....	14
2.20. 查询会议 (rtcPeer.queryAppointmentById) .....	14
2.20.1. 概述 .....	14
2.20.2. 参数 .....	15
2.20.3. 返回参数 .....	15
2.20.4. 示例 .....	15
2.21. 修改会议 (rtcPeer.updateAppointment) .....	16
2.21.1. 概述 .....	16
2.21.2. 参数 .....	16
2.21.3. 返回参数 .....	16
2.21.4. 示例 .....	17
2.22. 删除会议 (rtcPeer.deleteAppointment) .....	17
2.22.1. 概述 .....	17
2.22.2. 参数 .....	17
2.22.3. 返回参数 .....	17
2.23. 查询会议列表 (rtcPeer.queryAppointments) .....	17
2.23.1. 概述 .....	17
2.23.2. 参数 .....	17
2.23.3. 返回参数 .....	18
2.23.4. 示例 .....	18
2.24. 查询参会信息 (rtcPeer.getRoomByInviteCode) .....	18
2.24.1. 概述 .....	18
2.24.2. 参数 .....	18
2.24.3. 返回参数 .....	18
2.24.4. 示例 .....	18
2.25. 查询房间会议记录列表 (rtcPeer.queryAppointmentsByRoomId) .....	19
2.25.1. 概述 .....	19
2.25.2. 参数 .....	19
2.25.3. 返回参数 .....	19
2.25.4. 示例 .....	19
2.26. 查询用户列表 (rtcPeer.searchUserByOrgId) .....	20
2.26.1. 概述 .....	20
2.26.2. 参数 .....	20
2.26.3. 返回参数 .....	20

2.26.4. 示例 .....	20
2.27. 添加用户 (rtcPeer.addUser) .....	20
2.27.1. 概述 .....	20
2.27.2. 参数 .....	21
2.27.3. 返回参数 .....	21
2.27.4. 示例 .....	21
2.28. 修改用户 (rtcPeer.modifyUser) .....	21
2.28.1. 概述 .....	21
2.28.2. 参数 .....	21
2.28.3. 返回参数 .....	22
2.28.4. 示例 .....	22
2.29. 删除用户 (rtcPeer.deleteUser) .....	22
2.29.1. 概述 .....	22
2.29.2. 参数 .....	22
2.29.3. 返回参数 .....	22
2.29.4. 示例 .....	22
2.30. 查询流媒体当前状态 (rtcPeer.getMediaState) .....	23
2.30.1. 概述 .....	23
2.30.2. 参数 .....	23
2.30.3. 返回参数 .....	23
2.30.4. 示例 .....	23
2.31. 录制查询 (rtcPeer.getRecordByRange) .....	24
2.31.1. 概述 .....	24
2.31.2. 参数 .....	24
2.31.3. 返回参数 .....	24
2.31.4. 示例 .....	24
2.32. 删除录制 (rtcPeer.deleteRecords) .....	25
2.32.1. 概述 .....	25
2.32.2. 参数 .....	25
2.32.3. 返回参数 .....	25
2.32.4. 示例 .....	25
2.33. 创建即时会议 (rtcPeer.instantAppointment) .....	25
2.33.1. 概述 .....	25
2.33.2. 参数 .....	25
2.33.3. 返回参数 .....	26
2.33.4. 示例 .....	26
2.34. 修改密码 (rtcPeer.modifyPasswrod) .....	26
2.34.1. 概述 .....	26
2.34.2. 参数 .....	26
2.34.3. 返回参数 .....	26
2.34.4. 示例 .....	26
2.35. 重置密码 (rtcPeer.resetPasswrod) .....	27
2.35.1. 概述 .....	27
2.35.2. 参数 .....	27
2.35.3. 返回参数 .....	27

2.35.4. 示例 .....	27
<b>3. Webrtc 音视交互功能 .....</b>	<b>27</b>
3.1. 进入会议 (rtcPeer.meeting.enter) .....	28
3.1.1. 概述 .....	28
3.1.2. 参数 .....	28
3.1.3. 返回参数 .....	28
3.1.4. 示例 .....	28
3.2. 退出会议 (rtcPeer.meeting.exit) .....	29
3.2.1. 概述 .....	29
3.2.2. 参数 .....	29
3.2.3. 返回参数 .....	29
3.2.4. 示例 .....	29
3.3. 结束会议 (rtcPeer.meeting.finish) .....	29
3.3.1. 概述 .....	29
3.3.2. 参数 .....	29
3.3.3. 返回参数 .....	29
3.3.4. 监听事件 .....	29
3.3.5. 示例 .....	30
3.4. 推流到流媒体 (rtcPeer.meeting.publish) .....	30
3.4.1. 概述 .....	30
3.4.2. 参数 .....	30
3.4.3. 返回参数 .....	31
3.4.4. 示例 .....	31
3.5. 播放远端视频 (rtcPeer.meeting.play) .....	31
3.5.1. 概述 .....	31
3.5.2. 参数 .....	32
3.5.3. 返回参数 .....	32
3.5.4. 示例 .....	32
3.6. 关闭播放远端视频 (rtcPeer.meeting.closePlay) .....	32
3.6.1. 概述 .....	32
3.6.2. 参数 .....	32
3.6.3. 示例 .....	32
3.7. 打开或关闭本地视频 (rtcPeer.meeting.updateVideo) .....	33
3.7.1. 概述 .....	33
3.7.2. 参数 .....	33
3.7.3. 示例 .....	33
3.8. 打开或关闭本地麦克风 (meeting.updateMute) .....	33
3.8.1. 概述 .....	33
3.8.2. 参数 .....	33
3.8.3. 示例 .....	33
3.9. 指定用户静音消息 (rtcPeer.meeting.muteMsg) .....	34
3.9.1. 概述 .....	34
3.9.2. 参数 .....	34
3.9.3. 监听事件 .....	34
3.9.4. 示例 .....	34

3.10. 全体静音消息 (rtcPeer.meeting.muteAll) .....	35
3.10.1. 概述 .....	35
3.10.2. 参数 .....	35
3.10.3. 监听事件 .....	35
3.10.4. 示例 .....	35
3.11. 踢人消息 (rtcPeer.meeting.kickOut) .....	36
3.11.1. 概述 .....	36
3.11.2. 参数 .....	36
3.11.3. 监听事件 .....	36
3.11.4. 示例 .....	36
3.12. 举手消息 (rtcPeer.meeting.raiseHand) .....	37
3.12.1. 概述 .....	37
3.12.2. 参数 .....	37
3.12.3. 监听事件 .....	37
3.12.4. 示例 .....	37
3.13. 开始本地录制 (rtcPeer.meeting.startLocalRecord) .....	37
3.13.1. 概述 .....	37
3.13.2. 参数 .....	37
3.13.3. 示例 .....	38
3.14. 停止本地录制 (rtcPeer.meeting.stopLocalRecord) .....	38
3.14.1. 概述 .....	38
3.14.2. 参数 .....	38
3.14.3. 示例 .....	38
3.15. 暂停本地录制 (rtcPeer.meeting.pauseLocalRecord) .....	38
3.15.1. 概述 .....	38
3.15.2. 参数 .....	38
3.15.3. 示例 .....	38
3.16. 恢复本地录制 (rtcPeer.meeting.resumeLocalRecord) .....	38
3.16.1. 概述 .....	38
3.16.2. 参数 .....	38
3.16.3. 示例 .....	39
3.17. 设置主持人 (rtcPeer.meeting.setCompere) .....	39
3.17.1. 概述 .....	39
3.17.2. 参数 .....	39
3.17.3. 监听事件 .....	39
3.17.4. 示例 .....	39
3.18. 屏幕共享 (rtcPeer.meeting.shareScreen) .....	40
3.18.1. 概述 .....	40
3.18.2. 参数 .....	40
3.18.3. 监听事件 .....	40
3.18.4. 监听本地事件 .....	40
3.18.5. 示例 .....	41
3.19. 取消屏幕共享 (rtcPeer.meeting.stopShareScreen) .....	41
3.19.1. 概述 .....	41
3.19.2. 参数 .....	41

3.19.3. 示例 .....	41
3.20. 播放屏幕共享 (rtcPeer.meeting.playShareScreen) .....	41
3.20.1. 概述 .....	41
3.20.2. 参数 .....	41
3.20.3. 示例 .....	41
3.21. 挂断屏幕共享 (rtcPeer.meeting.hangupShareScreen) .....	42
3.21.1. 概述 .....	42
3.21.2. 参数 .....	42
3.21.3. 示例 .....	42
3.22. 切换摄像头 (rtcPeer.meeting.flipsCamera) .....	42
3.22.1. 概述 .....	42
3.22.2. 参数 .....	42
3.22.3. 示例 .....	42
3.23. 视频切语音 (rtcPeer.meeting.audioOnlyMeeting) .....	42
3.23.1. 概述 .....	42
3.23.2. 参数 .....	42
3.23.3. 监听事件 .....	43
3.23.4. 示例 .....	43
3.24. 发送静音消息 (rtcPeer.meeting.muteMsg) .....	44
3.24.1. 概述 .....	44
3.24.2. 参数 .....	44
3.24.3. 监听事件 .....	44
3.24.4. 示例 .....	45
3.25. 事件监听 (rtcPeer.on) .....	45
3.25.1. 概述 .....	45
3.25.2. 参数 .....	45
3.25.3. 示例 .....	46
3.26. 会议聊天 (rtcPeer.meeting.chat) .....	46
3.26.1. 概述 .....	46
3.26.2. 参数 .....	46
3.26.3. 监听事件 .....	46
3.26.4. 示例 .....	47
3.27. 会议私聊 (rtcPeer.meeting.privateChat) .....	47
3.27.1. 概述 .....	47
3.27.2. 参数 .....	47
3.27.3. 监听事件 .....	47
3.27.4. 示例 .....	48
3.28. 会议翻译 (rtcPeer.meeting.transcribe) .....	48
3.28.1. 概述 .....	48
3.28.2. 参数 .....	48
3.28.3. 监听事件 .....	48
3.28.4. 示例 .....	48
3.29. 会议翻译是否打开 (rtcPeer.meeting.isTranscribe) .....	49
3.29.1. 概述 .....	49
3.29.2. 参数 .....	49

3.29.3. 返回数据 .....	49
3.29.4. 示例 .....	49
3.30. 获取当前会议时长 (rtcPeer.meeting.getMeetingTimeLength) .....	49
3.30.1. 概述 .....	49
3.30.2. 示例 .....	49
3.31. 开始屏幕录制 (rtcPeer.meeting.startRecordScreen) .....	50
3.31.1. 概述 .....	50
3.31.2. 参数 .....	50
3.31.3. 示例 .....	50
3.32. 停止屏幕录制 (rtcPeer.meeting.stopRecordScreen) .....	50
3.32.1. 概述 .....	50
3.32.2. 示例 .....	50
3.33. 获取本地音视频设备 (rtcPeer.meeting.getMediaDevices) .....	50
3.33.1. 概述 .....	50
3.33.2. 示例 .....	50
3.34. 获取本地视频设备 (rtcPeer.meeting.getMediaVideoDevices) .....	50
3.34.1. 概述 .....	50
3.34.2. 示例 .....	51
3.35. 开始服务端录制 (rtcPeer.meeting.startRecord) .....	51
3.35.1. 概述 .....	51
3.35.2. 参数 .....	51
3.35.3. 返回参数 .....	51
3.35.4. 示例 .....	51
3.36. 停止服务端录制 (rtcPeer.meeting.stopRecord) .....	51
3.36.1. 概述 .....	51
3.36.2. 参数 .....	51
3.36.3. 返回 .....	52
3.36.4. 示例 .....	52
3.37. 检查本地音视频设备 (rtcPeer.meeting.checkDevice) .....	52
3.37.1. 概述 .....	52
3.37.2. 参数 .....	52
3.37.3. 返回参数 .....	52
3.37.4. 示例 .....	52
3.38. 检查本地视频设备 (rtcPeer.meeting.checkVideo) .....	53
3.38.1. 概述 .....	53
3.38.2. 参数 .....	53
3.38.3. 返回参数 .....	53
3.38.4. 示例 .....	53
3.39. 检查本地音频设备 (rtcPeer.meeting.checkAudio) .....	53
3.39.1. 概述 .....	53
3.39.2. 参数 .....	53
3.39.3. 返回参数 .....	53
3.39.4. 示例 .....	54
3.40. 切换视频播放源 (rtcPeer.meeting.switchVideoBySim) .....	54
3.40.1. 概述 .....	54

3.40.2. 参数 .....	54
3.40.3. 示例 .....	54
3.41. 单个视频播放源切换 (rtcPeer.meeting.switchVideoBySim) .....	54
3.41.1. 概述 .....	54
3.41.2. 参数 .....	54
3.41.3. 示例 .....	54
3.42. 视频播放最大数量 (rtcPeer.meeting.setVideoMax) .....	55
3.42.1. 概述 .....	55
3.42.2. 参数 .....	55
3.42.3. 示例 .....	55
3.43. 声控切换流是否激活 (rtcPeer.meeting.publishStreamActive) .....	55
3.43.1. 概述 .....	55
3.43.2. 参数 .....	55
3.43.3. 示例 .....	55
3.44. 修改会议名称 (rtcPeer.meeting.updateMeetingName) .....	55
3.44.1. 概述 .....	55
3.44.2. 参数 .....	55
3.44.3. 监听事件 .....	56
3.44.4. 示例 .....	56
3.45. 修改参会成员名称 (rtcPeer.meeting.updateClientName) .....	56
3.45.1. 概述 .....	56
3.45.2. 参数 .....	56
3.45.3. 监听事件 .....	57
3.45.4. 示例 .....	57
3.46. 切换音视频设备 (rtcPeer.meeting.switchDevice) .....	57
3.46.1. 概述 .....	57
3.46.2. 参数 .....	58
3.47. 监听本地麦克风音量 (rtcPeer.meeting.getMicVolume) .....	58
3.47.1. 概述 .....	58
3.47.2. 参数 .....	58
3.47.3. 示例 .....	58
3.48. 监听会议中当前说话 (rtcPeer.on("memberTalking")) .....	58
3.48.1. 概述 .....	58
3.48.2. 参数 .....	58
3.48.3. 示例 .....	58
<b>4. IM 音视交互功能 .....</b>	<b>59</b>
4.1. 发起呼叫 (rtcPeer.call) .....	59
4.1.1. 概述 .....	59
4.1.2. 1.2 参数 .....	59
4.1.3. 监听事件 .....	59
4.1.4. 示例 .....	59
4.2. 同意呼叫请求 (rtcPeer.agree) .....	60
4.2.1. 概述 .....	60
4.2.2. 2.2 参数 .....	60
4.2.3. 监听事件 .....	60

