

Gensee, Inc

Android Client RTSDK

目录

Android Client RTSDK	1
1 修订记录	9
2 文档介绍	12
2.1 文档目的	12
3 SDK 使用准备	13
3.1 开发环境要求	13
3.2 需要导入的 jar	13
3.3 需要添加的动态库	13
3.4 配置与信息	14
3.4.1 权限(targetSdkVersion>22 时候请动态申请)	14
3.4.2 version	14
3.4.3 OnTaskRet 辅助接口	14
3.4.4 InitParam 说明	14
3.4.5 Userinfo 说明	14
3.4.6 混淆打包事项	15
3.5 快速接入 gensee 直播间（课堂）	15
3.5.1 使用 RtSimpleImpl 加入直播间（课堂）	15
3.5.2 RtSdk 扩展更多功能	16
4 回调（代理）响应	18
4.1 初始化响应回调 RtComp.Callback	18
4.1.1 onInit(String joinParam)	18
4.1.2 onErr(int errCode)	18
4.2 直播间响应回调 IRoomCallBack	18
4.2.1 onInit(boolean result)直播间初始化响应	19
4.2.2 onRoomJoin(int result, UserInfo self, boolean svrFailover)直播间加入响应	19
4.2.3 onRoomLeave(int reason)直播间退出响应	19
4.2.4 onRoomReconnecting()直播间重连通知	19
4.2.5 onRoomLock(boolean locked)直播间锁定通知	19
4.2.6 onRoomUserJoin(UserInfo userInfo)直播间新用户加入通知	20
4.2.7 onRoomUserUpdate(UserInfo userInfo)直播间用户信息变化通知	20
4.2.8 onRoomUserLeave(UserInfo userInfo)直播间用户退出通知	20
4.2.9 onRoomPublish(State s)直播状态通知	20
4.2.10 onRoomRecord(State s)录制状态通知	20
4.2.11 onRoomData(String key, long value)直播间数据通知	20
4.2.12 onRoomBroadcastMsg(String msg)直播间广播消息	20
4.2.13 onRoomRollcall(int timeout)直播间点名	20
4.2.14 onRoomRollcallAck(long userId)用户应答点名后组织者收到的通知	20
4.2.15 onRoomHandup(long userId, String data)直播间用户举手通知	20
4.2.16 onRoomHanddown(long userId)直播间用户取消举手的通知	20
4.2.17 getServiceType()返回站点类型	20
4.2.18 onGetContext()返回 app context	21
4.2.19 onChatMode(int chatMode)聊天模式切换通知	21

4.2.20	onLottery(byte step, String userlist)抽奖流程通知	21
4.2.21	onSettingSet(String key, int val)直播间 int 类配置信息	21
4.2.22	onSettingSet(String key, String val)直播间 String 类配置信息.....	22
4.2.23	int onSettingQuery(String key, int val)直播间取回 int 类配置信息	22
4.2.24	String onSettingQuery(String key)直播间取回 String 类配置信息	22
4.2.25	onNetworkReport(byte level)网络质量通知	22
4.2.26	onNetworkBandwidth(int bpsSend, int bpsRecv) 带宽通知	22
4.2.27	onLiveInfo(LiveInfo info)直播简介等信息通知.....	22
4.2.28	onRoomWebLayoutChange(int weblayout)web 播放器布局切换通知	23
4.3	音频回调 IAudioCallBack	23
4.3.1	onAudioJoinConfirm(boolean ok)音频模块加载通知	23
4.3.2	onAudioMicAvailable(boolean isAvailable)mic 设备“可用”通知.....	23
4.3.3	onAudioMicOpened()mic 被打开通知	23
4.3.4	onAudioMicClosed()mic 被关闭通知.....	23
4.3.5	onAudioSpeakerOpened()扬声器被打开通知	23
4.3.6	onAudioSpeakerClosed()扬声器被关闭通知	23
4.3.7	onGetContext()返回 appContext	23
4.3.8	onAudioLevel(int level, long userId)用户扬声器电频通知	23
4.4	视频回调 IVideoCallBack	23
4.4.1	onVideoJoinConfirm(boolean ok)视频模块加载通知	23
4.4.2	onVideoCameraAvailable(boolean ok)摄像头可用通知.....	24
4.4.3	onVideoCameraOpened()摄像头被打开通知	24
4.4.4	onVideoCameraClosed()摄像头被关闭通知.....	24
4.4.5	onVideoJoin(UserInfo user)用户视频开启通知.....	24
4.4.6	onVideoLeave(long userId)用户视频关闭通知	24
4.4.7	onVideoActivated(UserInfo user, boolean bActivated)直播视频启用通知	24
4.4.8	onVideoDisplay(UserInfo user)观看 user 视频响应	24
4.4.9	onVideoUndisplay(long userId)取消观看 userId 视频响应.....	24
4.4.10	onVideoDataRender(long userId, int width, int height,int frameFormat, float displayRatio, byte[] data)视频数据流通知	24
4.5	桌面共享回调 IAsCallBack	24
4.5.1	onAsJoinConfirm(boolean ok)桌面共享模块加载通知	24
4.5.2	onAsBegin(long owner)桌面共享开启通知	24
4.5.3	onAsEnd()桌面共享结束通知	24
4.5.4	onAsData(byte[] data, int width, int height)桌面共享数据流通知	24
4.6	插播（媒体共享）回调 ILodCallBack	25
4.6.1	onLodFailed(String id)插播（媒体共享）失败通知	25
4.6.2	onLodStart(LiveodItem liveodItem) 插播（媒体共享）开始通知	25
4.6.3	onLodSkip(LiveodItem liveodItem) 插播（媒体共享）跳跃通知.....	25
4.6.4	onLodPause(LiveodItem liveodItem); 插播（媒体共享）暂停通知	25
4.6.5	onLodStop(LiveodItem liveodItem); 插播（媒体共享）停止通知	25
4.6.6	onLodPlaying(LiveodItem liveodItem); 插播（媒体共享）播放中	25
4.6.7	onLodAdd(LiveodItem liveodItem); 插播（媒体共享）被添加通知	25
4.6.8	onLodRemove(String id); 插播（媒体共享）被移除通知.....	25

4.7	聊天回调 IChatCallBack.....	25
4.7.1	onChatJoinConfirm(boolean bRet)聊天模块加载通知	26
4.7.2	onChatMessage(ChatMsg msg)聊天消息回调	26
4.7.3	onChatEnable(boolean enable)聊天禁言通知.....	27
4.7.4	onChatCensor(long userId, String msgId)聊天审核删除通知.....	27
4.8	投票、试卷、答题卡回调 IVoteCallBack.....	27
4.8.1	onVoteJoinConfirm(boolean bRet)投票模块加载通知	28
4.8.2	onVoteAdd(VoteGroup voteGroup)投票被添加通知	28
4.8.3	onVoteDel(VoteGroup voteGroup)投票被删除通知	28
4.8.4	onVotePublish(VoteGroup voteGroup)投票发布通知	28
4.8.5	onVotePublishResult(VoteGroup voteGroup)投票发布结果通知.....	29
4.8.6	onVoteSubmit(VoteGroup voteGroup)投票被回复通知	29
4.8.7	onVoteDeadline(VoteGroup voteGroup)投票结束通知.....	29
4.8.8	onVotePostUrl(String strURL, long optUserId)第三方投票通知	29
4.8.9	onCardPublish(Card card)答题卡发布通知	29
4.8.10	onCardSubmit(long userId, int[] itemIds)答题卡答题通知	29
4.8.11	onCardEnd()答题结束通知	29
4.8.12	onCardResultPublish(int type, int totalSubmitted, int[] itemSubmitted, int[] itemOptions)答题卡结果通知	29
4.9	问答回调 IQACallback.....	30
4.9.1	onQaJoinConfirm(int isOk) 问答模块加载通知	30
4.9.2	onQaSettingUpdate(boolean isAutoDispatchQuestion,boolean isAutoPublishQuestion, boolean isQAEnable) 问答权限通知.....	30
4.9.3	onQaQuestion(QaQuestion qaQuestion, int state) 问答通知	30
4.10	GSDocViewGx 事件回调接口 OnDocViewEventListener	31
4.10.1	onDoubleClicked(IGSDocView docView)	31
4.10.2	onSingleClicked(IGSDocView docView)	31
4.10.3	onEndHDirection(IGSDocView docView,int type,int eventType)	31
4.11	文档事件回调 IDocCallBack.....	31
4.11.1	onDocJoinConfirm 模块加载通知.....	31
4.11.2	onDocOpened(PduDoc doc)文档打开	31
4.11.3	onDocClosed(int docId)文档关闭通知.....	31
4.11.4	onDocGotoPage(int docId, int pageId,int aniStep)翻页	32
4.11.5	onDocPageReady(int docId, PduPage page, byte[] data)文档某页数据响应 ..	32
4.11.6	onDocRemoveAnno(int docId, int pageId, long annold)移除标注.....	32
4.11.7	onDocPageSize(int width, int height)当前页的分辨率.....	32
4.11.8	onDocUploadStatus(int docId, int status)文档(jpeg 图片上传结果通知)	32
4.12	点赞勋章事件回调 IMedalPraiseCallback	33
4.12.1	onPraiseInfo(int result,String praiseType,PraiseUserInfo info)点赞或勋章信息通知	33
4.12.2	onPraiseRecvList(int result,String praiseType,PraiseInfo[] praises)勋章排行 ..	33
4.12.3	onPraiseNotify(String praiseType, int time,PraiseUserInfo senderInfo,PraiseUserInfo receiverInfo,String comment)勋章或点赞变更通知	33
5	RTSDK API	35