4.2.4. 示例 .....	61
4.3. 拒绝呼叫请求 (rtcPeer.reject) .....	61
4.3.1. 概述 .....	61
4.3.2. 参数 .....	61
4.3.3. 监听事件 .....	61
4.3.4. 示例 .....	61
4.4. 取消呼叫请求 (rtcPeer.cancel) .....	62
4.4.1. 概述 .....	62
4.4.2. 4.2 参数 .....	62
4.4.3. 监听事件 .....	62
4.4.4. 示例 .....	62
4.5. 响铃消息 (rtcPeer.ring) .....	63
4.5.1. 概述 .....	63
4.5.2. 5.2 参数 .....	63
4.5.3. 监听事件 .....	63
4.5.4. 示例 .....	63
4.6. 忙消息 (rtcPeer.busy) .....	64
4.6.1. 概述 .....	64
4.6.2. 6.2 参数 .....	64
4.6.3. 监听事件 .....	64
4.6.4. 示例 .....	64
4.7. 连接成功消息 (rtcPeer.connected) .....	65
4.7.1. 概述 .....	65
4.7.2. 参数 .....	65
4.7.3. 监听事件 .....	65
4.7.4. 示例 .....	65
4.8. 自定义事件消息 (rtcPeer.doEvent) .....	66
4.8.1. 概述 .....	66
4.8.2. 参数 .....	66
4.8.3. 监听事件 .....	67
4.9. 呼叫注册 (rtcPeer.doRegister) .....	67
4.9.1. 概述 .....	67
4.9.2. 参数 .....	67
4.9.3. 返回参数 .....	67
<b>5. IM 音视交互功能 Java 接口 .....</b>	<b>67</b>
5.1. 呼叫接口 .....	68
5.1.1. 概述 .....	68
5.1.2. 参数 .....	68
5.1.3. 监听事件 .....	68
5.2. 同意呼叫请求接口 .....	68
5.2.1. 概述 .....	68
5.2.2. 参数 .....	69
5.2.3. 监听事件 .....	69
5.3. 拒绝呼叫请求接口 .....	69
5.3.1. 概述 .....	69

5.3.2. 参数 .....	69
5.3.3. 监听事件 .....	70
5.4. 取消呼叫请求接口 .....	70
5.4.1. 概述 .....	70
5.4.2. 参数 .....	70
5.4.3. 监听事件 .....	70
5.5. 事件消息通用接口 .....	71
5.5.1. 概述 .....	71
5.5.2. 参数 .....	71
5.5.3. 监听事件 .....	71
5.6. 呼叫注册接口 .....	72
5.6.1. 概述 .....	72
5.6.2. 参数 .....	72
5.6.3. 返回参数 .....	72
5.7. 响铃消息 .....	72
5.7.1. 概述 .....	72
5.8. 忙消息 .....	72
5.8.1. 概述 .....	72
5.9. 连接成功消息 .....	72
5.9.1. 概述 .....	72
5.10. 创建临时房间接口 .....	73
5.10.1. 概述 .....	73
5.10.2. 参数 .....	73
5.10.3. 返回参数 .....	73
5.11. 登出接口 .....	73
5.11.1. 概述 .....	73
5.11.2. 参数 .....	73
5.11.3. 返回参数 .....	74
5.12. 创建预约会议 .....	74
5.12.1. 概述 .....	74
5.12.2. 参数 .....	74
5.12.3. 返回参数 .....	75
6. 轻会议平台对接方案 .....	75
6.1. 生成签名接口 .....	75
6.1.1. 生成签名接口 .....	76
6.2. 用户信息认证接口 .....	76
6.2.1. 概述 .....	76
6.2.2. dotokenLogin 接口 .....	77
7. Web Url 接口 .....	78
7.1. 通过房间 ID 进入指定房间 .....	78
7.1.1. 概述 .....	78
7.1.2. 请求格式 .....	78
7.1.3. 参数 .....	78
7.2. 通过会议 ID 进入指定房间 .....	79
7.2.1. 概述 .....	79

---

7.2.2. 请求格式 .....	79
7.2.3. 参数 .....	79
8. Bean 类说明 .....	80
8.1. User .....	80
8.2. Company .....	80
8.3. Room .....	80
8.4. Client .....	81
8.5. Appointment .....	81

## 1. 系统概述

527meeting 包括一套技术平台和多种终端 SDK，用于简化 WebRTC 视频通讯开发，提高了 WebRTC 的稳定性，增加了很多易用功能。

使用 527meeting，开发者可以在 Web 端和移动端，通过较简单的直接调用，将视频通讯功能嵌入任何软件中。还可以进一步对通讯的连接、效果、录制等进行编程控制。527meeting 控制台的图形界面，即可容易直观地调试通讯状况，查看统计信息，分配管理权限。

总之，527meeting 将音视频实时通讯中复杂的底层连接和优化工作封装到云端和 SDK 中。这样一来，开发者既能享受 WebRTC 本身高效、低价、无插件、点对点等优秀特性，又可以大大简化开发，避免将珍贵的时间和人力投入到无休止的与最新 WebRTC 标准对接、适配和质量调试的过程中。未来还能逐步享受到 527meeting 平台更加全面易用的功能。为了更好的对接，我们编写 RTCPeer.js，以 Javascript 类包的形式提供对接接口，并提供了原型 Demo，希望能够更好的提供我们的服务。

## 2. RTCPeer 接口

RTCPeer.js 是以 jquery 为基础包的类包，提供基于 WebRTC 技术的音视频相关功能接口，通过 RTCPeer.js 您可以轻松构建一个音视频通道。在使用 RTCPeer 时，我们首先要新建 RTCPeer 实例，然后通过 RTCPeer 实例来调用音视频相关功能接口。

### 2.1. 创建 RTCPeer 实例

#### 2.1.1. 概述

构造函数。

#### 2.1.2. 参数

参数	类型	是否必选	参数说明
host	String	是	服务器地址
relayServer	String	是	Relay 服务器地址，域名+端口号（域名不要加 https）
socket	Boolean	否	true 支持呼叫功能（和控制服务器保持 socket 连接），false 不支持，默认不保持连接，使用时才连接
nginx	Boolean	否	是否使用 nginx 和流媒体连接
transport	String	否	协议：udp 或 tcp，默认传 tcp
relay	String	否	relay 或 all，默认传 relay
sign	String	是	api 签名，有签名才能调用，通过 java 生成签名接口可获取，文档中有
logout	Boolean	否	支持非登录用户呼叫

### 2.1.3. 示例

```

1. window.rtcPeer = new RTCPeer({
2.   mode: 'pc',
3.   host: 'https://shipinhuiyi.hedait.cn/app',
4.   relayServer: "shipinhuiyi.hedait.cn:30001",
5.   relay: 'relay',
6.   transport: 'tcp'
7. });

```

## 2.2. 登录 (rtcPeer.login)

### 2.2.1. 概述

用户登录接口。

### 2.2.2. 参数

参数	类型	必填	参数说明
name	String	是	账号
password	String	是	密码

### 2.2.3. 返回参数

参数	类型	参数说明
user	Object	用户信息
company	Object	公司信息
result	number	0：成功， 1：用户名密码不正确

### 2.2.4. 示例

```

1. rtcPeer.login({
2.   name: username,
3.   password: password
4. }).then(function(d) {
5.   $("#account").text(d.user.name);
6. });

```

## 2.3. 登出 (rtcPeer.logout)

### 2.3.1. 概述

用户登出接口。

### 2.3.2. 参数

参数	类型	必填	参数说明
userId	number	是	用户唯一标识

### 2.3.3. 返回参数

参数	类型	参数说明
result	number	0：退出成功， 1：退出失败

### 2.3.4. 示例

```
1.   rtcPeer.logout({  
2.     userId: rtcPeer.getCurrentUser().id  
3.   }).then(function(d) {  
4.     showLogin(true);  
5.   });
```

## 2.4. 当前用户 (rtcPeer.getCurrentUser)

### 2.4.1. 概述

登录后，获取当前用户信息。

### 2.4.2. 参数

无

### 2.4.3. 返回参数

参数	类型	参数说明
user	Object	当前用户对象

### 2.4.4. 示例

```
1.   var currentUser = rtcPeer.getCurrentUser();
```

## 2.5. 当前公司 (rtcPeer.getCurrentCompany)

### 2.5.1. 概述

登录后，获取当前用户所在公司信息。

## 2.5.2. 参数

无

## 2.5.3. 返回参数

参数	类型	参数说明
company	Object	当前用户所在公司信息

## 2.5.4. 示例

```
1. var currentOrg = rtcPeer.getCurrentCompany();
```

## 2.6. 查询房间 (rtcPeer.getRoomById)

### 2.6.1. 概述

根据房间唯一标识查询房间信息。

### 2.6.2. 参数

参数	类型	必填	参数说明
roomId	number	是	房间唯一标识码

### 2.6.3. 返回参数

参数	类型	参数说明
room	Object	虚拟房间信息

### 2.6.4. 示例

```
1. var room=null;
2. rtcPeer.getRoomById ({
3.   roomId:1012
4. }).then(function(d){
5.   room=d.room
6. });
```

## 2.7. 创建房间 (rtcPeer.createRoom)

### 2.7.1. 概述

创建虚拟房间。

## 2.7.2. 参数

参数	类型	必填	参数说明
roomtype	number	是	房间类型 1 音频会议 2 视频会议 3 视频回传
name	String	是	房间名称
desc	String	是	房间信息描述
pictureuri	String	是	房间 Logo
definition	String	是	清晰度 1080p 720p 480p 240p
numberLimit	number	是	房间人数上限
isPublic	Boolean	是	true 公共会议室 false 私有会议室

## 2.7.3. 返回参数

参数	类型	参数说明
roomId	number	虚拟房间唯一标识
result	number	0 成功 1 失败

## 2.7.4. 示例

```

1. var req = {
2.   isPublic: false,
3.   roomtype: 2,
4.   name: 'jiangshui',
5.   desc: "test",
6.   definition: '720p',
7.   numberLimit: 100
8. }
9. rtcPeer.createRoom(req).then(function(data) {
10.   Load.hide();
11.   if (data.result == 0) {
12.     $location.url("room");
13.   } else {
14.     $scope.message = $translate.instant("message_037");
15.   }
16.   apply($scope);
17. });

```

## 2.8. 修改房间 (rtcPeer.updateRoom)

### 2.8.1. 概述

修改虚拟房间。

### 2.8.2. 参数

参数	类型	必填	参数说明
roomType	number	是	房间类型 1 音频会议 2 视频会议 3 视频回传
roomId	number	是	虚拟房间唯一标识
name	String	是	房间名称
desc	String	是	房间信息描述
pictureuri	String	是	房间 Logo
definition	String	是	清晰度 1080p 720p 480p 240p
numberLimit	number	是	房间人数上限
isPublic	Boolean	是	true 公共会议室 false 私有会议室

### 2.8.3. 返回参数

参数	类型	参数说明
roomId	number	虚拟房间唯一标识
result	number	0 成功 1 失败

### 2.8.4. 示例

```

1. var req = {
2.   roomId: room.id,
3.   isPublic: $scope.rightIndex == 1,
4.   roomtype: $scope.typeObj.type,
5.   name: $scope.nameTxt,
6.   desc: $scope.descTxt,
7.   pictureuri: $scope.logo,
8.   definition: $scope.definition.value,
9.   numberLimit: $scope.numberTxt * 1
10. }
11. rtcPeer.updateRoom(req).then(function(data) {
12.   Load.hide();
13.   if (data.result == 0) {
14.     Session.setItem('roomdetailid', data.roomId);
15.     $location.url("roomdetail");
16.   } else {

```

```

17.     $scope.message = $translate.instant("message_037");
18. }
19. apply($scope);
20. });

```

## 2.9. 删除房间 (rtcPeer.deleteRoom)

### 2.9.1. 概述

删除虚拟房间。

### 2.9.2. 参数

参数	类型	必填	参数说明
roomId	number	是	房间唯一标识码

### 2.9.3. 返回参数

参数	类型	参数说明
result	number	0 成功 1 失败

### 2.9.4. 示例

```

1.   rtcPeer.deleteRoom({
2.     roomId: $scope.room.id
3.   }).then(function(data) {
4.     Load.hide();
5.     if (data.result == 0) {
6.       $location.url("room");
7.     } else {
8.       note($translate.instant("message_024"), false);
9.     }
10.    apply($scope);
11.  });

```

## 2.10. 创建临时房间 (rtcPeer.newTempRoom)

### 2.10.1. 概述

创建临时房间。

### 2.10.2. 参数

参数	类型	必填	参数说明
name	String	是	房间名称

orgId	number	是	公司唯一标识，私有部署可传 2
userId	number	是	用户唯一标识，私有部署可传 2
roomtype	number	是	房间类型 1 音频会议 2 视频会议 3 远程培训
name	String	是	房间名称
desc	String	否	房间信息描述
pictureuri	String	否	房间 Logo
definition	String	否	清晰度 1080p 720p 480p 240p,不传默认 480p
numberLimit	number	是	房间人数上限
isPublic	Boolean	是	true 公共会议室 false 私有会议室

### 2.10.3. 返回参数

参数	类型	参数说明
result	Long	0 成功
roomId	Long	会议室房间唯一标识

### 2.10.4. 示例

```

1.   rtcPeer.newTempRoom({
2.     "roomtype": 2,
3.     "numberLimit": 200,
4.     "name": "Test Room",
5.     "userId":2,
6.     "orgId":2,
7.     "isPublic":false
8.   }).then(function(d){
9.     console.log(d)
10. })

```

## 2.11. 修改房间密码 (rtcPeer.modifyRoomPwd)

### 2.11.1. 概述

修改房间密码。

### 2.11.2. 参数

参数	类型	必填	参数说明
roomId	number	是	会议室房间唯一标识
password	String	是	密码

### 2.11.3. 返回参数

参数	类型	参数说明
result	Long	0 成功

### 2.11.4. 示例

```

1.   rtcPeer.modifyRoomPwd({
2.     roomId:599,
3.     password:"1234"
4.   }).then(function(d){
5.     console.log(d
6.   )))

```

## 2.12. 检查房间密码 (rtcPeer.checkRoomPwd)

### 2.12.1. 概述

检查房间密码是否正确。

### 2.12.2. 参数

参数	类型	必填	参数说明
roomId	number	是	会议室房间唯一标识
password	String	是	密码

### 2.12.3. 返回参数

参数	类型	参数说明
result	Long	1 成功 0 失败

### 2.12.4. 示例

```

1.   rtcPeer.checkRoomPwd({
2.     roomId:599,
3.     password:"1234"
4.   }).then(function(d){
5.     console.log(d)
6.   })

```

## 2.13. 检查房间是否有密码 (rtcPeer.checkRoomHasPwd)

### 2.13.1. 概述

检查房间是否有密码。

### 2.13.2. 参数

参数	类型	必填	参数说明
roomId	number	是	会议室房间唯一标识

### 2.13.3. 返回参数

参数	类型	参数说明
isHasPwd	Boolean	true: 有密码; false : 没有密码

### 2.13.4. 示例

```

1.   rtcPeer.checkRoomHasPwd({
2.     roomId:599
3.   }).then(function(d){
4.     console.log(d)
5.   })

```

## 2.14. 添加房间成员 (rtcPeer.addRoomMan)

### 2.14.1. 概述

添加私有房间成员，添加到房间里的成员可以通过房间列表查询到此房间。

### 2.14.2. 参数

参数	类型	必填	参数说明
roomId	number	是	会议室房间唯一标识
userId	number	是	用户唯一标识

### 2.14.3. 返回参数

参数	类型	参数说明
result	Long	0 成功

## 2.14.4. 示例

1. rtcPeer.addRoomMan({roomId:9,userId:8})

## 2.15. 删除房间成员 (rtcPeer.deleteRoomMen)

### 2.15.1. 概述

将指定成员从私有房间中移除。

### 2.15.2. 参数

参数	类型	必填	参数说明
roomId	number	是	会议室房间唯一标识
moderatorId	number	是	用户唯一标识

### 2.15.3. 返回参数

参数	类型	参数说明
result	Long	0 成功

### 2.15.4. 示例

1. rtcPeer.deleteRoomMen({roomId:81,moderatorId:138})

## 2.16. 查询房间成员列表 (rtcPeer.queryRoomMen)

### 2.16.1. 概述

根据房间唯一标识码查询房间内成员。

### 2.16.2. 参数

参数	类型	必填	参数说明
roomId	number	是	会议室房间唯一标识

### 2.16.3. 返回参数

参数	类型	参数说明
result	Long	0 成功
moderators	Array	moderator 对象

## 2.16.4. 示例

```
1. rtcPeer.queryRoomMen({roomId:8}).then(function(d){console.log(d)})
```

## 2.17. 查询房间列表 (rtcPeer.queryMyRooms)

### 2.17.1. 概述

查询当前用户有权限进入的房间列表。

### 2.17.2. 参数

无

### 2.17.3. 返回参数

参数	类型	参数说明
rooms	Array	虚拟房间列表

## 2.17.4. 示例

```
1. var rooms=[];
2. rtcPeer.queryRoomsUniq().then(function(d){
3.   rooms=d.rooms
4. });
```

## 2.18. 检查房间是否正在使用 (rtcPeer.checkActiveRoom)

### 2.18.1. 概述

检查房间是否正在使用。

### 2.18.2. 参数

参数	类型	必填	参数说明
roomId	number	是	房间唯一标识码

### 2.18.3. 返回参数

参数	类型	参数说明
result	number	0 未使用 1 正在使用中

## 2.18.4. 示例

```

1.   rtcPeer.checkActiveRoom({
2.     roomId: $scope.room.id * 1
3.   }).then(function(data) {
4.     Load.hide();
5.     if (data.result == 0) {
6.       Session.setItem('room', $scope.room);
7.       $location.url("editroom");
8.     } else {
9.       note($translate.instant("message_107"), false);
10.    }
11.    apply($scope);
12.  });

```

## 2.19. 创建预约会议 (rtcPeer.createAppointment)

### 2.19.1. 概述

创建指定时间的会议记录，可以根据会议相关信息进入虚拟房间视频通信。

### 2.19.2. 参数

参数	类型	必填	参数说明
name	String	否	会议名称，默认‘即时会议’
description	String	否	描述
definition	String	否	清晰度 1080p 720p 480p 240p, 不传默认480p
type	Number	否	1:音频会议 2 网络会议 3 在线培训
startTime	Date	否	会议开始时间， 默认接口调用时间
endTime	Date	否	会议结束时间， 默认开始时间+1 小时
members	Array	否	邀请成员，数组内成员格式为字符串， 格式如下： <code>item.memberId + "," + item.name + ", ,"            + item.email + "," + item.id +            ",Etc/GMT+8," + item.role</code> 注：memberId 默认为 0， id 为视频系统的用户 id， 可以不传 name: 用户昵称

			email:邮件，可以为 null 角色 role: 会议场模式：1 是主持人 2 普通参会人员； 培训模式：1 是主持人 2 是讲师，3 普通参会人 员，建议只邀请主持人和讲师
topics	Array	否	网络会议 子议题 数组内的 Objet 格式： { "title":“议题”， "content": "具体内容" }

### 2.19.3. 返回参数

参数	类型	参数说明
roomId	number	会议对应房间的唯一标识码
meetingId	number	会议唯一标识码

### 2.19.4. 示例

```

1.   rtcPeer.createAppointment({
2.     name: "test",
3.     description: 'abc',
4.     type:2
5.   }).then(function(d){
6.     console.log(d)
7.   })
8.   var date = new Date();
9.   rtcPeer.createAppointment({
10.    name: "shui test abc",
11.    description: 'abc',
12.    type:2,
13.    members:["0,ceshi,,,80,Etc/GMT+8,2"],
14.    startTime:new Date(),
15.    endTime:new Date(date.setHours(date.getHours() + 1))
16.  }).then(function(d){
17.    console.log(d)
18.  })

```

## 2.20. 查询会议 (rtcPeer.queryAppointmentById)

### 2.20.1. 概述

根据会议唯一标识码查询会议信息。

## 2.20.2. 参数

参数	类型	必填	参数说明
meetingId	Number	是	会议唯一标识码

## 2.20.3. 返回参数

参数	类型	参数说明
meeting	Object	会议信息
members	Array	<p>邀请成员，数组内成员格式为对象， 格式如下：</p> <pre>{ "codeId": "test" , "name": "章三" , "email": test@qq.com","role": "2",id:1}</pre> <p>注： codeId 为用户对接的唯一标示 id 为视频系统的用户 id， name: 用户昵称 email: 邮件， 角色 role: 可以不传 默认普通参会人员 会议场模式：1 是主持人 2 普通参会人员； 培训模式：1 是主持人 2 是讲师 3 是普通参会人员，建议只邀请 主持人和讲师</p>
topics	Array	<p>网络会议 子议题</p> <p>数组内的 Objet 格式：</p> <pre>{ "title": "议题" , "content": "具体内容" }</pre>

## 2.20.4. 示例

```

1. var meeting=null;
2. rtcPeer.queryAppointmentById ({ 
3.   meetingId:1012
4. }).then(function(d){ 
5.   meeting =d.meeting;
6. });

```

## 2.21. 修改会议 (rtcPeer.updateAppointment)

### 2.21.1. 概述

修改会议相关信息。

### 2.21.2. 参数

参数	类型	必填	参数说明
name	String	是	会议名称， 默认‘即时会议’
description	String	否	描述
type	Number	是	1:音频会议 2 网络会议 3 在线培训
startTime	Date	是	会议开始时间， 默认接口调用时间
endTime	Date	是	会议结束时间， 默认开始时间+1 小时
meetingId	number	是	会议唯一标识码
members	Array	否	邀请成员， 数组内成员格式为 String, 格 式 如 下 : “ memberId,name,email,id, Etc/GMT+8,role” 注：memberId 默认为 0， id 为用户的唯一标识码 id， name:用户昵称 email:邮件， 可以为 null 角色 role: 会议场模式：1 是主持人 2 普通参会人员； 培训模式：1 是主持人 2 是讲师 3 是普通参付人 员， 建议只邀请主持人和讲师
topics	Array	否	网络会议 子议题 数组内的 Objet 格式： { "title":“议题”， "content": "具体内容" }

### 2.21.3. 返回参数

参数	类型	参数说明
roomId	number	会议对应房间的唯一标识码
meetingId	number	会议唯一标识码

## 2.21.4. 示例

```

1.   rtcPeer.updateAppointment({
2.     name:"testabc",
3.     description:'abceee',
4.     type:1,
5.     meetingId:255270
6.   }).then(function(d){
7.     console.log(d)
8. })

```

## 2.22. 删除会议 (rtcPeer.deleteAppointment)

### 2.22.1. 概述

根据会议唯一标识码删除指定会议。

### 2.22.2. 参数

参数	类型	必填	参数说明
type	Number	否	1:音频会议 2 网络会议 3 在线培训
meetingId	number	是	会议唯一标识码

### 2.22.3. 返回参数

参数	类型	参数说明
result	number	0 删除成功 2 正在使用中，不能删除

## 2.23. 查询会议列表 (rtcPeer.queryAppointments)

### 2.23.1. 概述

根据时间段查询会议列表。

### 2.23.2. 参数

参数	类型	必填	参数说明
startTime	Date	否	查询会议开始时间
endTime	Date	否	查询会议结束时间， 默认开始时间+24*3 小时
pageNo	number	否	页码 从 0 开始
pageSize	number	否	一页数据量

### 2.23.3. 返回参数

参数	类型	参数说明
meetings	Array	会议列表

### 2.23.4. 示例

```

1.   rtcPeer.queryAppointments({
2.     }).then(function(d){
3.       console.log(d)
4.     })

```

## 2.24. 查询参会信息 (rtcPeer.getRoomByInviteCode)

### 2.24.1. 概述

根据会议唯一标识码或随机码查询会议信息。

### 2.24.2. 参数

参数	类型	必填	参数说明
code	Number	是	会议唯一标识码
inviteCode	Number	否	随机码, 随码入会, 有值用随机码, 没值用 code

### 2.24.3. 返回参数

参数	类型	参数说明
result	number	0 成功, 3 未开始, 5 没有查询到, 其它无效会议号
room	Object	当前会议所在虚拟房间信息
meeting	Object	当前会议信息

### 2.24.4. 示例

```

1.   rtcPeer.getRoomByInviteCode({
2.     code: +code,
3.     //inviteCode: +code
4.   }).then(function(d) {
5.     if (d.result == 0) {
6.       enterRoom(d.room);
7.     } else if (d.result == 3) {
8.       //当前会议还未开始, 请稍后再打开链接 !
9.     }

```

```

10.     $("#message").text("当前会议还未开始，请稍后再打开链接！");
11. } else {
12.     //当前会议链接已失效！
13.     $("#message").text("当前会议链接已失效！");
14. }
15. });

```

## 2.25. 查询房间会议记录列表 (rtcPeer.queryAppointmentsByRoomId)

### 2.25.1. 概述

根据房间唯一标识码查询对应的会议列表。

### 2.25.2. 参数

参数	类型	必填	参数说明
roomId	number	是	房间唯一标识码
pageNo	number	是	页码从 0 开始
pageSize	number	是	一页数据量

### 2.25.3. 返回参数

参数	类型	参数说明
resultList	Array	会议列表

### 2.25.4. 示例

```

1.     rtcPeer.queryAppointmentsByRoomId({
2.         roomId: roomId * 1,
3.         pageNo: (pageNo - 1),
4.         pageSize: 10
5.     }).then(function(data) {
6.         $scope.roomLogs = data.resultList;
7.         showTotalCount(data.totalCount || 0)
8.         if (data.totalPage > 1) {
9.             $scope.totalPage = data.totalPage;
10.            var pageList = [];
11.            for (var i = 1; i <= data.totalPage; i++) {
12.                pageList.push(i);
13.            };
14.            $scope.pageList = pageList;
15.        } else {
16.            $scope.totalPage = 1;
17.            $scope.pageList = [1];

```

```
18.    }
19.    apply($scope);
20.  })
```

## 2.26. 查询用户列表 (rtcPeer.searchUserByOrgId)

### 2.26.1. 概述

根据公司唯一标识码查询用户列表。

### 2.26.2. 参数

参数	类型	必填	参数说明
orgId	number	是	公司唯一标识码
userId	number	是	用户唯一标识
pageNo	number	是	页码从 0 开始
pageSize	number	是	一页数据量

### 2.26.3. 返回参数

参数	类型	参数说明
users	Array	用户列表
totalCount	number	数据总数
totalPage	number	数据总页数

### 2.26.4. 示例

```
1.  rtcPeer.searchUserByOrgId({
2.    orgId:2,
3.    userId:6,
4.    pageNo: 0,
5.    pageSize: 12,
6.  }).then(function(d){
7.    console.log(d)
8.  })
```

## 2.27. 添加用户 (rtcPeer.addUser)

### 2.27.1. 概述

将用户添加到指定公司。

## 2.27.2. 参数

参数	类型	必填	参数说明
orgId	number	是	公司唯一标识码
account	number	是	用户帐号
password	String	是	默认密码
level	number	是	用户级别, 4:普通用户,5:参会用户
name	String	是	用户名称

## 2.27.3. 返回参数

参数	类型	参数说明
result	number	0 成功 1 账号重复

## 2.27.4. 示例

```

1.   rtcPeer.addUser({
2.     account:'jiangshui123456',
3.     password:'123456',
4.     email:'12345@qq.com',
5.     orgId:'2',
6.     level:'2',
7.     name:'jiangabc'
8.   }).then(function(d){
9.     console.log(d)
10. })

```

## 2.28. 修改用户 (rtcPeer.modifyUser)

### 2.28.1. 概述

将用户添加到指定公司。

### 2.28.2. 参数

参数	类型	必填	参数说明
userId	number	是	用户唯一标识码
phone	String	否	手机号
address	String	否	地址
job	String	否	用户级别, 4:普通用户,5:参会用户

name	String	否	用户名称
companyname	String	否	对应的所在公司

### 2.28.3. 返回参数

参数	类型	参数说明
result	number	0 成功 1 账号重复
user	Object	更新后的用户对象

### 2.28.4. 示例

```

1.   rtcPeer.modifyUser({
2.     userId:2,
3.     name:'jiangshui123456',
4.     job:'123456',
5.     address:'12345',
6.     phone:'15012345678',
7.     name:'jiangabc'
8.   }).then(function(d){
9.     console.log(d)
10. })