5.1	RtComp.....	35
5.1.1	initWithGensee(InitParam p) 直播参数初始化	35
5.2	RtSdk	36
5.2.1	initWithParam(String logFileName, String joinParam,IRoomCallBack roomCallBack) 直播间初始化.....	36
5.2.2	initWithLaunchCode(String logFileName, String launchCode,IRoomCallBack roomCallBack) 直播间初始化.....	36
5.2.3	join(OnTaskRet taskRet) 加入直播间.....	36
5.2.4	leave(boolean isClose, OnTaskRet taskRet) 退出直播间.....	37
5.2.5	release(OnTaskRet taskRet) 销毁本地直播间资源	37
5.2.6	getAllUsers()获取用户列表	37
5.2.7	getUserById(long userId) 获取用户	37
5.2.8	getSelfUserInfo()获取个人用户信息	37
5.2.9	roomHandup(String data, OnTaskRet taskRet) 直播间举手	37
5.2.10	roomHanddown(boolean isAll, OnTaskRet taskRet) 直播间取消举手.....	37
5.2.11	setVideoCallBack(IVideoCallBack videoCallBack) 设置视频回调句柄	38
5.2.12	setLocalVideoView(ILocalVideoView localVideoView) 设置本地视频采集的 UI 38	
5.2.13	displayVideo(long userId,OnTaskRet onTaskRet) 观看某用户视频	38
5.2.14	unDisplayVideo(long userId, OnTaskRet onTaskRet) 取消观看某用户视频 ...	38
5.2.15	setAsCallBack(IAasCallBack asCallBack) 设置桌面共享回调句柄	38
5.2.16	setLodCallBack(ILodCallBack lodCallBack) 设置插播（媒体共享）回调句柄	38
5.2.17	setAudioCallback(IAudioCallBack audioCallBack) 设置音频回调句柄	38
5.2.18	audioOpenMic(OnTaskRet taskRet)打开设备 mic.....	38
5.2.19	audioCloseMic(OnTaskRet taskRet)关闭设备 mic	38
5.2.20	audioOpenSpeaker(OnTaskRet taskRet)打开设备扬声器	39
5.2.21	audioCloseSpeaker(OnTaskRet taskRet)关闭扬声器	39
5.2.22	openUserAudio(long userId, OnTaskRet taskRet) 打开用户音频	40
5.2.23	closeUserAudio(long userId, OnTaskRet taskRet) 关闭用户音频.....	40
5.2.24	videoOpenCamera(OnTaskRet taskRet)打开本地摄像头.....	40
5.2.25	videoCloseCamera(OnTaskRet taskRet)关闭本地摄像头.....	40
5.2.26	videoActive(long userId, boolean isActive, OnTaskRet taskRet)设置直播视频	41
5.2.27	setChatCallBack(IChatCallBack chatCallBack)设置聊天回调	41
5.2.28	chatWithPublic(ChatMsg msg, OnTaskRet taskRet)发送公共聊天消息.....	41
5.2.29	chatWithPersion(ChatMsg msg,OnTaskRet taskRet)发送私人聊天消息	42
5.2.30	setVoteCallBack(IVoteCallBack voteCallBack)设置投票、试卷回调句柄.....	42
5.2.31	voteAdd(VoteGroup voteGroup, final OnTaskRet taskRet)添加一个投票.....	42
5.2.32	voteDel(String strVoteID, OnTaskRet taskRet)删除一个投票	42
5.2.33	votePublish(String strVoteID, OnTaskRet taskRet)发布投票	42
5.2.34	votePublishResult(String strVoteID, OnTaskRet taskRet)发布投票结果	42
5.2.35	voteSubmit(VoteGroup voteGroup, OnTaskRet taskRet)提交（参与）投票 ...	42
5.2.36	voteDeadline(String strID, OnTaskRet taskRet)终止投票	42
5.2.37	votePostUrl(String strURL, OnTaskRet taskRet)第三方投票	42
5.2.38	cardSubmit(int[] itemIds, OnTaskRet taskRet)提交答题卡	42

5.2.39	setDocView(GSDocView docView)设置文档显示的 UI	42
5.2.40	setGSDocViewGx(GSDocViewGx docView) 设置支持 ppt 动画的显示文档的 UI 43	
5.2.41	getDocModule()获取文档模块操作句柄	43
5.2.42	setQACallback(IQACallback qaCallback)设置问答回调	43
5.2.43	qaAddQuestion(String question, OnTaskRet taskRet) QA 提问	43
5.2.44	qaAddAnswer(String questId, String answer, OnTaskRet taskRet) QA 回复 ..	43
5.2.45	qaPublish(String questId,boolean isPublish,OnTaskRet taskRet) QA 发布	43
5.2.46	qaPush2Panelist(String questId, OnTaskRet taskRet) QA 推送嘉宾	43
5.2.47	qaAssign2Panelist(String questId, long userId, OnTaskRet taskRet) QA 指派给 嘉宾 43	
5.2.48	qaSetReplyByVoice(String questId, OnTaskRet taskRet) 将提问置为语音回复 43	
5.2.49	qaNowReplyingByVoice(String questId, boolean isNowReplying,OnTaskRet taskRet) 将问题置为语音回复状态	43
5.2.50	qaNowReplyingByText(String questId, int duration, OnTaskRet taskRet) 将问 题为正在进行文本回复	43
5.2.51	qaSetState(boolean isAutoPushQuestion, boolean isAutoPublishQuestion, boolean isQAEnable, OnTaskRet taskRet) 问答权限通知	44
5.2.52	loadLibrarys() 加载动态库（提高加入速度）	44
5.2.53	roomRollcall(int duration, OnTaskRet taskRet) 发起点名	44
5.2.54	roomRollcallAck(OnTaskRet taskRet) 点名应答	44
5.2.55	roomSaveRollcallResult(OnTaskRet taskRet) 发布点名统计	44
5.2.56	roomPublish(byte state, OnTaskRet taskRet) 直播控制	44
5.2.57	roomRecord(byte state, OnTaskRet taskRet) 录制控制	44
5.2.58	roomRename(long userId, String newName, OnTaskRet taskRet)修改用户名 44	
5.2.59	roomGrantHost(long userId, OnTaskRet taskRet) 设置组织者（老师）	45
5.2.60	roomGrantPresentor(long userId, OnTaskRet taskRet)设置主讲	45
5.2.61	roomPushUserCount2Web(boolean isPush, OnTaskRet taskRet)向 web 用户推 送在线人数	45
5.2.62	roomPushUserList2Web(boolean isPush, OnTaskRet taskRet)向 web 用户推送 用户列表 45	
5.2.63	roomNotifyBroadcastMsg(String msg, boolean isNeedCache,OnTaskRet taskRet) 发送广播消息	45
5.2.64	roomSetData(String key, long value, OnTaskRet taskRet) training 课堂席位、状 态等控制 45	
5.2.65	roomEjectUser (long userId, boolean isForbidIp, OnTaskRet taskRet)踢人	45
5.2.66	getIDCs() 优选网络选择列表	45
5.2.67	getCurlIDC()获取当前正在使用的网络 id	46
5.2.68	setCurlIDC(String idcId, OnTaskRet taskRet)设置新的网络 id	46
5.2.69	roomOpenUserVideo(long userId,OnTaskRet taskRet)打开他人视频命令	46
5.2.70	roomCloseUserVideo(long userId,OnTaskRet taskRet)关闭他人视频命令	46
5.2.71	getLiveInfo()获取直播简介、主讲、议程	46

5.2.72	initMedalPraise(IMedalPraiseCallback callback,int defMedal,int defFavour)点赞勋章初始化.....	46
5.2.73	sendPraise(String praiseType, long toUserId, String toUserName, String comment, OnTaskRet taskRet)发送勋章或点赞.....	47
5.2.74	getPraiseRecvList(String praiseType, int maxUser, OnTaskRet taskRet)获取勋章排行	47
5.3	GSVideoView	47
5.3.1	setDefColor(int color)视频显示设置.....	47
5.3.2	onReceiveFrame(byte[] data, int width, int height)视频显示 UI 接收数据	47
5.3.3	renderDefault()视频 UI 清屏.....	47
5.3.4	setRenderMode(RenderMode renderMode)设置视频显示模式.....	47
5.3.5	getRenderMode()获取视频显示模式.....	48
5.3.6	renderDrawble(Bitmap defBitmap,boolean isFillXY)渲染默认图片	48
5.3.7	getVideoWidth()获取视频宽度	48
5.3.8	getVideoHeight()获取视频高度	48
5.4	GSDocViewGx (GSDocView)	48
5.4.1	setPaintColor(int color)设置文档画笔工具的颜色	48
5.4.2	setStrokeWidth(LINE_SIZE width)设置文档画笔工具的线粗	48
5.4.3	setAnnoMakeType(DrawMode mode)设置即将开始的文档标注类型	48
5.4.4	setCtrlMode(CtrlMode mode)设置文档 UI 的控制模式	49
5.4.5	setOnDocViewClickedListener(OnDocViewEventListener l).....	49
5.4.6	undo(long userId)文档注释撤销	49
5.4.7	showFillView()文档内容充满 docview 显示.....	49
5.4.8	showAdaptView()文档内容适应 docview 显示	49
5.4.9	showSourceScall()文档内容原始大小显示	49
5.4.10	showAdaptViewWidth()文档内容适应 docview 宽度显示	49
5.4.11	showAdaptViewHeight()文档内容适应 docview 高度显示	49
5.4.12	setBackgroundColor(int color)设置文档背景色	49
5.4.13	forbidZoomGestrue(boolean isFobid)手势禁用	49
5.5	IDocModule 文档操作句柄	50
5.5.1	getDocs()获取文档列表.....	50
5.5.2	getCurrentDoc() 获取当前显示的文档.....	50
5.5.3	isCurrentDoc(PduDoc doc)	50
5.5.4	gotoPage(PduPage page,boolean syn,OnTaskRet taskRet)翻页	50
5.5.5	openDoc(String name, Bitmap bitmap)打开、上传文档	50
5.5.6	synCurrentDoc()同步文档.....	50
5.5.7	getPhotoCount()图片、照片计数	50
5.6	RtSimpleImpl	51
5.6.1	setVideoView(GSVideoView)设置视频显示的 UI.....	51
5.6.2	setDocView(GSDocView docView)设置文档显示的 UI	51
5.6.3	setGSDocViewGx(GSDocView docView) 设置支持 ppt 动画的文档显示控件	51
5.6.4	join(String logFileName,String joinParam)加入直播间.....	51
5.6.5	leave(boolean isClose)退出直播间	51
5.6.6	onVideoStart()视频开始, 视频加载成功.....	51

5.6.7	onVideoEnd() 视频结束	51
5.6.8	onGetContext()	51
5.6.9	RtSdk getRtSdk() 获取 RtSdk 句柄	51
5.7	ChatEditText 聊天输入框	51
5.7.1	insertAvatar(String avatar, int resId) 向输入框插入一个表情	51
5.8	MyTextViewEx 聊天消息富文本显示	52
5.8.1	setRichText(String richText) 富文本显示	52
5.9	GSLocalVideoView(LocalVideoViewEx) 视频采集预览 UI	52
5.9.1	doCameraSwitch 切换前后摄像头	52
5.9.2	setOrientation(int orientation) 设置视频采集模式	52
5.9.3	setVideoSize(int width, int height) 设置采集分辨率	52
5.9.4	switchBeauty(boolean isOpen) 美颜切换	52
5.10	GenseeLog GenseeSDK 日志工具	53
5.10.1	reportDiagonse(Context context, String desc, ServiceType serviceType) 上传日志	53
6	代码实例	54
6.1	Domain 等参数初始化代码实例	54
6.2	响应处理	54
6.2.1	RtComp.Callback onInit	54
6.2.2	IRoomCallBack onInit	54
6.2.3	IVideoCallBack onVideoJoin	55
6.2.4	IVideoCallBack onVideoJoin	55
6.2.5	IVideoCallBack onVideoDataRender	55
6.2.6	IAasCallBack onAsData	56
6.3	RtSimpleImpl	56

1 修订记录

修订日期	版本号	描述	修订人
2014-12-15	draft	■ 初稿	Gao Li
2015-03-09	1.0	补充 qa 的接口描述	Gao Li
2015-07-04	1.0	补充 initparam 和 userinfo 的描述; 增加 gensee-log.so	Gao.Li
2015-07-09	1.0	优化部分设备 sdcard 不存在或损坏时候文档显示问题	Gao.Li
5015-07-28	1.0	1、公共聊天接口 chatWithPublic(String sPublicMsg, OnTaskRet taskRet) 变更为 chatWithPublic(String chatText, String richText, OnTaskRet taskRet); 2、私人聊天接口 chatWithPersion(String sPersionMsg, long userId, OnTaskRet taskRet) 变更为 chatWithPersion(String chatText, String richText, long userId, OnTaskRet taskRet) 其中 chatText 和 richText 从 ChatEditText 里面 getChatText() 和 getRichText() 获取	Gao.Li
5015-08-14	3.1	1、RtSdk 增加一个静态函数 loadLibrarys (5.2.50) 2、补充代码混淆事项 (3.4.6)	Gao.Li
2015-09-14	3.1	1、修改 emoji 表情导致的 crash 2、增加支持 ppt 动画的显示控件 GSDocViewGx (5.2.39, 5.6.3) 3、增加 GSDocViewGx 设置接口 (5.4)	Gao.Li
2015-10-19	3.1	增加 5.2.53-5.2.62 的接口描述(直播、录制、点名、授权、广播消息)	Gao.Li
2015-12-11	3.2	增加课堂席位状态操作 (5.2.63) 增加禁用文档手势说明 (5.4.13) 增加踢人操作 (5.2.64) 及 ip 被封禁时, 加入和被踢响应码 (4.2.2-3)	Gao.Li
2016-02-03	3.2	去掉原 4.2.19 onRoomSubject, 启用现有 4.2.22 onSettingSet 代替 增加 onSettingSet 房间配置存储 (4.2.21, 4.2.22) 增加 onSettingQuery 取回配置信息 (4.2.23, 4.2.24)	Gao.Li
2016-03-15	3.2	直播增加第三方认证 ininParam.setK, 如果启用认证, 请填写有效的 K, 没启用忽略 增加网络质量回调 (4.2.25) 增加带宽回调 (4.2.25)	Gao.Li
2016-07-11	3.3	增加打开、关闭他人视频命令接口 (5.2.68 5.2.69) 增加视频预览 LocalVideoViewEx (5.9)	Gao.Li
2016-11.14	3.3.2	增加美颜以及 9:16 和 16:9 的采集模式 (5.9, 5.9.4) 增加文档响应事件 (4.11 IDocCallback) 说明 增加日志上传 (5.10)	Gao.Li
2017-05-04	3.5	1、统一 (修改) 聊天接口 IChatCallback 及回调参数, 个人发的私聊不再回调 2、聊天消息增加消息 id, 修改聊天发送参数 ChatMsg	Gao.Li

		3、增加聊天审核回调接口 4、增加 IVoteCallback 回调答题卡回调 5、增加了 libucrtpt.so 动态库 6、改进文档（图片）上传的参数为 bitmap	
2017-9-14	3.6	1、增加直播简介获取（5.2.70）和响应（4.2.27）	
2018-7-4	3.6.3	1、增加点赞勋章回调 4.12，接口 5.2.71-73	
2019-5-14	3.7.3	1、增加 web 布局控制回调（4.2.28）	

2 文档介绍

2.1 文档目的

众多客户希望能拥有更多的 app 设计能力, 特别是对于希望拥有自己风格的 UI, 开发人员资源的公司。为此, 特提供 android 实时 SDK, 以便客户可以针对自己的业务场景设计最符合自己业务场景的客户端。

本 sdk 是实时的, 与 web 有本质的区别。

其中 sdk 中对一路视频的实现, 如果只需要观看直播或者听课, 可以使用 RtSimpleIMPL 来加入直播, 该路视频的内容包含直播视频、桌面共享、插播。

3 SDK 使用准备

3.1 开发环境要求

pc 操作系统: 64 位 windows 系统

jdk: 1.5 以上

android studio 或

eclipse: 3.5 以上 建议使用官方已经集成 adt 的 eclipse

android: 4.0 及以上

设备: 设备操作系统 4.0 以上, 需要访问硬件, 请不要使用模拟器进行开发

3.2 需要导入的 jar

[gensee_common.jar](#)

[gensee_hb.jar](#)

[gensee_rt_audio.jar](#)

[gensee_rtsdk.jar](#)

[gensee_doc.jar](#)

[gensee_expression.jar](#)

[gensee_video.jar](#)

如果有聊天, 请添加聊天资源 GSRtDemo-AndroidStudio 中 resourceLib 库, 并在聊天消息要显示或输入前初始化聊天资源: `ChatResource.initChatResource(Context context)`, 调用一次即可。

3.3 需要添加的动态库

libandroid_routine.so

libAVCDecoder.so

libH264Android.so

libstlport_shared.so

libRtRoutine.so

libspeex.so

libucbase.so

libucnet.so

libucpdu.so

libucpingpdu.so

libucwcc.so

libwebrtc_audio_preprocessing.so

libgensee-log.so

libucrtp.so(3.5 rtsdk 增加)

//文档需要, 如果没有文档 (ppt) 功能, 可以不用添加下面的库

libandroid-swfplayer.so

libSwfView.so

libucjpeg.so

libuctinyxml.so

将上述动态库文件添加到工程 `libs\ armeabi` 目录下, 如果没有该文件夹则新加 `libs\ armeabi` 文件夹。x86 同理。若有 `armeabi-v7a` 也从 `armeabi-v7a` 复制一份到里面。

3.4 配置与信息

3.4.1 权限(targetSdkVersion>22 时候请动态申请)

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera.autofocus" />
<uses-feature android:name="android.hardware.camera.flash" />
<uses-permission android:name="android.permission.RECORD_VIDEO" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.CHANGE_CONFIGURATION" />
```

3.4.2 version

```
<uses-sdk android:minSdkVersion="14" android:targetSdkVersion="19" />
```

3.4.3 OnTaskRet 辅助接口

下文中 OnTaskRet 辅助接口, 每一次调用执行结果通知, onTaskRet(**boolean** ret, **int** id, String desc)中 ret 为 true 代表执行成功, false 执行失败, id 和 desc 目前没有意义。

3.4.4 InitParam 说明

setDomain(String domain)	//设置站点域名 如 demo.gensee.com 不能为空
setNumber(String number)	//设置站点编号 “8 位” 数字字符串, 没有 liveld 时必填
setLoginAccount(String loginAccount)	//设置站点登录帐号(若后台设置需要登录, 必填)
setLoginPwd(String loginPwd)	//设置站点登录密码(若后台设置需要登录, 必填)
setNickName(String nickName)	//设置加入昵称 用于显示和统计(不要填空)
setJoinPwd(String joinPwd)	//设置加入口令 (和后台设置的对应角色口令保持一直, 没有设置口令一定要填写 “”)

setLiveld(String liveld)//设置直播 id, 等价于 setNumber, 如果设置了直播 id, 就不用 number
setUserId(long userId)//设置自定义用户 id, 使用的情况下请保证大于 1000000000, 否则无效
setUserData(String userData)//设置用户信息, 若使用, 请保证 key 和后台一致
setK(String k)//设置第三方认证 K 值, 如果启用认证, 请填写有效的 K, 没启用忽略

3.4.5 Userinfo 说明

getId()	//用户 id
getName	//用户昵称
getRole()	//用户角色 (复合角色)