```

## 2.29. 删除用户 (rtcPeer.deleteUser)

### 2.29.1. 概述

根据用户唯一标识码删除用户。

### 2.29.2. 参数

参数	类型	必填	参数说明
orgId	number	是	公司唯一标识码
userId	number	是	用户唯一标识

### 2.29.3. 返回参数

参数	类型	参数说明
result	number	0 成功 1 账号重复

### 2.29.4. 示例

```

1.   rtcPeer.deleteUser({
2.     orgId:2,
3.     userId:84,

```

```

4.     }).then(function(d){
5.         console.log(d)
6.     })

```

## 2.30. 查询流媒体当前状态 (rtcPeer.getMediaState)

### 2.30.1. 概述

查询流媒体当前状态。

### 2.30.2. 参数

无

### 2.30.3. 返回参数

参数	类型	参数说明
object	Array	多台流媒体数组

数组中的media参数说明

参数	类型	参数说明
mediaAddress	String	流媒体地址
mediaId	number	流媒体对应的唯一标识
mediaName	String	流媒体名称
participants	number	参会总人员
roomSize	String	会议室数量
status	number	0 正常, 其它:不正常

### 2.30.4. 示例

```

1.     rtcPeer.getMediaState().then(function(d){
2.     })
3.     Received data::
4.     [{...}]
5.     0:
6.     mediaAddress: "huawei.527meeting.com"
7.     mediaId: 1
8.     mediaName: "huawei"
9.     participants: 9
10.    roomSize: "2"
11.    status: 0

```

## 2.31. 录制查询 (rtcPeer.getRecordByRange)

### 2.31.1. 概述

网络录制查询接口。

### 2.31.2. 参数

参数	类型	必填	参数说明
orgId	number	是	公司唯一标识码
userId	number	否	用户唯一标识，传值查询当前用户的录制，不传 查询公司下所有录制
name	String	否	模糊查询
startDate	String	否	查询会议开始时间，传入 UTC 时间格式 “yyyyMMddHHmmss”
endDate	String	否	查询会议结束时间，传入 UTC 时间格式 “yyyyMMddHHmmss”
pageNo	number	是	页码从 0 开始
pageSize	number	是	一页数据量

### 2.31.3. 返回参数

参数	类型	参数说明
recordList	Array	录制数据
totalPage	number	总页数

### 2.31.4. 示例

```

1.   rtcPeer.getRecordByRange({
2.     userId:2,
3.     name:"",
4.     pageNo:0,
5.     pageSize: 10
6.   }).then(function(d){
7.     console.log(d)
8.   })

```

注：域名+'app/upload/videos/'+record.comment

## 2.32. 删除录制 (rtcPeer.deleteRecords)

### 2.32.1. 概述

删除录制接口。

### 2.32.2. 参数

参数	类型	必填	参数说明
recordIds	Array	是	录制 Id 数组

### 2.32.3. 返回参数

参数	类型	参数说明
result	number	0 成功 1 失败

### 2.32.4. 示例

```
1.   rtcPeer.deleteRecords({
2.     recordIds: [1,2,3]
3.   }).then(function(data) {
4. });


```

## 2.33. 创建即时会议 (rtcPeer.instantAppointment)

### 2.33.1. 概述

创建即时会议。

### 2.33.2. 参数

参数	类型	必填	参数说明
name	String	是	会议名称
orgId	number	是	公司唯一标识，私有部署可传 2
startTime	String	是	会议开始时间，传入 UTC 时间格式 “yyyyMMddHHmmss”
endTime	String	是	会议结束时间，传入 UTC 时间格式“yyyyMMddHHmmss”
type	number	是	房间类型 1 音频会议 2 视频会议 3 远程培训
isPublic	Boolean	是	ture 公共会议室 false 私有会议室

### 2.33.3. 返回参数

参数	类型	参数说明
meetingId	number	会议唯一标识
roomId	number	对应的房间 id
result	number	0 成功 1 失败

### 2.33.4. 示例

```

1.   rtcPeer.instantAppointment({
2.     "type": 2,
3.     "name": "Test Room",
4.     "userId":2,
5.     "oraId":2,
6.     "isPublic":false
7.   }).then(function(d){
8.     console.log(d)
9. })

```

## 2.34. 修改密码 (rtcPeer.modifyPasswrod)

### 2.34.1. 概述

修改用户密码接口。

### 2.34.2. 参数

参数	类型	必填	参数说明
oldpassword	String	是	旧密码
password	String	是	新密码
userId	Number	是	用户唯一标识

### 2.34.3. 返回参数

参数	类型	参数说明
result	number	0 代表密码修改成功， 1 代表旧密码不正确， 2 代表修改失败

### 2.34.4. 示例

```

1.   rtcPeer.modifyPasswrod({
2.     userId:2,
3.     oldpassword:"222222",
4.     password:"123456"
5.   }).then(function(data) {
6. });

```

## 2.35. 重置密码 (rtcPeer.resetPasswrod)

### 2.35.1. 概述

重置密码接口。

### 2.35.2. 参数

参数	类型	必填	参数说明
password	String	是	新密码
userId	Number	是	用户唯一标识

### 2.35.3. 返回参数

参数	类型	参数说明
result	number	0 成功 1 失败

### 2.35.4. 示例

```

1.   rtcPeer.resetPasswrod({
2.     userId:2,
3.     password:"123456"
4.   }).then(function(data) {
5. });

```

## 3. WebRTC 音视交互功能

针对音视频互通功能需求，提示围绕音视频相关的功能接口，使开发用户能够方便快捷的搭建音视频会议。

注：监听 `onPublished` 事件，`vCode == 'rePlay'`时，一定要修改成员列表中的 `client` 的 `peerId` 值，不然播放远端流播放不成功。

```

1.   rtcPeer.off("onPublished").on("onPublished", function (data) {
2.     if (data.roomId == room.id && data.client.id != myClient.id) {
3.       if (data.vCode == 'rePlay') {
4.         //同步流媒体通道成员 Id *****重要，不同步会影响 play*****
5.         clientList.forEach(function (item) {
6.           if (item.id == data.client.id) {
7.             item.peerId = data.client.peerId;
8.           }
9.         });
10.      }
11.    });

```

### 3.1. 进入会议 (rtcPeer.meeting.enter)

#### 3.1.1. 概述

进入会议房间。

#### 3.1.2. 参数

参数	类型	必填	参数说明
userId	Object	是	用户唯一标识
roomId	number	是	会议虚拟房间唯一标识
roomType	number	是	1 音频会议 2 视频会议 3 在线培训
username	String	否	用户名称, 游客时使用

#### 3.1.3. 返回参数

参数	类型	参数说明
client	Object	虚拟参会人员信息,当前用户 client
members	Array	虚拟参会人员列表, 不包含当前用户 client
room	Object	当前会议所在虚拟房间信息

#### 3.1.4. 示例

```

1.   rtcPeer.meeting.enter({
2.     roomId: tempRoom.id,
3.     roomType: tempRoom.roomType,
4.     maxNumber: 4,
5.     codec: currentCodec
6.   }).then(function(d) {
7.     room = d.room;
8.     myClient = d.client;
9.     clientList.push(d.client);
10.    clientList = clientList.concat(d.members);
11.    //播放自己的视频
12.    rtcPeer.meeting.publish({
13.      audio: true,
14.      video: true,
15.      domId: 'local'
16.    });
17.    ....
18.    }, function(d) {
19.      console.error("createMeeting::: " + d);
20.    });
21.  });

```

## 3.2. 退出会议 (rtcPeer.meeting.exit)

### 3.2.1. 概述

退出会议。

### 3.2.2. 参数

无

### 3.2.3. 返回参数

参数	类型	参数说明
result	number	0：成功， 1：失败

### 3.2.4. 示例

```
1. meeting.exit().then(function() {  
2.     showVideoArea(false);  
3.     $('#local').empty();  
4.     $('#network').empty();  
5. });
```

## 3.3. 结束会议 (rtcPeer.meeting.finish)

### 3.3.1. 概述

所有参会人员全部退出会议。

### 3.3.2. 参数

无

### 3.3.3. 返回参数

参数	类型	参数说明
result	number	0：成功， 1：失败

### 3.3.4. 监听事件

参数	类型	必填	参数说明
onFinish	String	是	会议结束消息事件

### 事件获取数据

参数	类型	参数说明
userId	number	调用 finish 操作的用户唯一标识

### 3.3.5. 示例

```

1.   rtcPeer.meeting.finish().then(function(){
2.     showVideoArea(false);
3.     $('#local').empty();
4.     $('#network').empty();
5.   });

```

## 3.4. 推流到流媒体 (rtcPeer.meeting.publish)

### 3.4.1. 概述

创建通道和流媒体建立连接，使用本地的摄像头和麦克风获取音视频流，并将本地流推送到流媒体服务器上。Html video 标签要加 autoplay muted，html 要增加 audio 标签作为声音外放，如：`<audio id="mixedaudio" autoplay playsinline></audio>`，不加自动创建。

### 3.4.2. 参数

参数	类型	是否必选	参数说明
audio	Boolean	是	本地麦克风是否使用获取声音
video	Boolean	是	本地视频是否使用获取视频
useVideo	Boolean	否	是否使用摄像头,如果不使用摄像头, video 属性设置为 false,默认 true
localTag	String	否	本地视频挂载的 Dom(Video tag id)
remoteTag	String	否	远端视频挂载的 Dom(Video tag id), 展示时是声控切换流 (默认流)
definition	String	否	清晰度 1080p 720p 480p 240p
maxFrameRate	number	否	最大帧率 , 范围 1~30
minFrameRate	number	否	最小帧率,范围 1~30, 注 : minFrameRate<=maxFrameRate
callback	Function	否	成功后回调
useCamera	String	否	传摄像头 deviceId 时, 使用指定摄像头, 传 any 随机选择一个;不传默认 any。注 : 可以使用 rtcPeer.meeting.getMediaVideoDevices

			拿本地摄像头设备或 rtcPeer.meeting.getMediaDevices 拿所有设备，在结果中选择其中一个摄像头设备的 deviceId
useMic	String	否	传麦克风 deviceId 时，使用指定麦克风，传 any 随机选择一个；不传默认 any 注：可以使用 rtcPeer.meeting.getMediaDevices 拿所有设备，在结果中选择其中一个麦克风设备的 deviceId
device	Object	否	通过 rtcPeer.meeting.checkDevice 检查本地设备结果，处理没有摄像头、麦克风设备等情况下，仍然可以进入；不传结果默认有摄像头、麦克风设备。

### 3.4.3. 返回参数

参数	类型	参数说明
getBitrate	Function	获取当前通道码率和帧率
client	object	虚拟参会方

### 3.4.4. 示例

Html:

```
1. <audio id="mixedaudio" autoplay playsinline></audio>
2. <video id="video_0" autoplay muted></video>
3. <video id="remote" autoplay muted></video>
```

Js code:

```
1. rtcPeer.meeting.publish({
2.   audio: true,
3.   video: true,
4.   localTag: 'video_0',
5.   remoteTag: 'remote',
6. });
```

## 3.5. 播放远端视频 (rtcPeer.meeting.play)

### 3.5.1. 概述

播放其它参会议成员的视频。如：播放 A 的视频，必须 A 已经打开自己的本地视频。

### 3.5.2. 参数

参数	类型	必填	参数说明
client	Object	是	用户参会的虚拟身份
domId	String	是	视频挂载的 Dom(Video tag id)
callback	Function	否	播放成功回调

### 3.5.3. 返回参数

参数	类型	参数说明
getBitrate	Function	获取当前通道码率和帧率
client	object	虚拟参会方

### 3.5.4. 示例

```

1. meeting.play({
2.   client: data.client,
3.   domId: 'network',
4.   callback: function(option) {
5.     $('#network').append('<div class="name-
tag" id="client' + option.client.id + '"><span>' + option.client.name + '</span></div>');
6.   }
7. })

```

## 3.6. 关闭播放远端视频 (rtcPeer.meeting.closePlay)

### 3.6.1. 概述

停止播放其它参会议成员的视频。

### 3.6.2. 参数

参数	类型	必填	参数说明
object	Object	是	用户参会的虚拟身份
callback	Function	否	停止播放成功回调

### 3.6.3. 示例

```

1. rtcPeer.meeting.stopPlay({
2.   client: data.client,
3.   callback: function(option) {
4.     $('#client' + option.client.id).remove();

```

```
5.      }
6.  });

```

### 3.7. 打开或关闭本地视频 (rtcPeer.meeting.updateVideo)

#### 3.7.1. 概述

打开或关闭本地视频。

#### 3.7.2. 参数

参数	类型	必填	参数说明
client	Object	是	用户参会的虚拟身份
video	Boolean	是	true:打开视频 false:关闭视频
callback	Function	否	回调函数

#### 3.7.3. 示例

```
1.  meeting.updateVideoMode({
2.    video: isOpenVideo,
3.    client: myClient,
4.    callback: function(myClient) {
5.      ^
6.    });

```

### 3.8. 打开或关闭本地麦克风 (meeting.updateMute)

#### 3.8.1. 概述

打开或关闭本地麦克风。

#### 3.8.2. 参数

参数	类型	必填	参数说明
isMute	Boolean	是	false:打开麦克风 true:关闭麦克风

#### 3.8.3. 示例

```
1.  rtcPeer.meeting.updateAudioMode(isAudio);
```

### 3.9. 指定用户静音消息 (rtcPeer.meeting.muteMsg)

#### 3.9.1. 概述

发送关闭或打开指定用户的麦克风消息,指定用户接收到消息后,将自己静音或取消静音。

#### 3.9.2. 参数

参数	类型	必填	参数说明
userId	number	是	Client 中的 id 值
roomId	number	是	当前会议房间的唯一标识 (room id)
isMute	Boolean	是	false : 打开麦克风 true: 关闭麦克风
super	String	否	'fromSuper' 主持人发送

#### 3.9.3. 监听事件

参数	类型	必填	参数说明
onMute	String	是	静音消息事件

事件获取数据

参数	类型	必填	参数说明
userId	number	是	Client 中的 id 值
roomId	number	是	当前会议房间的唯一标识 (room id)
isMute	Boolean	是	false : 打开麦克风 true: 关闭麦克风
super	String	否	'fromSuper' 主持人发送

#### 3.9.4. 示例

```

1.   rtcPeer.meeting. muteMsg ({
2.     userId:1234,
3.     roomId:1008,
4.     isMute:true,
5.     super: "fromSuper"
6.   });
7.   //事件监听
8.   //接收到主持人的是否禁音指令, 执行对应操作
9.   rtcPeer.off("onMute ").on("onMute ", function(data) {
10.     if (room.id == data.roomId) {
11.       if (data.userId == myClient.id && data.fromSuper) {
12.         rtcPeer.meeting. updateAudioMode (!data.isMute);
13.       }
14.     }
15.   });