```

IsHost()      //是否老师/组织者
IsPresentor() //是否主讲
IsPanelist()  //是否嘉宾/助教
IsAttendee()  //是否为客户端普通参与者/学生
IsAttendeeWeb() //是否为 web 端普通参与者/学生
IsHandup()    //是否举手

```

3.4.6 混淆打包事项

如果 app 打包需要混淆, 请添加混淆代码

```

-dontwarn com.gensee.**
-keep class com.gensee.**{*;}

```

3.5 快速接入 gensee 直播间 (课堂)

3.5.1 使用 RtSimpleImpl 加入直播间 (课堂)

Step1、创建好 RtSimpleImpl 实例。

这里要实现抽象函数, 并可重写其他函数来接收直播间的通知信息。onGetContext 必须返回有效的 app context。否则无声音。

```
RtSimpleImpl simpleImpl = new RtSimpleImpl() {...} //可参考 demo
```

设置视频显示的 view

```
simpleImpl.setVideoView(gsVideoView);
```

设置文档显示的 view

```
simpleImpl.setDocView(gsDocView);
```

Step2、获取直播的相关信息, 调用 RtComp 的 initWithGensee(initParam)进行初始化, 随后 RtComp.Callback 中 onInitied(String rtParam)响应并返回直播加入参数 rtParam。

Webcast 直播间信息和参数对照图:

```

String nickName = edtNick.getText().toString();
String joinPwd = edtJoinPwd.getText().toString();

RtComp action = new RtComp(MultiVideoActivity.this,

InitParam p = new InitParam();
// domain
p.setDomain(domain);
// 编号 (直播间号), 如果没有编号却有直播id的情况请
p.setNumber(number);
// number和liveId至少要一个
p.setLiveId("");
// 站点认证帐号, 根据情况可以填""
p.setLoginAccount(account);
// 站点认证密码, 根据情况可以填""
p.setLoginPwd(pwd);
// 昵称, 供显示用
p.setNickName(nickName);
// 加入口令, 没有则填""
p.setJoinPwd(joinPwd);
// 站点类型ServiceType.ST_CASTLINE-直播webcast.
// ServiceType.ST_MEETING 会议meeting.
// ServiceType.ST_TRAINING 培训 training
p.setServiceType(serviceType);

action.initWithGensee(p);
}

```

Training 课堂信息和参数对照图:

The diagram illustrates the mapping between the Java code on the left and the 'Training 直播' (Training Live Broadcast) form on the right. Red boxes highlight specific values in the form, and red arrows point from the code to these values.

Code Snippets:

```

@Override
public void onClick(View v) {
    String domain = edtDomain.getText().toString();
    String number = edtNum.getText().toString();
    String account = edtAccount.getText().toString();
    String pwd = edtPwd.getText().toString();
    String nickName = edtNick.getText().toString();
    String joinPwd = edtJoinPwd.getText().toString();

    InitParam p = new InitParam();
    //domain
    p.setDomain(domain);
    //编号 (直播间号), 如果没有编号却有直播id的情况请使用
    p.setNumber(number);
    p.setLiveId("b9743c6063934ec4ac2fb6bc83951d80");
    //站点认证帐号, 根据情况可以填""
    p.setLoginAccount(account);
    p.setLoginPwd(pwd);
    //昵称, 供显示用
    p.setNickName(nickName);
    //加入口令, 没有则填""
    p.setJoinPwd(joinPwd);
    //站点类型ServiceType.ST_CASTLINE 直播webcast,
    //ServiceType.ST_MEETING 会议meeting,
    //ServiceType.ST_TRAINING 培训
    p.setServiceType(serviceType);

    RtComp comp = new RtComp(getApplicationContext(),
        SimpleActivity.this);
    comp.initWithGensee(p);
}
    
```

Form Fields and Values:

- 课堂名称: android
- 开始时间: 2015-05-14 13:14:03
- 失效时间:
- 分享至社交媒体: 否
- 场景模板: 大讲堂
- 学生登录方式: 所有
- 加入URL: <http://192.168.1.142/training/site/r/01687827>
- 学生端加入URL: <http://192.168.1.142/training/site/s/01687827>
- 老师口令: 111111
- 助教口令: 222222
- 学生客户端口令: 333333
- 学生WEB端口令: 444444
- 所属课程: /
- 是否公开: 不公开

Annotations:

- 域名 (domain) points to the domain field.
- 直播编号 (live broadcast number) points to the number field.
- 站点类型 (service type) points to the service type field.
- 口令 (password) points to the password field.

Step3、使用 rtParam 和 simpleImpl 加入直播间（课堂）

```

public void onInit(String rtParam) {
    simpleImpl.joinWithParam("", rtParam);
}
    
```

正常加入后 sdk 会进行直播间状态、人员等信息的通知，可以根据需要重写 RtSimpleImpl 的相关函数来接收直播间的通知，RtSimpleImpl 实现了 IRoomCallback，因此可参考 IRoomCallback 说明来重写对应函数。

3.5.2 RtSdk 扩展更多功能

如果来实现聊天、问答、投票等更多功能，可获取 rtSdk 句柄进行扩展，下述回调设置代码尽可能在加入直播间(simpleImpl.joinWithParam())之前进行调用，以保证直播间通知的及时性、完整性。

```

RtSdk rtSdk = simpleImpl.getRtSdk(); //获取 rtSdk 实例

rtSdk.setLocalVideoView(localVideoView); //设置本地视频预览 view，观看端不需要

设置事件回调
rtSdk.setLodCallBack(lodCallback); //设置插播（媒体共享）模块回调
rtSdk.setChatCallBack(chatCallback); //设置聊天模块回调
rtSdk.setVoteCallBack(voteCallback); //设置投票模块回调
rtSdk.setQACallBack(qaCallback); //设置问答回调

聊天功能调用
rtSdk.chatWithPublic("公共聊天消息", null); //发送公共聊天消息
rtSdk.chatWithPerson("私聊消息", receiveUserId, null); //发送私聊消息

音频功能调用
rtSdk.audioOpenMic(null); //打开麦克风
rtSdk.audioCloseMic(null); //关闭麦克风
rtSdk.audioCloseSpeaker(null); //关闭扬声器
    
```



```
rtSdk.audioOpenSpeaker(null);           //打开扬声器
```

```
...
```

更多功能调用参考文档关于 **RtSdk** 的介绍。

4 回调（代理）响应

4.1 初始化响应回调 `RtComp.Callback`

本接口是 `RtComp` 中 `initWithGensee` 被调用后的结果响应。

4.1.1 `onInited(String joinParam)`

说明: `initWithGensee` 调用成功, `joinParam` 作为后续 `api` 的参数。

4.1.2 `onErr(int errCode)`

说明: `initWithGensee` 调用失败, 并将错误结果返回。`errCode` 即为错误码。其中错误码定义在 `Callback` 中。

使用: `RtComp.Callback.ERR_NUMBER_UNEXIST` //直播间不存在

错误码:

```
int ERR_JSON_ERROR = -1; //响应数据格式不正确, 请检查网络或参数
int ERR_DOMAIN = -100; // ip(domain)不正确
int ERR_TIME_OUT = -101; // 超时
int ERR_UNKNOWN = -102; // 未知错误
int ERR_SITE_UNUSED = -103; // 站点不可用
int ERR_UN_NET = -104; // 无网络
int ERR_DATA_TIMEOUT = -105; // 数据过期
int ERR_SERVICE = -106; // 服务不正确
int ERR_PARAM = -107; // 参数不正确
int ERR_THIRD_CERTIFICATION_AUTHORITY = -108; //第三方认证(用户权限校验)失败
int ERR_UN_CONNECTED = -109; //无法连接 请检查网络
int RESULT_SUCCESS = 1; // 成功

int ERR_NUMBER_UNEXIST = 0; // 直播间不存在
int ERR_TOKEN = 4; // 口令错误
int ERR_LOGIN = 5; // 用户名或密码错误
int ERR_FAIL_WEBCAST = 3; // 直播 id 不正确
int ERR_WEBCAST_UNSTART = 6; // 直播未开始//课堂过期
int ERR_ISONLY_WEB = 7; // 只支持 web
int ERR_ROOM_UNEABLE = 8; // 直播间不可用
int ERR_ROOM_OVERDUE = 11; // 过期
int ERR_AUTHORIZATION_NOT_ENOUGH = 12; // 授权不够
int ERR_UNTIMELY = 13; // 太早
```

4.2 直播间响应回调 `IRoomCallBack`

说明: 主要用作直播（课堂）相关状态的回调, 如加入状态、直播状态、录制状态、用户加入离开。

4.2.1 onInit(boolean result)直播间初始化响应

回调说明：直播（课堂）初始化，result 为 true 代表成功，false 代表失败。

返回 false 时请注意：

- 1、onGetContext 没有返回正确的 App Context;
- 2、动态库没有加载或本身不存在，请检查各架构对应的 so，也可以从日志里面查看缺少了什么。

4.2.2 onRoomJoin(int result, UserInfo self, boolean svrFailover)直播间加入响应

回调说明：加入直播（课堂）响应，result 为 0 代表加入成功，self 为当前用户的用户信息。svrFailover 是否是 server 重启恢复。

result 非 0 代表加入失败，参考 IRTEvent.IRoomEvent.JoinResult 中 join code。

```
int JR_OK = 0; //加入成功
int JR_ERROR = JR_OK + 1; //加入错误
int JR_ERROR_LOCKED = JR_OK + 2; //直播间（课堂）被锁定
int JR_ERROR_HOST = JR_OK + 3; //组织者（老师）已经加入
int JR_ERROR_LICENSE = JR_OK + 4; //当前课堂人数已满，无法加入
int JR_ERROR_CODEC = JR_OK + 5; //音频编码不匹配，请取消 AAC 配置
int JR_ERROR_TIMESUP = JR_OK + 6; //超时
int JR_ERROR_IP_DENY = JR_OK + 7; //IP 被封禁
int JR_ERROR_TOO_EARLY = JR_OK + 8; //加入过早。受后台（直播前加入）配置影响
int JR_ERROR_GETPARAM = JR_OK - 1; //参数获取失败（页面启动）
int JR_ERROR_LIB_DISABLE = JR_OK - 2; //动态库(so)缺失或加载失败
```

4.2.3 onRoomLeave(int reason)直播间退出响应

回调说明：当前用户退出直播（课堂）。reason 为原因如被踢出、主动退出等。

其中 reason 取值定义在 IRTEvent.IRoomEvent.LeaveReason 中：

```
int LR_NORMAL = 0; //用户自行退出
int LR_EJECTED = LR_NORMAL + 1; //被踢出
int LR_TIMESUP = LR_NORMAL + 2; //时间
int LR_CLOSED = LR_NORMAL + 3; //直播（课堂）已经结束（被组织者结束）
int LR_EJECTED_IP_DENY = LR_NORMAL + 4; //因 ip 被封禁而被踢出
int LR_DUPLICATE = LR_NORMAL + 5; //被挤出房间，相同 id 的用户在其他设备加入
```

一旦该函数被调用的时候可以进行数据清理和相关的对象释放，RtSdk 中 release。

4.2.4 onRoomReconnecting()直播间重连通知

回调说明：正在重连，网络条件不好或网络断开后会被调用。重新连上等同于一次加入直播，因此会重新通知加入结果 onRoomJoin。

4.2.5 onRoomLock(boolean locked)直播间锁定通知

回调说明：直播锁定状态。true 为锁定，false 未锁定

4.2.6 onRoomUserJoin(UserInfo userInfo)直播间新用户加入通知

回调说明：有新用户加入。

4.2.7 onRoomUserUpdate(UserInfo userInfo)直播间用户信息变化通知

回调说明：用户状态更新。

4.2.8 onRoomUserLeave(UserInfo userInfo)直播间用户退出通知

回调说明：用户退出。

4.2.9 onRoomPublish(State s)直播状态通知

回调说明：直播状态变化。

状态定义在 State 中：

S_NONE //默认 直播未开始

S_RUNNING //直播中

S_STOPPED //直播停止

S_PAUSED //直播暂停

4.2.10 onRoomRecord(State s)录制状态通知

回调说明：录制状态变化。

4.2.11 onRoomData(String key, long value)直播间数据通知

回调说明：附加状态通知，例如课堂中上讲台、上提问席等，根据具体情况处理。

4.2.12 onRoomBroadcastMsg(String msg)直播间广播消息

回调说明：广播消息

4.2.13 onRoomRollcall(int timeout)直播间点名

回调说明：点名开始，timeout 为点名有效时间。

4.2.14 onRoomRollcallAck(long userId)用户应答点名后组织者收到的通知

回调说明：userId 已经对本次点名做出了应答，由组织者客户端处理这个回调。

4.2.15 onRoomHandup(long userId, String data)直播间用户举手通知

回调说明：userId 对应的用户举手了。

4.2.16 onRoomHanddown(long userId)直播间用户取消举手的通知

回调说明：userId 对应的用户取消了举手。

4.2.17 getServiceType()返回站点类型

回调说明：返回 service 类型如 castline、training、meeting。定义在 ServiceType 中。

4.2.18 onGetContext()返回 app context

回调说明: android 客户端返回 app context。

4.2.19 onChatMode(int chatMode)聊天模式切换通知

回调说明: 直播间内聊天模式通知, 即聊天权限变化通知。默认为 CHAT_MODE_ALLOW_ALL

chatMode 取值定义在 IChatCallBack 当中:

```
int CHAT_MODE_DENY = 0; // 禁止聊天, 禁止后要控制当前用户不能发送聊天消息  
int CHAT_MODE_ALLOW_ALL = 1; // 允许任何形式的聊天  
int CHAT_MODE_ALLOW_PUBLIC = 2; // 允许进行公共聊天  
int CHAT_MODE_ALLOW_PRIVATE = 3; // 允许进行私人聊天
```

4.2.20 onLottery(byte step, String userlist)抽奖流程通知

回调说明: 抽奖进行的步骤通知, 其中 step 取值定义在 IRoomCallback 中:

```
byte LOTTERY_BEGIN = 1; // 抽奖开始  
byte LOTTERY_END = 2; // 抽奖结束, userlist 为被抽中人员名称  
byte LOTTERY_CANCEL = 3; // 抽奖取消
```

4.2.21 onSettingSet(String key, int val)直播间 int 类配置信息

回调说明: sdk 将直播间配置以 key-value 的对应关系告知 app 应用层, 原则上应用层需要存储一份, 因为后续操作当中可能再次向应用层要回配置信息, 这个函数主要针对 int 类的配置, 例如视频分辨率和帧率。Key 定义在 RTSharedPref 里面:

```
String KEY_VIDEO_W = "save.video.width";           //视频采集宽，默认 val 320
String KEY_VIDEO_H = "save.video.height";          //视频采集高，默认 val 240
String KEY_VIDEO_FPS = "save.video.fps";           //视频采集帧率，默认 val 15，不要随意更改
String KEY_CHAT_DISABLE_ATTENDEE_PRIVATE = "chat.disable.attendee.private";
//0:enabled [default]; 1: disabled 请将这个聊天配置和 onChatMode 的聊天权限合并使用（都可以私聊的条件下才可发送私聊消息）

val 为对应的值。
```

4.2.22 onSettingSet(String key, String val)直播间 String 类配置信息

回调说明: 同 onSettingSet(String,int),sdk 将直播间配置以 key-value 的对应关系告知 app 应用层, 这个函数主要针对 String 类的配置。Key 定义在 RTSharedPref 里面:

```
String KEY_CLASS_NAME = "training.class.name";//直播间名称，取代原有的 onRoomSubject。
```

val 为对应的值。

4.2.23 int onSettingQuery(String key, int val)直播间取回 int 类配置信息

回调说明: 主要取回 int 类配置信息, 一般学生端用不上, 请按需返回, 默认请返回 0, 例如取回分辨率宽 (仅举例):

```
int onSettingQuery(String key, int val){
    if(KEY_VIDEO_W.equals(key)){
        return 320;
    }
    //默认返回 0
    return 0;
}
```

4.2.24 String onSettingQuery(String key)直播间取回 String 类配置信息

回调说明: 主要取回 String 类配置信息, 一般学生端用不上, 请按需返回, 默认请返回 null, 例如返回直播间名称 (仅举例):

```
String onSettingQuery(String key){
    if(KEY_CLASS_NAME.equals(key)){
        return 已经存储的名称;
    }
    //默认返回 null
    return null;
}
```

4.2.25 onNetworkReport(byte level)网络质量通知

回调说明:网络质量通知, level 取值 0-100 (差-好)。

4.2.26 onNetworkBandwidth(int bpsSend, int bpsRecv) 带宽通知

回调说明: 上下行带宽通知 bpsSend 发送带宽, bpsRecv 接收带宽

4.2.27 onLiveInfo(LiveInfo info)直播简介等信息通知

这个回到返回直播的简介、主讲人介绍、直播议程三个信息。需要加入 initWithParam 之后再调用 getLiveInfo 才会回调。