```

### 3.10. 全体静音消息 (rtcPeer.meeting.muteAll)

#### 3.10.1. 概述

主持人发送全体静音消息,用户接收到消息后,将自己静音。

#### 3.10.2. 参数

参数	类型	必填	参数说明
userId	number	是	Client 中的 id 值
roomId	number	是	当前会议房间的唯一标识 (room id)
isMute	Boolean	是	false:打开麦克风 true:关闭麦克风

#### 3.10.3. 监听事件

参数	类型	必填	参数说明
onMuteAll	String	是	全体静音消息事件

事件获取数据

参数	类型	必填	参数说明
userId	number	是	Client 中的 id 值
roomId	number	是	当前会议房间的唯一标识 (room id)
isMute	Boolean	是	false:打开麦克风 true:关闭麦克风

#### 3.10.4. 示例

```

1. meeting.muteAll ({
2.   userId:1234,
3.   roomId:1008,
4.   isMute:true
5. });
6. //事件监听
7. //接收到全体静音事件, 不是主持人的用户将执行关闭麦克风操作
8. rtcPeer.off("onMuteAll").on("onMuteAll", function(data) {
9.   if (room.id == data.roomId) {
10.     meeting.updateMute (data.isMute);
11.   }
12. })

```

### 3.11. 踢人消息 (rtcPeer.meeting.kickOut)

#### 3.11.1. 概述

发送指定用户退出会议室消息，指定用户收到消息后，调用退出会议室方法，退出会议室。

#### 3.11.2. 参数

参数	类型	必填	参数说明
userId	number	是	Client 中的 id 值
roomId	number	是	当前会议房间的唯一标识 (room id)

#### 3.11.3. 监听事件

参数	类型	必填	参数说明
tiren	String	是	踢人消息事件

事件获取数据

参数	类型	必填	参数说明
userId	number	是	Client 中的 id 值
roomId	number	是	当前会议房间的唯一标识 (room id)

#### 3.11.4. 示例

```

1.   rtcPeer.meeting.kickOut ({
2.     userId:1234,
3.     roomId:1008
4.   });
5.   //事件监听
6.   rtcPeer.off("tiren").on("tiren ", function(data) {
7.     if (room.id == data.roomId) {
8.       //如果得到的 userid 和自己的 id 一样，调用退出会议方法，退出会议室
9.       if (data.userId == currentId) {
10.         //退出会议室并提示用户
11.       }
12.     }
13.   });

```

### 3.12. 举手消息 (rtcPeer.meeting.raiseHand)

#### 3.12.1. 概述

发送举手消息给当前会议中的所有成员。

#### 3.12.2. 参数

无

#### 3.12.3. 监听事件

参数	类型	必填	参数说明
onRaiseHand	String	是	踢人消息事件

事件获取数据

参数	类型	必填	参数说明
client	Object	是	Client
roomId	number	是	当前会议房间的唯一标识 (room id)

#### 3.12.4. 示例

```

1.   rtcPeer.meeting. raiseHand()
2. //事件监听
3.   rtcPeer.off("onRaiseHand").on("onRaiseHand", function(d) {
4.     if (d.client.id != myClient.id) {
5.       console.log('举手----', d.client.name);
6.     }
7.   });

```

### 3.13. 开始本地录制 (rtcPeer.meeting.startLocalRecord)

#### 3.13.1. 概述

通过屏幕共享和录制插件抓屏，将屏幕和声音录制下来。只支持 PC 端。

#### 3.13.2. 参数

参数	类型	必填	参数说明
fileName	String	是	保存文件名
callback	Function	是	回调函数

### 3.13.3. 示例

```
1. rtcPeer.meeting.startLocalRecord ({  
2.   fileName : "abcd"  
3.});
```

## 3.14. 停止本地录制 (rtcPeer.meeting.stopLocalRecord)

### 3.14.1. 概述

停止本地录制。只支持 PC 端，浏览器需要安装插件。

### 3.14.2. 参数

参数	类型	必填	参数说明
callback	Function	是	回调函数

### 3.14.3. 示例

```
1. rtcPeer.meeting.stopLocalRecord();
```

## 3.15. 暂停本地录制 (rtcPeer.meeting.pauseLocalRecord)

### 3.15.1. 概述

暂停本地录制。只支持 PC 端。

### 3.15.2. 参数

无

### 3.15.3. 示例

```
1. rtcPeer.meeting.pauseLocalRecord();
```

## 3.16. 恢复本地录制 (rtcPeer.meeting.resumeLocalRecord)

### 3.16.1. 概述

恢复本地录制。只支持 PC 端。

### 3.16.2. 参数

无

### 3.16.3. 示例

```
1. rtcPeer.meeting.resumeLocalRecord();
```

## 3.17. 设置主持人 (rtcPeer.meeting.setCompere)

### 3.17.1. 概述

将指定用户设置成主持人,设置成功后,全体会议成员都会收到设置主持人消息,各终端收到事件可在成员列表中找到该主持人。

### 3.17.2. 参数

参数	类型	必填	参数说明
userId	number	是	Client 中的 id 值
roomId	number	是	当前会议房间的唯一标识 (room id)

### 3.17.3. 监听事件

参数	类型	必填	参数说明
onCompere	String	是	设置主持人消息事件

事件获取数据

参数	类型	必填	参数说明
userId	number	是	Client 中的 id 值
roomId	number	是	当前会议房间的唯一标识 (room id)

### 3.17.4. 示例

```
1. rtcPeer.meeting.setCompere({
2.   userId:1234,
3.   roomId:1008
4. });
5. //事件监听
6. rtcPeer.off("onCompere ").on("onCompere ", function(data) {
7.   if (room.id == data.roomId) {
8.   }
9. });
```

### 3.18. 屏幕共享 (rtcPeer.meeting.shareScreen)

#### 3.18.1. 概述

将自己的屏幕共享到其它客户端。只支持PC端。会议中，同一时间只能有一个人分享屏幕，html video标签一定要加autoplay muted。

#### 3.18.2. 参数

参数	类型	必填	参数说明
shareScreenId	number	是	临时创建的分享唯一标识
domId	String	是	video 标签唯一 dom id

#### 3.18.3. 监听事件

参数	类型	必填	参数说明
onShareScreen	String	是	分享控制事件

事件获取数据

参数	类型	参数说明
shareScreenId	number	shareScreenId 有值表是分享，为空表册关闭
roomId	number	分享所在的房间唯一标识
client	Object	分享的发起者

#### 3.18.4. 监听本地事件

参数	类型	必填	参数说明
shareScreen	String	是	分享屏幕者的本地事件

事件获取数据

参数	类型	参数说明
type	String	type 值： play, stop, 事件类型 play:屏幕分享 stop:停止分享
shareScreenId	number	shareScreenId 分享 id

### 3.18.5. 示例

Html:

```
1. <video id='screenvideo' autoplay muted></video>
```

Js code:

```
1. rtcPeer.meeting.shareScreen({  
2.   domId:'screenvideo'  
3. });
```

## 3.19. 取消屏幕共享 (rtcPeer.meeting.stopShareScreen)

### 3.19.1. 概述

取消当前自己共享的屏幕。只支持 PC 端。

### 3.19.2. 参数

参数	类型	必填	参数说明
callback	Function	否	回调函数

### 3.19.3. 示例

```
1. rtcPeer.meeting.stopShareScreen();
```

## 3.20. 播放屏幕共享 (rtcPeer.meeting.playShareScreen)

### 3.20.1. 概述

播放远端屏幕共享。只支持 PC 端。html video 标签一定要加 autoplay muted。

### 3.20.2. 参数

参数	类型	必填	参数说明
shareScreenId	number	是	临时创建的分享唯一标识
domId	String	是	video 标签唯一 dom id

### 3.20.3. 示例

Html:

```
1. <video id='screenvideo' autoplay muted></video>
```

Js code:

```
1.   rtcPeer.meeting.playShareScreen({  
2.       shareScreenId: data.shareScreenId * 1,  
3.       domId:"screenvideo"  
4.   });
```

### 3.21. 挂断屏幕共享 (rtcPeer.meeting.hangupShareScreen)

#### 3.21.1. 概述

非主持人停止播放主持人屏幕共享时，调用的接口。

#### 3.21.2. 参数

无

#### 3.21.3. 示例

```
1.   rtcPeer.meeting.hangupScreenshare();
```

### 3.22. 切换摄像头 (rtcPeer.meeting.flipsCamera)

#### 3.22.1. 概述

移动端切换摄像头。

#### 3.22.2. 参数

参数	类型	必填	参数说明
callback	Function	否	回调函数

#### 3.22.3. 示例

```
1.   rtcPeer.meeting.flipsCamera(function(){});
```

### 3.23. 视频切语音 (rtcPeer.meeting.audioOnlyMeeting)

#### 3.23.1. 概述

将视频会议切换成语音会议模式。

#### 3.23.2. 参数

参数	类型	必填	参数说明
----	----	----	------

client	Object	是	用户参会的虚拟身份
callback	Function	否	回调函数

### 3.23.3. 监听事件

参数	类型	必填	参数说明
onPublished	String	是	视频消息事件

事件获取数据

参数	类型	参数说明
event	String	event 值： play, close, audioMode 事件类型 play:播放视频 close:关闭视频 audioMode:视频电话切语音电话
client	Object	发起动作的用户信息

### 3.23.4. 示例

```

1.   rtcPeer.meeting.audioOnlyMeeting ({
2.     client: myClient,
3.     callback: function(myClient) {
4.       $("#local").css('display', 'block');
5.       clientList.forEach(function(item) {
6.         if (item.id == myClient.id) {
7.           item.isOpenVideo = false;
8.         }
9.       });
10.      changeVideoSize();
11.    }
12.  });
13. //接收其他客户端播放或停止事件
14. rtcPeer.off("onPublished").on("onPublished", function(data) {
15.   if (data.roomId == room.id && data.client.id != myClient.id) {
16.     if (data.event == 'close' || data.event == 'audioMode') {
17.       console.info("关闭播放 client " + data.client.name);
18.       rtcPeer.meeting.stopPlay({
19.         client: data.client,
20.         callback: function(client) {
21.           $('#network' + client.id).remove();
22.           clientList.forEach(function(item) {
23.             if (item.id == data.client.id) {
24.               item.isOpenVideo = false;
25.             }
26.           });
27.           changeVideoSize();
28.         }
29.       });
30.     }
31.   }
32. });

```

```

29.      });
30.      if (isOpenVideo && data.event == 'audioMode') {
31.          isOpenVideo = false;
32.          $("#togglerVideo").removeClass('camera_focus');
33.          rtcPeer.meeting.updateVideoMode({
34.              video: isOpenVideo,
35.              client: myClient,
36.              callback: function(myClient) {
37.                  $("#local").css('display', 'none');
38.                  clientList.forEach(function(item) {
39.                      if (item.id == myClient.id) {
40.                          item.isOpenVideo = isOpenVideo;
41.                      }
42.                  });
43.                  changeVideoSize();
44.              }
45.          });
46.      }
47.  }
48. }
49. });

```

## 3.24. 发送静音消息 (rtcPeer.meeting.muteMsg)

### 3.24.1. 概述

通过发送指定某用户的麦克风打开关闭。

### 3.24.2. 参数

参数	类型	必填	参数说明
userId	number	是	用户唯一标识
roomId	number	是	房间唯一标识
isMute	Boolean	是	是否静音, true 静音, false 取消静音

### 3.24.3. 监听事件

参数	类型	必填	参数说明
onMute	String	是	静音消息事件

事件获取数据

参数	类型	参数说明
userId	number	用户唯一标识
roomId	number	房间唯一标识
isMute	Boolean	是否静音, true 静音, false 取消静音

### 3.24.4. 示例

```

1. $scope.muteOther = function(user) {
2.   rtcPeer.meeting.muteMsg({
3.     roomId: room.id,
4.     userId: user.id,
5.     isMute: !user.muted
6.   })
7. };
8. //接收是否静音
9. $scope.$on('onMute', function(e, data) {
10.   if (room.id == data.roomId) {
11.     angular.forEach($scope.userList, function(member) {
12.       //其它人看到自己的状态
13.       if (member.id == data.userId) {
14.         member.muted = data.isMute;
15.         return;
16.       }
17.     });
18.   if (data.userId == client.id) {
19.     $scope.isMute = data.isMute;
20.     rtcPeer.meeting.setAudio(!$scope.isMute);
21.   }
22. }
23. });

```

## 3.25. 事件监听 (rtcPeer.on)

### 3.25.1. 概述

监听事件方法。

### 3.25.2. 参数

参数	类型	必填	参数说明
Key	String	是	消息事件名：换成对应事件名

主要事件

参数	类型	参数说明
EnterRoom	String	进入房间事件
ExitRoom	String	退出房间事件
onLocalPublished	String	本地流连接流媒体服务器成功 本地播放成功事件
onPublished	String	视频播放或停止事件
onLinkSlow	String	code: 1001 本地视频网络上传较差 code: 1002 远端视频网络下载较差

onLocalRecord	String	录制本地流事件，type 为 startLocalRecord 开始录制，stopLocalRecord 停止录制，pauseLocalRecord 暂停录制，resumeLocalRecord 恢复录制
---------------	--------	--

### 3.25.3. 示例

```

1.   rtcPeer.on("EnterRoom", function(data) {
2.     if (room.id == data.roomId) {
3.       if (data.result == 9) {
4.         //当前账号在其他客户端上进入当前会议
5.         meeting.exitRoom().then(function() {
6.           showVideoArea(false);
7.         });
8.       } else if (data.client.id != myClient.id) {
9.         console.info("进入房间 client " + data.client.name);
10.        clientList.push(data.client);
11.      }
12.    }
13.  })

```

## 3.26. 会议聊天 (rtcPeer.meeting.chat)

### 3.26.1. 概述

会议聊天。

### 3.26.2. 参数

参数	类型	必填	参数说明
userId	number	是	Client 中的 id 值，当前用户
roomId	number	是	当前会议房间的唯一标识 (room id)
content	String	是	聊天内容

### 3.26.3. 监听事件

参数	类型	必填	参数说明
onChat	String	是	聊天消息事件

事件获取数据

参数	类型	参数说明
userId	number	Client 中的 id 值
roomId	number	当前会议房间的唯一标识 (room id)

content	String	聊天内容
client	Client	发送信息的人信息

### 3.26.4. 示例

```

1.   rtcPeer.meeting.chat({content:"1233eeeeeee"})
2. //事件监听
3. //接收到主持人的是否禁音指令，执行对应操作
4. rtcPeer.off("onChat").on("onChat", function(data) {
5.     if (room.id == data.roomId) {
6.         console.log("chat message", data.content);
7.     }
8. });

```

## 3.27. 会议私聊 (rtcPeer.meeting.privateChat)

### 3.27.1. 概述

会议指定和某一个人聊天。

### 3.27.2. 参数

参数	类型	必填	参数说明
userId	number	是	Client 中的 id 值，当前用户
otherUserId	number	是	指定聊天的的 client id，当前会议已存的用户
roomId	number	是	当前会议房间的唯一标识 (room id)
content	String	是	聊天内容

### 3.27.3. 监听事件

参数	类型	必填	参数说明
onChat	String	是	聊天消息事件

事件获取数据

参数	类型	参数说明
userId	number	Client 中的 id 值
roomId	number	当前会议房间的唯一标识 (room id)
content	String	聊天内容
client	Client	发送信息的人信息

### 3.27.4. 示例

```

1. rtcPeer.meeting.privateChat({content:"1233eeeeeee",otherUserId})
2. //事件监听
3. //接收到主持人的是否禁音指令，执行对应操作
4. rtcPeer.off("onPrivateChat").on("onPrivateChat", function(data) {
5.     if (room.id == data.roomId) {
6.         console.log("chat message", data.content);
7.     }
8. });

```

## 3.28. 会议翻译 (rtcPeer.meeting.transcribe)

### 3.28.1. 概述

会议中参会人员的声音翻译。

### 3.28.2. 参数

参数	类型	必填	参数说明
action	String	是	start stop pause resume]启动翻译，停止翻译，暂停和恢复

### 3.28.3. 监听事件

参数	类型	必填	参数说明
onTranscription	String	是	翻译消息事件

事件获取数据

参数	类型	必填	参数说明
var	String	是	翻译文字
onebest	String	是	最终调整后的翻译文字（不是每次翻译后都有 onebest）
bg	String	是	同一句翻译，唯一标识

### 3.28.4. 示例

```

1. //开始翻译
2. rtcPeer.meeting.transcribe({ action: start});
3. //停止翻译
4. rtcPeer.meeting.transcribe({ action: 'stop' });
5. //接收翻译事件
6. var transcriptionTxt = "";

```

```

7. var transcriptionNumber = ""; //同样的编号属于一句话
8. rtcPeer.off("onTranscription").on("onTranscription", function(d) {
9.     console.log(d);
10.    if (d.data) {
11.        transcriptionTxt = d.data.var || d.data.onebest;
12.        $("#transcribeTxt").text(transcriptionTxt);
13.    }
14. });

```

## 3.29. 会议翻译是否打开 (rtcPeer.meeting.isTranscribe)

### 3.29.1. 概述

会议中参会人员的声音翻译。

### 3.29.2. 参数

无

### 3.29.3. 返回数据

参数	类型	参数说明
message	String	描述
result	Boolean	true 翻译打开状态, false 翻译关闭状态

### 3.29.4. 示例

```

1. Mixer.instance.api.isTranscribe().then(function(d){console.error(d)})
2. //结果：
3. {message: "+OK conference 100000508 transcription is on", sessid: "ba63d78c-5a4b-c541-dff4-
ce13b337720b", result: true}
4. message:"+OK conference 100000508 transcription is on"
5. result:true
6. sessid:"ba63d78c-5a4b-c541-dff4-ce13b337720b"

```

## 3.30. 获取当前会议时长 (rtcPeer.meeting.getMeetingTimeLength)

### 3.30.1. 概述

获取当前正在进行的会议已经进行了多长时间。

### 3.30.2. 示例

```

1. rtcPeer.utils.dateTimeFormat(rtcPeer.meeting.getMeetingTimeLength());
2. ....
3. "00:00:48"

```

### 3.31. 开始屏幕录制 (rtcPeer.meeting.startRecordScreen)

#### 3.31.1. 概述

通过屏幕共享和录制插件抓屏，将屏幕和声音录制到服务端。只支持PC端。

#### 3.31.2. 参数

参数	类型	必填	参数说明
fileName	String	否	保存文件名

#### 3.31.3. 示例

```
1. rtcPeer.meeting.startRecordScreen();
```

### 3.32. 停止屏幕录制 (rtcPeer.meeting.stopRecordScreen)

#### 3.32.1. 概述

停止录制。只支持 PC 端，浏览器需要安装插件。

#### 3.32.2. 示例

```
1. rtcPeer.meeting.stopRecordScreen();
```

### 3.33. 获取本地音视频设备 (rtcPeer.meeting.getMediaDevices)

#### 3.33.1. 概述

获取本地麦克风、摄像头、外放列表。

#### 3.33.2. 示例

```
1. rtcPeer.meeting.getMediaDevices(function(list){  
2.   console.log("list",list);  
3. });
```

### 3.34. 获取本地视频设备 (rtcPeer.meeting.getMediaVideoDevices)

#### 3.34.1. 概述

获取本地摄像头设备列表。如果没有摄像头设备，返回为空。

### 3.34.2. 示例

```
1. rtcPeer.meeting.getMediaVideoDevices(function(list){  
2.   console.log("list",list);  
3. });
```

## 3.35. 开始服务端录制 (rtcPeer.meeting.startRecord)

### 3.35.1. 概述

发起服务端录制，将会议中用户声控切换的视频画面和声音录制下来。

### 3.35.2. 参数

参数	类型	必填	参数说明
fileName	String	否	保存文件名，不传已当前时间为名称
startTime	Date	否	录制开始时间，不传默认当前时间

### 3.35.3. 返回参数

参数	类型	参数说明
result	number	0 成功
recordId	number	录制唯一标识

### 3.35.4. 示例

```
1. rtcPeer.meeting.startLocalRecord({  
2.   fileName : "abcd"  
3. });
```

## 3.36. 停止服务端录制 (rtcPeer.meeting.stopRecord)

### 3.36.1. 概述

停止服务端录制。

### 3.36.2. 参数

参数	类型	必填	参数说明
recordId	number	是	录制唯一标识

### 3.36.3. 返回参数

参数	类型	参数说明
result	number	0 成功

### 3.36.4. 示例

```
1. rtcPeer.meeting.stopRecord({recordId:123});
```

## 3.37. 检查地音视频设备 (rtcPeer.meeting.checkDevice)

### 3.37.1. 概述

检查当前机器设备情况。

### 3.37.2. 参数

参数	类型	必填	参数说明
definition	String	是	分辨率：1080p 或 720p
callback	function	是	回调函数，结果返回

### 3.37.3. 返回参数

参数	类型	参数说明
video	Object	code:1000 有摄像头, 1001:没有权限, 1002 : 没有发现、不能读取或不支持, 1003 其它错误
audio	Object	code:1000 有麦克风, 1001:没有权限, 1002 : 没有发现、不能读取或不支持, 1003 其它错误

### 3.37.4. 示例

```
1. rtcPeer.meeting.checkDevice({
2.   definition: tmpRoom.definition,
3.   callback: function (data) {
4.     if (data.video.code == 1001) {
5.       //摄像头或麦克风不存在
6.       showMessage("请允许使用当前摄像头");
7.     } else if (data.audio.code == 1001) {
8.       showMessage("请允许使用当前麦克风");
9.     } else {
10.       checkDeviceData = data;
11.     }
12.   }
13. })
```

### 3.38. 检查本地视频设备 (rtcPeer.meeting.checkVideo)

#### 3.38.1. 概述

检查当前机器设备情况。

#### 3.38.2. 参数

参数	类型	必填	参数说明
definition	String	是	分辨率：1080p 或 720p
callback	function	是	回调函数，结果返回

#### 3.38.3. 返回参数

参数	类型	参数说明
video	Object	code:1000 有摄像头, 1001:没有权限, 1002 : 没有发现、不能读取或不支持, 1003 其它错误

#### 3.38.4. 示例

```
1.   rtcPeer.meeting.checkVideo({
2.     definition: tmpRoom.definition,
3.     callback: function (data) {}
4.   })
```

### 3.39. 检查本地音频设备 (rtcPeer.meeting.checkAudio)

#### 3.39.1. 概述

检查当前机器设备情况。

#### 3.39.2. 参数

参数	类型	必填	参数说明
callback	function	是	回调函数，结果返回

#### 3.39.3. 返回参数

参数	类型	参数说明
audio	Object	code:1000 有摄像头, 1001:没有权限, 1002 : 没有发现、不能读取或不支持, 1003 其它错误

### 3.39.4. 示例

```
1.   rtcPeer.meeting.checkAudio({  
2.       callback: function (data) {}  
3.   })
```

## 3.40. 切换视频播放源 (rtcPeer.meeting.switchVideoBySim)

### 3.40.1. 概述

根据传参切换视频流大小，当前所有播放的视频流都将切换到同一大小。

### 3.40.2. 参数

参数	类型	必填	参数说明
sim	number	是	0 小流，1 中流，2 大流，默认大流

### 3.40.3. 示例

```
1.   rtcPeer.meeting.switchVideoBySim(2);
```

## 3.41. 单个视频播放源切换 (rtcPeer.meeting.switchVideoBySim)

### 3.41.1. 概述

根据视频的 domid 和 sim 纸数，指定切换视频流大小。

### 3.41.2. 参数

参数	类型	必填	参数说明
sim	number	是	0 小流，1 中流，2 大流，默认大流
domId	String	是	对应的 video 标签 Id

### 3.41.3. 示例

```
1.   rtcPeer.meeting.switchVideoSimByDom('video_1',2);
```

### 3.42. 视频播放最大数量 (rtcPeer.meeting.setVideoMax)

#### 3.42.1. 概述

当前会议最大展示视频个数，包括自己本地视频。如设置 4，播放第 5 个会自动将最先播放的视频流替换。

#### 3.42.2. 参数

参数	类型	必填	参数说明
number	number	是	最大播放视频个数

#### 3.42.3. 示例

```
1. rtcPeer.meeting.setVideoMax(25);
```

### 3.43. 声控切换流是否激活 (rtcPeer.meeting.publishStreamActive)

#### 3.43.1. 概述

声控切换流是否激活,true 会从流媒体将声控切换流拉下来播放，false 将将声控切换流关闭。

#### 3.43.2. 参数

参数	类型	必填	参数说明
isPlay	Boolean	是	是否打开

#### 3.43.3. 示例

```
1. rtcPeer.meeting.publishStreamActive(false);
```

### 3.44. 修改会议名称 (rtcPeer.meeting.updateMeetingName)

#### 3.44.1. 概述

在会议中，修改会议名称，并发送消息给所有参会方。

#### 3.44.2. 参数

参数	类型	必填	参数说明
userId	number	是	用户唯一标识
roomId	number	是	房间唯一标识
appointmentName	String	是	会议名称
appointmentId	String	是	会议唯一标识 room 对象中

		appointmentId，进入会议时返回
--	--	-----------------------

### 3.44.3. 监听事件

参数	类型	必填	参数说明
onUpdateMeeting	String	是	静音消息事件

事件获取数据

参数	类型	参数说明
userId	number	用户唯一标识
roomId	number	房间唯一标识
appointmentName	String	修改后的会议名称

### 3.44.4. 示例

```

1.   rtcPeer.meeting.updateMeetingName({
2.     roomId: room.id,
3.     userId: user.id,
4.     appointmentName: "abc",
5.     appointmentId: room.appointmentId
6.   })
7.   //接收是否静音
8.   $scope.$on('onUpdateMeeting', function(e, data) {
9.     if (room.id == data.roomId) {
10.       room.appointmentName = data.appointmentName;
11.     }
12.   });

```

## 3.45. 修改参会成员名称 (rtcPeer.meeting.updateClientName)

### 3.45.1. 概述

在会议中，修改参会成员名称，并发送消息给所有参会方。

### 3.45.2. 参数

参数	类型	必填	参数说明
userId	number	是	用户唯一标识
roomId	number	是	房间唯一标识
name	String	是	成中名称

### 3.45.3. 监听事件

参数	类型	必填	参数说明
onUpdateClient	String	是	修改成员名称消息事件

事件获取数据

参数	类型	参数说明
userId	number	用户唯一标识
roomId	number	房间唯一标识
client	Object	成员对象

### 3.45.4. 示例

```

1.   rtcPeer.meeting.muteMsg({
2.     roomId: room.id,
3.     userId: user.id,
4.     name: 'abc'
5.   })
6.   //接收是否静音
7.   $scope.$on('onUpdateClient', function(e, data) {
8.     if (room.id == data.roomId) {
9.       angular.forEach($scope.userList, function(member) {
10.         //其它人看到自己的状态
11.         if (member.id == data.client.id) {
12.           member.name= data.client.name;
13.           return;
14.         }
15.       });
16.       if (data.userId == user.id) {
17.         user.name = data.client.name;
18.       }
19.     }

```

## 3.46. 切换音视频设备 (rtcPeer.meeting.switchDevice)

### 3.46.1. 概述

在会议中，通过 `rtcPeer.meeting.getMediaDevices` 接口获取音视频设备，将需要的音视频对象传切换到对应的设备上。

### 3.46.2. 参数

参数	类型	必填	参数说明
video	Object	否	摄像头对象
audio	Object	否	麦克风对象
callback	function	是	成功后回调

注：video、audio对象必须有一个，否者报错

## 3.47. 监听本地麦克风音量 (rtcPeer.meeting.getMicVolume)

### 3.47.1. 概述

在会议中，获取当前使用的麦克风音量值。

### 3.47.2. 参数

参数	类型	必填	参数说明
callback	function	是	返回声音大小

### 3.47.3. 示例

```
1.   rtcPeer.meeting.getMicVolume(function (volume) {
2.     console.log("volume", (volume / 30000).toFixed(2) * 100 + "%");
3.   })
```

## 3.48. 监听会议中当前说话 (rtcPeer.on("memberTalking"))

### 3.48.1. 概述

在会议中，监听会议中当前说话人。

### 3.48.2. 参数

参数	类型	参数说明
member	Object	当前说话人

### 3.48.3. 示例

```
1.   rtcPeer.off("memberTalking").on("memberTalking", function (data) {
2.     console.log("memberTalking", data.member);
3.   });
```

## 4. IM 音视交互功能

提供在线用户之间的呼叫过程中的交互功能接口，通过接口可以实现发起呼叫、接收或拒绝呼叫，进入指定会议室等需求。

### 4.1. 发起呼叫 (rtcPeer.call)

#### 4.1.1. 概述

指定用户发起呼叫。

#### 4.1.2. 1.2 参数

参数	类型	必填	参数说明
targetUserId	number	是	用户 User 中的 id 值
ownDeviceType	String	否	设备类型 pc、Android、iOS、webphone、tv
roomId	number	否	呼叫后进入指定会议室