```
String scheduleInfo = info.getScheduleInfo(); //直播议程、直播安排、课堂安排
```

```
String speakerInfo = info.getSpeakerInfo();//主讲信息、主讲介绍、老师介绍
```

String description = info.getDescription();//直播简介、直播信息、课堂信息

4.2.28 onRoomWebLayoutChange(int weblayout)web 播放器布局切换通知

该回调返回的 web 播放器的布局方式，默认是 0。当组织者操作切换布局控制的时候，会回调相应的模式，**效果**请开发者以及产品自行操作组织者端并对照 pc web 了解。

weblayout 取值：

0 – 文档为主（ppt word pdf 等）

1 – 视频最大化

2 – 文档最大化

3 – 视频为主

4.3 音频回调 IAudioCallBack

本接口是音频相关回调，基本上不需要处理，一般可能会处理的是电频值。

4.3.1 onAudioJoinConfirm(boolean ok)音频模块加载通知

回调说明：音频模块加载成功。

4.3.2 onAudioMicAvailable(boolean isAvailable)mic 设备“可用”通知

回调说明：音频 mic 可用度。isAvailable 为 true 可用。

4.3.3 onAudioMicOpened()mic 被打开通知

回调说明：mic 已开启。

4.3.4 onAudioMicClosed()mic 被关闭通知

回调说明：mic 已关闭。

4.3.5 onAudioSpeakerOpened()扬声器被打开通知

回调说明：扬声器已打开。

4.3.6 onAudioSpeakerClosed()扬声器被关闭通知

回调说明：扬声器已关闭。

4.3.7 onGetContext()返回 appContext

回调说明：android 客户端获取 app 上下文。

4.3.8 onAudioLevel(int level, long userId)用户扬声器电频通知

回调说明：userId 对应用户的音频电频值，注：电频值不是音量大小，大小在 0-100 之间。

4.4 视频回调 IVideoCallBack

本接口是视频相关状态回调，以及用户视频数据接收。

4.4.1 onVideoJoinConfirm(boolean ok)视频模块加载通知

回调说明：ok 标识视频模块加载是否成功，true 代表成功，false 代表失败。

4.4.2 onVideoCameraAvailable(boolean ok)摄像头可用通知

回调说明：设备摄像机可用状态，ok 为 true 表示可用，false 不可用。

4.4.3 onVideoCameraOpened()摄像头被打开通知

回调说明：摄像机已经打开。

4.4.4 onVideoCameraClosed()摄像头被关闭通知

回调说明：摄像机已经关闭。

4.4.5 onVideoJoin(UserInfo user)用户视频开启通知

回调说明：用户视频开启通知，如果需要显示某用户的视频，可以调用 rtsdk 的 displayvideo 来订阅该用户的视频，同时也可以取消订阅，从而达到数据流量的节省。

注：userId == UserInfo.LOD_USER_ID 代表插播，其他是具体的某个用户

4.4.6 onVideoLeave(long userId)用户视频关闭通知

回调说明：用户视频关闭，此时可以取消该用户的视频。

4.4.7 onVideoActivated(UserInfo user, boolean bActivated)直播视频启用通知

回调说明：bActivated 为 true 代表直播视频开启，false 代表直播视频取消，本质上是某个用户的视频，需要显示该用户视频的话，则要订阅该用户的视频数据。

4.4.8 onVideoDisplay(UserInfo user)观看 user 视频响应

回调说明：观看（displayVideo）user 用户视频响应。

4.4.9 onVideoUndisplay(long userId)取消观看 userId 视频响应

回调说明：停止观看（unDisplayVideo）userId 用户视频的响应。

4.4.10 onVideoDataRender(long userId, int width, int height, int frameFormat, float displayRatio, byte[] data)视频数据流通知

回调说明：userId 对应的用户视频数据，其中该数据传递给 GSVideoView 就能显示其视频，请将 GSVideoView 和用户对应起来，不要交错的把不同用户数据传递给 GSVideoView。

4.5 桌面共享回调 IAsCallBack

本接口是做桌面共享回调使用的，通知共享状态及其数据。

4.5.1 onAsJoinConfirm(boolean ok)桌面共享模块加载通知

回调说明：桌面共享加载状态，ok 为 true 代表加载成功，false 为失败。

4.5.2 onAsBegin(long owner)桌面共享开启通知

回调说明：桌面共享开始，owner 启用桌面共享的 userId。

4.5.3 onAsEnd()桌面共享结束通知

回调说明：桌面共享结束。

4.5.4 onAsData(byte[] data, int width, int height)桌面共享数据流通知

回调说明：桌面共享数据，使用 GSVideoView 来显示，将 data 传递给 GSVideoView。

4.6 插播（媒体共享）回调 ILodCallBack

本接口是用作插播播放状态通知使用的，包括播放开始、结束、进度等状态。

4.6.1 onLodFailed(String id)插播（媒体共享）失败通知

回调说明：播放失败。

4.6.2 onLodStart(LiveodItem liveodItem) 插播（媒体共享）开始通知

回调说明：插播（liveodItem）播放开始。

4.6.3 onLodSkip(LiveodItem liveodItem) 插播（媒体共享）跳跃通知

回调说明：插播进度变化。

4.6.4 onLodPause(LiveodItem liveodItem); 插播（媒体共享）暂停通知

回调说明：插播暂停。

4.6.5 onLodStop(LiveodItem liveodItem); 插播（媒体共享）停止通知

回调说明：插播停止。

4.6.6 onLodPlaying(LiveodItem liveodItem); 插播（媒体共享）播放中

回调说明：插播正在播放，进度变更。

4.6.7 onLodAdd(LiveodItem liveodItem); 插播（媒体共享）被添加通知

回调说明：添加了一个插播。一般是由 pc 客户端主动添加。

4.6.8 onLodRemove(String id); 插播（媒体共享）被移除通知

回调说明：插播被删除，id 对应于 liveodItem 的 url。一般由 pc 客户端主动删除。

4.7 聊天回调 IChatCallBack

本接口为接收聊天消息，包括私聊、公聊消息。注：聊天表情显示需要添加聊天资源 RTRoutineLib 库，并在聊天消息要显示或输入前初始化聊天资源：ChatResource.initChatResource(Context context)，调用一次即可。表情处理：

- 1、优先初始化聊天资源 ChatResource.initChatResource(Context context)
- 2、用 MyTextViewEx 显示一条聊天消息的富文本。MyTextViewEx.setRichText(richText);
- 3、表情选择面板：
 - (1)SpanResource.getBrowMap 取到所有的表情 Map<key,value>, 即 Map<String, Drawable>, 一个表情对应一个 key-value, 取出(map.entrySet())所有的 value 就是所有的表情 Drawable, 绑定到界面上。
 - (2)界面上选择一个表情, 得到对应的 key, 将 key 传递到 ChatEditText 输入框即可。
- 4、编辑发送聊天消息用 ChatEditText 接收上面 3-(2)中的 key: ChatEditText.insertAvatar(key,0)
- 5、发送聊天消息, 取出输入框 ChatEditText 的纯文本(getChatText())和富文本(getRichText())作为公聊或私聊的参数进行发送。发送接口请在本文中查找公聊和私聊详细说明。

增加回调函数 `onChatMessage(ChatMsg msg)`：公聊、私聊消息的回调，取代原有的 `onChatWithPublic` 和 `onChatWithPersion`。

4.7.1 `onChatJoinConfirm(boolean bRet)`聊天模块加载通知

回调说明：聊天模块加载状态，`bRet` 为 `true` 代表成功，`false` 代表失败。

4.7.2 `onChatMessage(ChatMsg msg)`聊天消息回调

公聊、私聊消息的回调：（注意聊天时间都是收到消息时的系统时间）

```
int msgType = msg.getChatMsgType();//消息类型 公聊、私聊
```

```
String id = msg.getId();//消息 id
```

```
String senderName = msg.getSender();//发送者昵称
```

```
String text = msg.getContent();//聊天纯文本
```

```
String richText = msg.getRichText();//聊天富文本
```

```
long senderId = msg.getSenderId();//发送者 id
```

```
int role = msg.getSenderRole();//发送者角色
```

```
switch (msgType) { //消息类型 公共聊天/私聊
```

```
    case ChatMsg.CHAT_MSG_TYPE_PUBLIC: //公共聊天
```

//老用户更新处理：调用原有代码的 `onChatWithPublic` 去掉 “@verride” 再加上一个 `role`，没有用到 `role` 忽略 `role`

```
        onChatWithPublic(senderId, senderName, text, richText, id);
```

```
        break;
```

```
    case ChatMsg.CHAT_MSG_TYPE_PRIVATE: //私聊
```

//老用户更新处理：调用原有代码的 `onChatWithPersion` 去掉 “@verride” 再加上一个 `role`，没有用到 `role` 忽略 `role`

```
        onChatWithPersion(senderId, text, richText, id);
```

```
        break;
```

```
    default:
```

```
        break;
```

```
}
```

去掉 `onChatToPersion`，即发出的私聊消息不再回调，修改

```
String msgId = UUID.randomUUID().toString();//消息 id,即加上 uuid
```

```
ChatMsg msg = new ChatMsg(text, rich, ChatMsg.CHAT_MSG_TYPE_PRIVATE, msgId);
```

```
msg.setReceiverId(userId);//接收人的 id
```

//私聊消息 `msgId`，`text`，`rich` 都不为空且接收人 `id` 正确才有效

```

mRtSdk.chatWithPersion(msg,new OnTaskRet() {

    @Override

    public void onTaskRet(boolean ret, int arg1, String arg2) {

        if(ret){

            //自己发送的私聊消息添加到列表，如果有其他处理方式，忽略。

            onChatToPersion(userId, text, rich,msgId);

        }

    }

});

```

附老回调接口，仅供老用户更新参考：

`onChatWithPersion(UserInfo userInfo, String msg, String richText)`收到私聊消息

回调说明：收到 `userInfo` 发来的私人聊天消息，`msg` 为文本消息，`richText` 为对应的富文本消息。

`onChatWithPublic(UserInfo userInfo, String msg, String richText)`收到公聊消息

回调说明：收到 `userInfo` 发来的公共聊天消息，`msg` 为文本消息，`richText` 为对应的富文本消息

`onChatToPersion(long userId, String msg, String richText)`发出私聊消息回送

回调说明：自己发送的私聊发送回传，`msg` 为文本消息，`richText` 为对应的富文本消息

4.7.3 onChatEnable(boolean enable)聊天禁言通知

回调说明：聊天权限被控制通知，`enable` 为 `true` 代表可以进行聊天，`false` 不能聊天，客户端需要由此来控制聊天功能。本通知要结合 `IRoomCallback` 中 `onChatMode` “与”的关系使用，两处都要同时满足聊天才能发送消息。

4.7.4 onChatCensor(long userId, String msgId)聊天审核删除通知

回调说明：当组织者或有权限的人发出删除该条消息或某一人的所有消息时候回调，接收方应对应的去删除聊天消息，已达到审核的目的。`userId` 不等于 0 的时候代表删除 `userId` 用户的所有聊天消息。当 `userId` 等于 0 而 `msgId` 不为空的时候，对应删除消息 `id` 等于 `msgId` 的聊天消息。

4.8 投票、试卷、答题卡回调 IVoteCallBack

本接口为投票调查（试卷）接收使用的，需要接收则需要注册该回调，`rtSdk` 中 `setVoteCallBack`。

投票调查的数据结构：

`VoteGroup` 投票调查结构：

```

getM_strId() //投票调查的 id

getM_strText() //投票调查的主题

getM_questions() //投票调查问题列表(List<VoteQuestion>)

isM_bForce() //是否为强制投票

isM_bPublish()//是否已经发布

```

isM_bDeadline()//是否已经终止答题

isM_bPublishResult()//是否已经发布最终结果（公布）

VoteQuestion 投票调查问题结构:

getM_strId() //问题 id

getM_strText()//问题的内容（题目）

getM_strType()//问题类型（"multi"//多选 "single"//单选 "text"//问答题，VoteQuestionType 中）