#### 4.1.3. 监听事件

参数	类型	必填	参数说明
doInvite	String	是	被呼叫者收到的邀请消息事件

事件获取数据

参数	类型	参数说明
ownUserId	number	发起呼叫者的用户唯一标识 Id
ownDeviceType	String	发起呼叫者的用户设备类型

#### 4.1.4. 示例

```

1.   rtcPeer.call({
2.     targetUserId: callUser.id
3.   }).then(function(d) {
4.     if (d.result == 1) {
5.       clearTimeout(ringTimeoutIndex);
6.       clearTimeout(ringCallbackIndex);
7.       showMessage("您呼叫的用户当前不在线。");
8.     } else if (d.result == 0) {
9.       room = {id:d.roomId,roomType:2,name:'即时会议',isCall:true};
10.      callRoom = room;
11.      if (!isOtherBusy) {
12.        callMessage(callUser);
13.      }

```

```

14.     clearTimeout(ringCallbackIndex);
15. }
16. });
17. //事件监听
18. rtcPeer.off("doInvite").on("doInvite", function(data) {
19.     console.info("doInvite", data);
20.     if (isAtRoom) {
21.         rtcPeer.tellBusy({
22.             'targetUserId': data.ownUserId,
23.             'targetDeviceType': data.ownDeviceType
24.         });
25.     } else {
26.         rtcPeer.ring({
27.             'targetUserId': data.ownUserId,
28.             'targetDeviceType': data.ownDeviceType
29.         });
30.     }
31. });

```

## 4.2. 同意呼叫请求 (rtcPeer.agree)

### 4.2.1. 概述

同意呼叫请求，并发送信息告诉呼叫。

### 4.2.2. 2.2 参数

参数	类型	必填	参数说明
targetUserId	number	是	呼叫者用户唯一标识码
targetDeviceType	String	否	设备类型 pc、Android、iOS、webphone、tv
ownDeviceType	String	否	设备类型 pc、Android、iOS、webphone、tv

### 4.2.3. 监听事件

参数	类型	必填	参数说明
doAgree	String	是	被呼叫者发送同意请求消息事件

事件获取数据

参数	类型	参数说明
ownUserId	number	被呼叫者的用户唯一标识 Id
ownDeviceType	String	被呼叫者的用户设备类型

#### 4.2.4. 示例

```

1.   rtcPeer.agree({
2.     'targetUserId': receiveUser.id,
3.     'targetDeviceType': receiveUser.deviceType
4.   });
5.   rtcPeer.off("doAgree").on("doAgree", function(data) {
6. });

```

### 4.3. 拒绝呼叫请求 (rtcPeer.reject)

#### 4.3.1. 概述

拒绝呼叫请求，并发送信息告诉呼叫。

#### 4.3.2. 参数

参数	类型	必填	参数说明
targetUserId	number	是	呼叫者用户唯一标识码
targetDeviceType	String	否	设备类型 pc、Android、iOS、webphone、tv
ownDeviceType	String	否	设备类型 pc、Android、iOS、webphone、tv

#### 4.3.3. 监听事件

参数	类型	必填	参数说明
reject	String	是	被呼叫者发送拒绝请求消息事件

事件获取数据

参数	类型	参数说明
ownUserId	number	被呼叫者的用户唯一标识 Id
ownDeviceType	String	被呼叫者的用户设备类型

#### 4.3.4. 示例

```

1.   rtcPeer.reject({
2.     'targetUserId': receiveUser.id,
3.     'targetDeviceType': receiveUser.deviceType
4.   });
5.   rtcPeer.off("reject").on("reject", function(data) {
6.     console.info("reject", data);
7.     if (callUser.id == data.ownUserId) {
8.       $("#callMessage").hide();

```

```

9.      $("#" + "#ringAudio")[0].src = "";
10.     clearTimeout(ringCallbackIndex);
11.     clearTimeout(ringTimeoutIndex);
12.   }
13. });

```

## 4.4. 取消呼叫请求 (rtcPeer.cancel)

### 4.4.1. 概述

取消呼叫请求，并发送信息告诉被呼叫者。

### 4.4.2. 4.2 参数

参数	类型	必填	参数说明
targetUserId	number	是	呼叫者用户唯一标识码
targetDeviceType	String	否	设备类型 pc、Android、iOS、webphone、tv
ownDeviceType	String	否	设备类型 pc、Android、iOS、webphone、tv

### 4.4.3. 监听事件

参数	类型	必填	参数说明
cancel	String	是	取消呼叫请求消息事件

事件获取数据

参数	类型	参数说明
ownUserId	number	被呼叫者的用户唯一标识 Id
ownDeviceType	String	被呼叫者的用户设备类型

### 4.4.4. 示例

```

1.   rtcPeer.cancel({
2.     'targetUserId': callUser.id
3.   });
4.   rtcPeer.off("cancel").on("cancel", function(data) {
5.     console.info("cancel", data);
6.     if (receiveUser.id == data.ownUserId) {
7.       $("#" + "#receiveMessage").hide();
8.       $("#" + "#ringAudio")[0].src = "";
9.       clearTimeout(ringCallbackIndex);
10.      clearTimeout(ringTimeoutIndex);
11.    }
12.  });

```

## 4.5. 响铃消息 (rtcPeer.ring)

### 4.5.1. 概述

被呼叫方接收到邀请事件 (doInvite) 后，被呼叫方当前处于空闲状态时，被呼叫方自动发送响铃消息，告诉呼叫方被呼叫方已响铃。

### 4.5.2. 5.2 参数

参数	类型	必填	参数说明
targetUserId	number	是	呼叫者用户唯一标识码
targetDeviceType	String	否	呼叫者设备类型 pc、Android、iOS、webphone、tv
ownDeviceType	String	否	设备类型 pc、Android、iOS、webphone、tv

### 4.5.3. 监听事件

参数	类型	必填	参数说明
doEvent	String	是	消息事件 event: onRinging 响铃事件 onBusy 正在忙事件 onConnected 连接成功事件，开始音视频连接

#### 事件获取数据

参数	类型	参数说明
ownUserId	number	被呼叫者的用户唯一标识 Id
ownDeviceType	String	被呼叫者的用户设备类型

### 4.5.4. 示例

```

1.   rtcPeer.ring({
2.     'targetUserId': data.ownUserId,
3.     'targetDeviceType': data.ownDeviceType
4.   });
5.   rtcPeer.off("doEvent").on("doEvent", function(data) {
6.     console.info("doEvent", data);
7.     switch (data.event) {
8.       case "onRinging":
9.         //clearTimeout(ringTimeoutIndex);
10.      break;
11.    }
12.  });

```

## 4.6. 忙消息 (rtcPeer.busy)

### 4.6.1. 概述

被呼叫方接收到邀请事件 (doInvite) 后，被呼叫方当前处于忙状态（正在开会中）时，被呼叫方自动发送正在忙消息，告诉呼叫方被呼叫方无法接听。

### 4.6.2. 6.2 参数

参数	类型	必填	参数说明
targetUserId	number	是	呼叫者用户唯一标识码
targetDeviceType	String	否	呼叫者设备类型 pc、Android、iOS、webphone、tv
ownDeviceType	String	否	设备类型 pc、Android、iOS、webphone、tv

### 4.6.3. 监听事件

参数	类型	必填	参数说明
doEvent	String	是	消息事件 event: onRinging 响铃事件 onBusy 正在忙事件 onConnected 连接成功事件，开始音视频连接

事件获取数据

参数	类型	参数说明
ownUserId	number	被呼叫者的用户唯一标识 Id
ownDeviceType	String	被呼叫者的用户设备类型

### 4.6.4. 示例

```

1.   rtcPeer.busy({
2.     'targetUserId': data.ownUserId,
3.     'targetDeviceType': data.ownDeviceType
4.   });
5.   rtcPeer.off("doEvent").on("doEvent", function(data) {
6.     console.info("doEvent", data);
7.     switch (data.event) {
8.       case 'onBusy':
9.         isOtherBusy = true;
10.        $("#callMessage").hide();
11.        $("#ringAudio")[0].src = "";
12.        clearTimeout(ringTimeoutIndex);
13.        clearTimeout(ringCallbackIndex);

```

```

14.     showMessage("您呼叫的用户当前正在忙。");
15.     break;
16. }
17. });

```

## 4.7. 连接成功消息 (rtcPeer.connected)

### 4.7.1. 概述

接收到被呼叫方接同意事件 (doAgree) 后，呼叫方进入会议并发送连接成功消息给被呼叫方，被呼叫方接收到 connected 事件后，进入指定会议。

### 4.7.2. 参数

参数	类型	必填	参数说明
targetUserId	number	是	呼叫者用户唯一标识码
targetDeviceType	String	否	呼叫者设备类型 pc、Android、iOS、webphone、tv
roomId	Number	是	虚拟房间唯一标识码
roomType	Number	是	房间类型：1：音频会议 2：视频会议 3：在线培训

### 4.7.3. 监听事件

参数	类型	必填	参数说明
doEvent	String	是	消息事件 event: onRinging 响铃事件 onBusy 正在忙事件 onConnected 连接成功事件，开始音视频连接

事件获取数据

参数	类型	参数说明
ownUserId	number	被呼叫者的用户唯一标识 Id
ownDeviceType	String	被呼叫者的用户设备类型

### 4.7.4. 示例

```

1.   rtcPeer.connected({
2.     'targetUserId': data.ownUserId,
3.     'targetDeviceType': data.ownDeviceType,
4.     'roomId': d.room.id,

```

```

5.      'roomType': d.room.roomType
6.    }).then(function(data) {
7.      clearTimeout(ringCallbackIndex);
8.      clearTimeout(ringTimeoutIndex);
9.      $("#callMessage").hide();
10.     $("#ringAudio")[0].src = "";
11.   });
12.   rtcPeer.off("doEvent").on("doEvent", function(data) {
13.     console.info("doEvent", data);
14.     switch (data.event) {
15.       case 'onConnected':
16.         clearTimeout(ringCallbackIndex);
17.         clearTimeout(ringTimeoutIndex);
18.         $("#ringAudio")[0].src = "";
19.         room = { id: data.roomId, name: '即时会议', isCall: true };
20.         callRoom = room;
21.         _enterRoom({
22.           id: data.roomId,
23.           roomType: data.roomType,
24.           name: '即时会议'
25.         });
26.         break;
27.     }
28.   });

```

## 4.8. 自定义事件消息 (rtcPeer.doEvent)

### 4.8.1. 概述

响铃消息、忙消息、连接成功消息都是通过自定doEvent 接口中event 的值来实现。

### 4.8.2. 参数

参数	类型	必填	参数说明
targetUserId	number	是	呼叫者用户唯一标识码
targetDeviceType	String	是	呼叫者设备类型 pc、Android、iOS、webphone、tv
roomId	Number	否	虚拟房间唯一标识码
roomType	Number	否	房间类型：1：音频会议 2：视频会议 3：在线培训
event	String	是	自定义消息事件名：如 onRinging

### 4.8.3. 监听事件

参数	类型	必填	参数说明
doEvent	String	是	消息事件 event: onRinging 响铃事件 onBusy 正在忙事件 onConnected 连接成功事件，开始音视频连接

事件获取数据

参数	类型	参数说明
ownUserId	number	被呼叫者的用户唯一标识 Id
ownDeviceType	String	被呼叫者的用户设备类型

## 4.9. 呼叫注册 (rtcPeer.doRegister)

### 4.9.1. 概述

登录后，调用呼叫注册接口，注册当前用户已在线，可以被呼叫,非SID对接方案不使用，类库内会自动在登录后注册。

### 4.9.2. 参数

参数	类型	必填	参数说明
userId	number	是	当前用户唯一标识码

### 4.9.3. 返回参数

参数	类型	参数说明
result	number	0 呼叫注册成功

## 5. IM 音视交互功能 Java 接口

提供在线用户之间的呼叫过程中的交互功能接口，通过接口可以实现发起呼叫、接收或拒绝呼叫，进入指定会议室等需求。在客户端通过建立socket连接控制服务器，实现消息的实时传输。

## 5.1. 呼叫接口

### 5.1.1. 概述

指定用户发起呼叫。

### 5.1.2. 参数

参数	类型	是否必选	参数说明
module	String	是	值 : call
method	String	是	方法名 : doInvite
ownUserId	Long	是	当前用户唯一标识
targetUserId	Long	是	被呼叫用户 User 中的 id (用户唯一标识)
ownDeviceType	String	否	设备类型 pc、Android、iOS、webphone、tv
targetDeviceType	String	否	被呼叫设备类型 pc、Android、iOS、webphone、tv
type	Long	否	1 音频会议 2 视频会议 3 视频回传
roomId	Long	是	会议室房间唯一标识，通过创建房间接口获取 (newTempRoom 接口)

### 5.1.3. 监听事件

参数	类型	必填	参数说明
doInvite	String	是	被呼叫者收到的邀请消息事件

事件获取数据

参数	类型	参数说明
ownUserId	number	发起呼叫者的用户唯一标识 Id
ownDeviceType	String	发起呼叫者的用户设备类型
type	number	1 音频会议 2 视频会议 3 视频回传

## 5.2. 同意呼叫请求接口

### 5.2.1. 概述

同意呼叫请求，并发送信息告诉呼叫。

### 5.2.2. 参数

参数	类型	必填	参数说明
module	String	是	值：call
method	String	是	方法名：doAgree
ownUserId	Long	是	当前用户唯一标识
targetUserId	Long	是	呼叫用户 User 中的 id (用户唯一标识)
ownDeviceType	String	否	设备类型 pc、Android、iOS、webphone、tv
targetDeviceType	String	否	呼叫设备类型 pc、Android、iOS、webphone、tv
type	Long	否	1 音频会议 2 视频会议 3 视频回传
roomId	Long	是	通过呼叫接口得到

### 5.2.3. 监听事件

参数	类型	必填	参数说明
doAgree	String	是	被呼叫者发送同意请求消息事件

事件获取数据

参数	类型	参数说明
ownUserId	number	被呼叫者的用户唯一标识 Id
ownDeviceType	String	被呼叫者的用户设备类型

## 5.3. 拒绝呼叫请求接口

### 5.3.1. 概述

拒绝呼叫请求，并发送信息告诉呼叫。

### 5.3.2. 参数

参数	类型	必填	参数说明
module	String	是	值：call
method	String	是	方法名：reject
ownUserId	Long	是	当前用户唯一标识
targetUserId	Long	是	呼叫用户 User 中的 id (用户唯一标识)
ownDeviceType	String	否	设备类型 pc、Android、iOS、webphone、tv
targetDeviceType	String	否	呼叫设备类型 pc、Android、iOS、webphone、tv

### 5.3.3. 监听事件

参数	类型	必填	参数说明
reject	String	是	被呼叫者发送拒绝请求消息事件

事件获取数据

参数	类型	参数说明
ownUserId	number	被呼叫者的用户唯一标识 Id
ownDeviceType	String	被呼叫者的用户设备类型

## 5.4. 取消呼叫请求接口

### 5.4.1. 概述

取消呼叫请求，并发送信息告诉被呼叫者。

### 5.4.2. 参数

参数	类型	必填	参数说明
module	String	是	值：call
method	String	是	方法名：cancel
ownUserId	Long	是	当前用户唯一标识
targetUserId	Long	是	呼叫用户 User 中的 id (用户唯一标识)
ownDeviceType	String	否	设备类型 pc、Android、iOS、webphone、tv
targetDeviceType	String	否	呼叫设备类型 pc、Android、iOS、webphone、tv

### 5.4.3. 监听事件

参数	类型	必填	参数说明
cancel	String	是	取消呼叫请求消息事件

事件获取数据

参数	类型	参数说明
ownUserId	number	被呼叫者的用户唯一标识 Id
ownDeviceType	String	被呼叫者的用户设备类型

## 5.5. 事件消息通用接口

### 5.5.1. 概述

响铃消息、忙消息、连接成功消息都是通过自定doEvent接口中event的值来实现。

### 5.5.2. 参数

参数	类型	是否必选	参数说明
module	String	是	值：call
method	String	是	方法名：doEvent
ownUserId	Long	是	当前用户唯一标识
targetUserId	Long	是	呼叫用户 User 中的 id (用户唯一标识)
ownDeviceType	String	否	设备类型 pc、Android、iOS、webphone、tv
targetDeviceType	String	否	呼 叫 设 备 类 型 pc 、 Android 、 iOS 、 webphone、tv
event	String	是	自定义事件名：onRinging 响铃事件 onBusy 正在忙事件 onConnected 连接成功事件，开始音视频连接

### 5.5.3. 监听事件

参数	类型	必填	参数说明
doEvent	String	是	消息事件 event: onRinging 响铃事件 onBusy 正在忙事件 onConnected 连接成功事件，开始音视频连接

事件获取数据

参数	类型	参数说明
ownUserId	number	被呼叫者的用户唯一标识 Id
ownDeviceType	String	被呼叫者的用户设备类型

## 5.6. 呼叫注册接口

### 5.6.1. 概述

登录后，调用呼叫注册接口，注册当前用户已在线，可以被呼叫。

### 5.6.2. 参数

参数	类型	必填	参数说明
module	String	是	值：call
method	String	是	方法名：doRegister
SID	String	是	Session ID 由认证接口成功后返回
userId	Long	是	当前用户唯一标识
deviceType	String	否	设备类型 pc、Android、iOS、webphone、tv

### 5.6.3. 返回参数

参数	类型	参数说明
result	Long	呼叫注册成功

## 5.7. 响铃消息

### 5.7.1. 概述

被呼叫方接收到邀请事件（doInvite）后，被呼叫方当前处于空闲状态时，被呼叫方自动发送响铃消息，告诉呼叫方被呼叫方已响铃。

## 5.8. 忙消息

### 5.8.1. 概述

被呼叫方接收到邀请事件（doInvite）后，被呼叫方当前处于忙状态（正在开会中）时，被呼叫方自动发送正在忙消息，告诉呼叫方被呼叫方无法接听。

## 5.9. 连接成功消息

### 5.9.1. 概述

接收到被呼叫方接同意事件（doAgree）后，呼叫方进入会议并发送连接成功消息给被呼叫方，被呼叫方接收到 connected 事件后，进入指定会议。

## 5.10. 创建临时房间接口

### 5.10.1. 概述

创建临时房间。

### 5.10.2. 参数

参数	类型	必填	参数说明
module	String	是	值：room
method	String	是	方法名：newAppRoom
name	String	是	房间名称
numberLimit	Long	是	会议室人员上限，最大 200 人
roomtype	Long	是	1 音频会议 2 视频会议 3 在线培训
userId	Long	是	当前用户唯一标识

### 5.10.3. 返回参数

参数	类型	参数说明
result	Long	0 成功
roomId	Long	会议室房间唯一标识

## 5.11. 登出接口

### 5.11.1. 概述

登出。

### 5.11.2. 参数

参数	类型	必填	参数说明
module	String	是	值：account
method	String	是	方法名：logout
SID	String	是	Session ID 由认证接口成功后返回
userId	Long	是	当前用户唯一标识

### 5.11.3. 返回参数

参数	类型	参数说明
result	Long	0 成功

## 5.12. 创建预约会议

### 5.12.1. 概述

创建预约会议接口, 创建即时会议, 将开始时间设置为当前时间。

### 5.12.2. 参数

参数	类型	是否必选	参数说明
method	String	是	方法名, 值 : new
module	String	是	值 : meeting
SID	String	是	Session ID 用户登录后产生的身份标识码
name	String	是	会议名称
startTime	String	是	会议开始时间 , 传入 UTC 时间格式 “yyyyMMddHHmmss”
endTime	String	是	会议结束时间 , 传入 UTC 时间格式 “yyyyMMddHHmmss”
hour	String	否	会议时长 (小时)
leverage	String	否	会议容量人数 (自然数, 200 方) ; 主要看流媒体服务器的 CPU
type	Long	是	会议类型 : 1 音频会议 2 视频会议 3 视频回传
isPublic	Boolean	是	是否公开, true 公共会议, false: 私人会议。建议传 false,
pin	Long	否	会议进入校验密码
members	Array	否	邀请成员, 数组内成员格式为 String, 格式如下 : “ memberId,name,email,id, Etc/GMT+8,role”

			注：memberId 默认为 0， id 为用户的唯一标识码 id, name:用户昵称 email:邮件，可以为 null 角色 role: 会议场模式：1 是主持人 2 普通参会人员； 培训模式：1 是主持人 2 是讲师，3 普通参付人 员，建议只邀请主持人和讲师
template	Long	否	模板
postUrl	String	否	海报
orgId	Long	是	公司唯一标识码 Id，可以提供，一般情况传 2
roomId	Long	否	房间唯一标识 id 不传 id 会重新创建一个，传存在的 room id， 用此 room
baseURL	String	否	邮件链接地址
topics	Array	否	网络会议子议题 数组内的 Objet 格式： { "title":“议题”， "content": "具体内容" }

### 5.12.3. 返回参数

参数	类型	参数说明
id	Int	会议唯一标识
roomId	Int	房间的唯一标识，新建会议，如果不传 roomId,会议和房间的关系是一对一。
result	Int	0：成功， 1：失败

## 6. 轻会议平台对接方案

第三方平台和私有部署轻会议平台对接方案。

### 6.1. 生成签名接口

- 1) 轻会议给第三方平台配置 appid 值
- 2) 第三方平台通过 appid 和 secret 到轻会议平台拿签名（轻会议平台通过 appid 和 secret 加密保存返回），第三方平台调用轻会议平台 api,必需要有签名才能调用

## 6.1.1. 生成签名接口

### 6.1.1.1. 请求格式

请求地址	https://xxx.xxx.com/app/getSign
请求模式	GET
Content-Type	multipart/form-data

### 6.1.1.2. 参数

参数	类型	是否必选	参数说明
appId	String	是	公司接入 api 认证 key (由轻会议提供)
secret	String	是	钥匙

### 6.1.1.3. 返回参数

参数	类型	参数说明
code	int	200
data	String	api 签名
message	String	描述

### 6.1.1.4. 入参格式

https://xxx.xxx.com/app/getSign?appId=56789537388&secret=kksakjkjkk11

## 6.2. 用户信息认证接口

### 6.2.1. 概述

- 用户信息对接，具体：
- 调用轻会议平台 getSign 接口生成签名；
  - 调用轻会议平台 dotokenLogin 接口进行认证，认证成功返回用户信息；
  - 第三方平台可以通过 SID 和用户唯一标识调用轻会议平台相关接口；
- 注：第三方提供认证接口，认证成功返回第三方用户信息。

### 6.2.2. dotokenLogin 接口

#### 6.2.2.1. 请求格式

请求地址	https://gj.527meeting.com/app/account
请求模式	POST

<b>Content-Type</b>	application/x-www-form-urlencoded
---------------------	-----------------------------------

### 6.2.2.2. 参数

参数	类型	是否必选	参数说明
method	String	是	方法名，值：dockingLogin
avatar	String	否	第三方用户头像
account	String	是	第三方用户唯一标识
userName	String	否	第三方用户名
appkey	String	否	公司接入 api 认证 key (由轻会议提供)
sign	String	是	api 签名，有签名才能调用，通过 java 接口可获取
job	String	否	职位
level	Long	否	用户级别

### 6.2.2.3. 返回参数

参数	类型	参数说明
SID	String	Session ID 用户登录后产生的身份标识码
organisation	Company	公司信息，参考 been 类中的 Company。
user	User	用户信息，参考 been 类中的 User.
result	Long	0 成功；1 公司名不正确；2 公司 appkey 验证失败；3 回调请求失败；4 注册失败

### 6.2.2.4. 入参格式

```
1. "params": {"avatar": "https://static.oschina.net/new-osc/img/osc_app_download_qrcode.png?t=1451964198000", "account": "test212", "userName": "名字", "appkey": "jsadoijdfoja!iod@sfj", "method": "dockingLogin"}
```

POST https://app/account

Header Query Body 认证 预执行脚本 后执行脚本 Mock 服务

参数名 参数值 必填 类型 参数描述

params	{"avatar": "https://static.oschina.net/n...ew-osc/img/osc_app_download_qrcode.png?t=1451964198000", "accou...": "test212", "userName": "名字", "appkey": "jsadoijdfojaliod@sfj", "metho...	<input checked="" type="checkbox"/>	Text	参数描述,用于生成文档
参数名		<input checked="" type="checkbox"/>	Text	参数描述,用于生成文档

实时响应 请求头(6) 响应头(9) Cookie(1) 成功响应示例 错误响应示例 14:44:31 响应码: 200 396.00ms 1.20kb

美化 原生 预览 断言 可视化 ⌂ ⌂ 绑定响应结果到变量?

```

1 {
2   "result": 0,
3   "method": "dockingLogin",
4   "module": "account",
5   "organisation": {
6     "totalNumber": 8,
7     "city": null,
8     "pictureurl": "./img/login.png",
9     "icon": null,
10    "creatorName": "Admin",
11    "orgAddress": null,
12    "title": null,
13    "addOpened": true,
14    "creatorUID": 2,
15    "contactOpened": true,
16    "loginStr": "?".

```

## 7. Web Url 接口

登录用户或非登录用户可以通过 url 链接跳转到会议室中。

### 7.1. 通过房间 ID 进入指定房间

#### 7.1.1. 概述

通过房间 ID,进入指定房间,登录用户传有效 SID,非登录用房不传。

#### 7.1.2. 请求格式

https:// 61.182.226.42:10443/app/#?role=1&rd= 6656&name=小明

#### 7.1.3. 参数

参数	类型	是否必选	参数说明
SID	String	是	Session ID 用户登录后产生的身份标识码
uid	String	否	用户唯一标识, 登录用户传
role	number	否	角色 role: 会议场模式 : 1 是主持人 2 普通参会人员 ;

			培训模式：1 是主持人 2 是讲师， 3 普通参付人员
rd	number	是	房间唯一标识码
v	number	否	进入会议室默认：1:摄像头打开， 0：摄像头不打开
name	String	否	参会人员昵称

## 7.2. 通过会议 ID 进入指定房间

### 7.2.1. 概述

通过会议 ID,进入指定房间,登录用户传有效 SID,非登录用房不传。

### 7.2.2. 请求格式

<https://61.182.226.42:10443/app/#/?role=1&code=6656&name=小明>

### 7.2.3. 参数

参数	类型	是否必选	参数说明
SID	String	是	Session ID 用户登录后产生的身份标识码
uid	String	否	用户唯一标识，登录用户传
role	number	否	角色 role: 会议场模式：1 是主持人 2 普通参会人员； 培训模式：1 是主持人 2 是讲师， 3 普通参付人员
code	number	是	会议唯一标识码
v	number	否	进入会议室默认：1:摄像头打开， 0：摄像头不打开
name	String	否	参会人员昵称

## 8. Bean 类说明

对文中的 Object 的对象补充说明。

### 8.1. User

参数	类型	参数说明
id	Number	用户唯一标识码
account	String	账号
name	String	昵称
email	String	邮件地址
phone	String	手机号
photo	String	头像
company	String	公司名称

### 8.2. Company

参数	类型	参数说明
id	Number	公司唯一标识码
name	String	公司名
email	String	公司邮件地址
phone	String	公司手机号
logo	String	公司 Logo
address	String	公司地址

### 8.3. Room

参数	类型	参数说明
id	Number	唯一标识码
name	String	房间名
ownerId	Number	房间拥有者用户唯一标识码
roomType	Number	房间类型 1 音频 2 视频 3 培训
definition	String	清晰度 1080p 720p 480p 240p
numberLimit	Number	房间人数上限
isPublic	Boolean	true 公共会议室 false 私有会议室
isHasPwd	Boolean	是否有密码

appointmentId	Number	房间对应的预约会议唯一标识，仅在会议中使用
appointmentName	String	对应会议的名称，仅在会议中使用
superModeId	Number	仅在会议中使用,会议中的管理员的唯一标识
logo	String	Logo
comment	String	描述

## 8.4. Client

参数	类型	参数说明
id	Number	用户唯一标识码
name	String	昵称
email	String	邮件地址
phone	String	手机号
pictureurl	String	头像
company	String	公司名称
mode	String	终端类型
streamId	Number	用户唯一标识码
broadCastId	number	用户唯一标识码 流媒体使用
isMuted	Boolean	是否是静音状态
isSuperModerator	Boolean	是否是主持人

## 8.5. Appointment

参数	类型	参数说明
id	Number	唯一标识码
name	String	会议名称
startTime	Date	开始时间
endTime	Date	结束时间
from	User	创建者
timeLength	Number	分钟
description	String	描述
streamId	Number	用户唯一标识码
broadCastId	number	用户唯一标识码 流媒体使用
actuallyStartTime	Date	真实开始时间
actuallyEndTime	Date	真实结束时间