```
public static class VoteQuestionType {  
    public static final String MULTI_TYPE = "multi";//多选  
    public static final String SINGLE_TYPE = "single";//单选  
    public static final String TEXT_TYPE = "text";//问答题  
}
```

getM_nScore()//问题的得分（如 10 分）

getM_nResultUser()//参与该问题的人数

getM_strAnswer()//如果是问答题，该函数返回文字答案

setM_strAnswer(String answer)//如果是问答题，用户填写的答案通过该函数设置

getM_answers()//如果是单选题或多选题，那么选项从这里返回（List<VoteAnswer>）

VoteAnswer 问题选项结构:

getM_strId()//选项 id

getM_strText()//选项内容（选项答案）

getM_nResultUser()//发布结果时候，选择该选项的人数

isM_bCorrect()//是否是正确答案

setM_bChoose(boolean m_bChoose)//根据界面上用户是否选择该选项来设置 true/false

isM_bChoose()//返回 setM_bChoose 设置的值

对于投票调查：不需要重新构造一个 VoteGroup，只要将响应回来的 VoteGroup 对应的绑定到界面上，随着界面的操作调用 setM_strAnswer(String answer)或 setM_bChoose(boolean m_bChoose)之后，就可以将这个 VoteGroup 提交(voteSubmit(VoteGroup voteGroup, OnTaskRet taskRet))。至于其他的交互根据响应来即可，例如强制投票是一定需要参与的，不能关闭，发布结果的时候，就显示最终结果。

4.8.1 onVoteJoinConfirm(boolean bRet)投票模块加载通知

回调说明：投票模块加载状态，bRet 为 true 代表成功，反之失败。

4.8.2 onVoteAdd(VoteGroup voteGroup)投票被添加通知

回调说明：新添加了一个投票调查 voteGroup。

4.8.3 onVoteDel(VoteGroup voteGroup)投票被删除通知

回调说明：一个投票调查 voteGroup 被删除了。

4.8.4 onVotePublish(VoteGroup voteGroup)投票发布通知

回调说明：一个投票调查 voteGroup 被发布了（开始答题）。

4.8.5 onVotePublishResult(VoteGroup voteGroup)投票发布结果通知

回调说明：一个投票调查结果发布了（答题结束，公布结果）。

4.8.6 onVoteSubmit(VoteGroup voteGroup)投票被回复通知

回调说明：某个用户提交了投票（提交了答案）。

4.8.7 onVoteDeadline(VoteGroup voteGroup)投票结束通知

回调说明：投票 voteGroup 终止，停止答题。

4.8.8 onVotePostUrl(String strURL, long optUserId)第三方投票通知

回调说明：第三方投票，strURL 是一个第三方投票的链接地址，调用浏览器打开；optUserId 操作者用户 id。

4.8.9 onCardPublish(Card card)答题卡发布通知

回调说明：小班课老师发答题卡时候回调，一般是学生端处理，显示答题卡进行选择答题

Card:

//答题卡类型，单选：Card.CARD_SINGLE_OPTION,多选：Card.CARD_MULTI_OPTION

int type = card.getType();

int timeout = card.getTimeout();//答题时间

int[] itemIds = card.getItemIds();//选项 id 的集合数组

String[] itemTexts = card.getItemTexts();//选项的文本表示集合数组，对应 itemIds，如 A,B,C,D。

4.8.10 onCardSubmit(long userId, int[] itemids)答题卡答题通知

回调说明：有人提交答题卡通知，一般是老师端处理，用于统计答题情况。

userId:答题人用户 id

itemids:答题人选择的选项 id 集合

4.8.11 onCardEnd()答题结束通知

回调说明：小班课中老师终止答题时学生端响应的事件，学生端应该终止答题。

4.8.12 onCardResultPublish(int type, int totalSubmitted, int[] itemSubmitted, int[] itemOptions) 答题卡结果通知

回调说明：小班课老师发布最后的统计结果。

type:答题卡类型，多选和单选

totalSubmitted: 参与了答题的人数，也就是提交答题卡的人数。

itemSubmitted: 选项所对应的选择人数

itemOptions: 所有选项的正确性，数组中的值是 0 错误或 1 正确。

itemSubmitted 和 itemOptions 数组长度和 card 中的选项数目是完全一样的。且是对应关系，

如 itemOptions[0] == 1,代表 card 中第一个选项是正确选项，0 为错误选项，以此类推；

itemSubmitted[0] 代表选择了第一个选项的人数，itemSubmitted[1]为选择了第二个选项的人数，以此类推。

4.9 问答回调 IQACallback

本接口是作为问答通信使用的，使用问答功能则要进行回调注册，通过 `rtSdk` 中 `setQACallback(IQACallback qaCallback)`来注册。

4.9.1 onQaJoinConfirm(int isOk) 问答模块加载通知

问答模块加载结果，`isOk` 为 `true` 代表加载成功，`false` 加载失败。

4.9.2 onQaSettingUpdate(boolean isAutoDispatchQuestion,boolean isAutoPublishQuestion, boolean isQAEnable) 问答权限通知

问答权限通知，当组织者调用 `qaSetState` 的时候，其他客户端的 `onQaSettingUpdate` 被调用。App 要处理好这个权限，`isQAEnable` 为 `false` 的时候禁止用户提问。

4.9.3 onQaQuestion(QaQuestion qaQuestion, int state) 问答通知

问答通知包括新问题、新回复、回复状态、发布、推送的通知，其中通知类型使用 `state` 来区分，`qaQuestion` 为当前的问题及答案：

```
String questId = qaQuestion.getStrQuestionId();// 问题 id
String question = qaQuestion.getStrQuestionContent();// 问题内容
String requester = qaQuestion.getStrQuestionOwnerName();// 提问者名字
String requesterFixer = qaQuestion.getStrTaggedUserName();// 问题处理人名字
long qTime = qaQuestion.getDwQuestionTime();// 提问时间（秒）
List<QaAnswer> answers = qaQuestion.getQaAnswerList();// 答案列表
if (answers != null) {
    for (QaAnswer answer : answers) {
        if (answer != null) {
            String answerId = answer.getStrAnswerId();// 答案 id
            String answerContent = answer.getStrAnswerContent();// 答案内容
            String answerOwner = answer.getStrAnswerOwnerName();// 答案回复者名字
            long answerOwnerId = answer.getLIAnswerOwnerId();// 答案回复者 id
            long anwerTime = answer.getDwAnswerTime();// 回复时间（秒）
        }
    }
}
```

`state` 定义在 `IQACallback` 中，取值如下：

`int QA_STATE_QUESTION = 0;//新问题`

`int QA_STATE_ANSWER = QA_STATE_QUESTION + 1;//新回复`

```

int QA_STATE_PUBLISH = QA_STATE_QUESTION + 2; //发布
int QA_STATE_CANCEL_PUBLISH = QA_STATE_QUESTION + 3; //取消发布
int QA_STATE_PUSH = QA_STATE_QUESTION + 4; //推送
int QA_STATE_ASSIGN = QA_STATE_QUESTION + 5; //指派
int QA_STATE_VOICE_REPLY = QA_STATE_QUESTION + 6; //语音回复
int QA_STATE_VOICE_REPLYING = QA_STATE_QUESTION + 7; //正在语音回复
int QA_STATE_TEXT_REPLYING = QA_STATE_QUESTION + 8; //正在文字回复

```

4.10 GSDocViewGx 事件回调接口 OnDocViewEventListener

OnDocViewEventListener 主要是为了 app 能够根据 GSDocViewGx 的事件来处理一些事情，例如全屏切换、隐藏显示其他 ui 元素。因此，app 需要实现这个回调并将其句柄设置给 GSDocViewGx。注：这个回调只有在 GSDocViewGx 处于观看模式的时候才会被调用，编辑模式是无效的。

4.10.1 onDoubleClicked(IGSDocView docView)

回调说明：双击事件。app 处理了这个事件之后返回 true，表示该事件已经被处理。

4.10.2 onSingleClicked(IGSDocView docView)

回调说明：点击事件。app 处理了这个事件之后返回 true，表示该事件已经被处理。

4.10.3 onEndHDirection(IGSDocView docView,int type,int eventType)

回调说明：GSDocViewGx 水平方向拖到边界。

type 匹配：

GSDocView.DRAG_END_LEFT = 1; //左边界

GSDocView.DRAG_END_RIGHT = 2; //右边界

eventType:手势 拖拽或放开

4.11 文档事件回调 IDocCallBack

IDocCallBack 是文档模块的相关事件回调，包括文档打开、关闭、翻页、动画等。

4.11.1 onDocJoinConfirm 模块加载通知

文档模块加载完成

4.11.2 onDocOpened(PduDoc doc)文档打开

一个文档打开，所有人会响应这个函数，代码打开了文档，但打开仅仅是文档信息，不（training 小班）一定代表翻页动作，因此这种情况下其他人或自己不会显示的，具体看 pc 客户端的表现。

doc 为打开的文档，可能是 ppt、pdf、word 等支持的类型的一种。

4.11.3 onDocClosed(int docId)文档关闭通知

id 为 docId 的文档被关闭，有一个关闭了，所有人都会响应。

4.11.4 onDocGotoPage(int docId, int pageId, int aniStep)翻页

翻页响应，一般情况下收到翻页动作才会显示翻到的这一页，所有人都会响应。

docId 文档 id。

pageId 文档中被翻页的页的 id。

aniStep 动画步数，ppt 类型的文档有效。

4.11.5 onDocPageReady(int docId, PduPage page, byte[] data)文档某页数据响应

当文档打开后，会随着翻页或翻页之前把对应的文档实际内容返回，有了返回后才能显示。

onDocReceiveAnno(int docId, int pageId, AbsAnno anno)收到标注响应

老师或组织者做标注时，其他人收到的标注。这部分是自动渲染到 GSDocViewGx 上的，不用理会

docId 文档 id。

page 所属页，

anno 具体标注内容。

4.11.6 onDocRemoveAnno(int docId, int pageId, long annold)移除标注

组织者或老师删除标注的时候响应，实际上也同时会在 GSDocViewGx 上取消渲染，不用理会

docId 文档 id。

pageId 所属页 id，

annold 具体标注的 id

4.11.7 onDocPageSize(int width, int height)当前页的分辨率

当翻页的时候，会响应当前页的分辨率，需要的情况可以使用

width 宽

height 高

4.11.8 onDocUploadStatus(int docId, int status)文档(jpeg 图片上传结果通知)

如果有打开调用 openDoc 来增加文档的情况下，可以根据这个响应来更新上传情况。

docId 上传的文档 id

status 上传结果：定义在 IDocCallback 中

int DOC_UPLOAD_BEGIN = 1;//上传开始

int DOC_UPLOAD_END = 3;//上传完成

int DOC_UPLOAD_GOING = 2;//正在上传

int DOC_UPLOAD_FAILURE = 0;//上传失败

4.12 点赞同勋章事件回调 IMedalPraiseCallback

勋章：一般是老师发给学生的，点赞：一般是学生给老师点赞或发给老师的点赞。在小班课堂中，点赞是学生给讲台上的用户（可以是老师、助教和学生）；勋章是老师发给学生的。实际使用可根据自己的业务来确定点赞、勋章的发送逻辑控制。点赞和勋章类型通过如下定义来区分(如果有 windows 或 mac 用户参与，请严格按照小班课逻辑实现)：

IMedalPraiseCallback.TYPE_FAOUR//点赞

IMedalPraiseCallback.TYPE_MEDAL//勋章

请相应在网站后台站点设置分别配置点赞和勋章功能为启用：



PraiseUserInfo info:点赞或勋章信息

info.getUserId()//对应用户的 userId

info.getRemain()//剩余量

info.getSend()//发送量

info.getRecv()//接收量

4.12.1 onPraiseInfo(int result,String praiseType,PraiseUserInfo info)点赞或勋章信息通知

当用户进行点赞或勋章功能的初始化后，会响应这个函数，此时可以得知当前用户的勋章发送数、接收数、剩余数；点赞发送数、接收数、剩余数。

result 结果码，0 代表 OK，后面的参数有效，其他值代表失败，后面的参数无效。

praiseType 取值：

IMedalPraiseCallback.TYPE_FAOUR//点赞

IMedalPraiseCallback.TYPE_MEDAL//勋章

info:点赞或勋章信息

4.12.2 onPraiseRecvList(int result,String praiseType,PraiseInfo[] praises)勋章排行

一般是获取勋章排行时候响应，praises 从高到低排列。

result 结果码，0 代表 OK，后面的参数有效，其他值代表失败，后面的参数无效。

praiseType 取值：

IMedalPraiseCallback.TYPE_FAOUR//点赞

IMedalPraiseCallback.TYPE_MEDAL//勋章

praises:排行榜数组。

PraiseInfo info: 排行榜单元信息

info.getUserId()//对应用户的 userId

info.getUserName()//对应用户名

info.getTotal()//获得勋章或点赞的总数量

4.12.3 onPraiseNotify(String praiseType, int time,PraiseUserInfo senderInfo,PraiseUserInfo receiverInfo,String comment)勋章或点赞变更通知

在勋章和点赞发送后，所有 rtsdk 用户都会响应该函数，也就是说收到该通知，一般有两个变化，发送者和接收者。

praiseType 取值：

IMedalPraiseCallback.TYPE_FAOUR//点赞

IMedalPraiseCallback.TYPE_MEDAL//勋章

time: 发送时间, 单位秒。

senderInfo 发送者的最新勋章或点赞信息, ui 更新对应 userId 用户的信息。

receiverInfo 接收者的最新点赞或勋章信息, ui 更新对应 useId 用户的信息。

comment 暂时无用

5 RTSDK API

5.1 RtComp

说明：首先要 domain、number 等参数获取 RtSdk 的加入参数。先调用 initWithGensee，响应为 Callback 的 onInited。其参数就是 RtSdk 需要使用的参数。

5.1.1 initWithGensee(InitParam p) 直播参数初始化

调用说明：InitParam 设置如下：

```
InitParam p = new InitParam();  
//domain, 必填项, 如: demo.gensee.com  
p.setDomain("");  
//编号 (直播间号) 必填项, 如: 93857580, 与 liveld 至少填一个  
p.setNumber("");  
//直播(课堂)id, 必填项, 与编号至少填一个  
p.setLiveld("");  
//站点认证帐号, 根据情况可以填""  
p.setLoginAccount("");  
//站点认证密码, 根据情况可以填""  
p.setLoginPwd("");  
//昵称, 供显示用  
p.setNickName("");  
//加入口令, 没有则填""  
p.setJoinPwd("");  
//站点类型 ServiceType.WEBCAST 直播 webcast 站点类型,  
//      ServiceType.TRAINING 培训 training 站点类型  
p.setServiceType(ServiceType.TRAINING);
```

相关的参数可以参照下图:

直播



demo.gensee.com/webcast/	
基本信息	
嘉宾	
文档	
投票调查	
主 题	RtSDK TEST
编 号	37916496
开始时间	2014-12-11 16:33
结束时间	
持续时间	
是否公开	不公开
是否要求登录	否
Web端是否允许切换	否
加入 URL (web)	http://demo.gensee.com/webcast/sit
加入 URL (Client)	http://demo.gensee.com/webcast/sit
组织者口令	111111
嘉宾口令	222222
普通参加者口令	333333

Traning:

实时课堂信息	
课堂名称:	mobile test
开始时间:	2014-12-17 17:01:02
失效时间:	
场景模板:	小班课
学生登录方式:	所有
加入URL:	http://demo.gensee.com/training/site/r/42655460
学生端加入URL:	http://demo.gensee.com/training/site/s/42655460
老师口令:	111111
助教口令:	222222
学生客户端口令:	333333
学生WEB端口令:	444444

调用响应: Callbak, 即 4.1.1 或 4.1.2。在得到 rtSdk 的加入参数即可以进入加入直播的流程。

5.2 RtSdk

RtSdk 包含了直播中的基本操作, 加入、退出以及相关控制操作。聊天、音频、视频、投票等, 可以按需要进行回调的设置。

5.2.1 initWithParam(String logFileName, String joinParam,IRoomCallBack roomCallBack) 直播间初始化

RtSdk 初始化, logFileName 可以传入一个有效的文件路径或者传 "", joinParam 即为 4.1.1 响应的结果 "joinParam", roomCallBack 是接收直播 (课堂) 流程状态的必需参数, 由 app 自行实现。调用结果由 roomCallBack 的 onInit (4.2.1) 返回, app 处理其结果, 若结果是调用成功, 可以继续调用 join (5.2.2)。

5.2.2 initWithLaunchCode(String logFileName, String launchCode,IRoomCallBack roomCallBack) 直播间初始化

RtSdk 初始化, 使用于网页启动时候调用。logFileName logFileName 可以传入一个有效的文件路径或者传 "", launchCode 为网页传过来的一个字符串。样例:

```
6170693d687474703a2f2f64656d6f2e67656e7365652e636f6d2f636c69656e746170692f6170696
368616e6e656c0a65767449443d363433343764353132326538343663636233616435373537393
23962623162320a7573657249443d343038363131320a7369746549443d330a757064617465526
f6f743d0a70726f647563744e616d653d436173744c696e650a70726f647563744c616e673d7a685f
434e3b0a
```

响应流程同 5.2.1。

5.2.3 join(OnTaskRet taskRet) 加入直播间

加入直播 (课堂)。一般在 rtSdk 初始化 (5.2.1) 成功后再调用。app 响应为 onRoomJoin (4.2.2), 若结果是加入成功, 那么各个模块如音频、视频、投票、插播、桌面共享等会相应地加载, 且相关的

状态和控制操作将在各自的回调里面通知，正式开启了直播的所有交互。如果失败则无后续响应。
`taskRet` 可以传入 `null`，也可以实现之。

5.2.4 `leave(boolean isClose, OnTaskRet taskRet)` 退出直播间

退出直播（课堂）。加入直播（课堂）成功后退出时调用，`isClose(true)`是代表结束整个直播（课堂）只有组织者有这样的结束权限，其他参与者只能够退出而不能结束，因此非组织者只能传入 `false`，组织者用户可根据实际情况“结束”或“仅退出”。

响应为 4.2.3，得到 4.2.3 后可以进行一些数据清理，对象释放。如 `RtSdk` 的 `release`。便于下次加入。

注：直播（课堂）里面的相关操作接口均应该在加入与退出之间进行，其他时间均视为无效。

5.2.5 `release(OnTaskRet taskRet)` 销毁本地直播间资源

释放对象，销毁本次加入创建的相关数据。一般在 `onRoomLeave` 回调里面在调用，`app` 也可以选择一个适当的时机调用。`taskRet` 响应后说明释放完成。调用如下：

// 释放房间实例，`onTask` 响应之后才能重新加入

```
rtSdk.release(new OnTaskRet() {  
  
    @Override  
    public void onTask(boolean ret, int id, String desc) {  
        // 房间从 init-join-leave-release 到此结束  
        toast("释放完成，你可以重新加入");  
        // app 应该在此时关闭界面，至于触发退出到响应执行到此的时间间隔，app 可  
        //以做一个对话框来增加交互的友好性  
    }  
});
```

5.2.6 `getAllUsers()`获取用户列表

获取用户列表。`RtSdk` 简单维护了用户列表，`app` 可以在加入成功后取得该列表，用户加入、退出、状态变更会同步到该列表中。`app` 也可以自行维护用户列表，参考 4.2.6、4.2.7、4.2.8。

5.2.7 `getUserById(long userId)` 获取用户

获取 `userId` 对应的用户信息。

5.2.8 `getSelfUserInfo()`获取个人用户信息

获取当前用户的用户信息。需要加入成功后获取有效。

5.2.9 `roomHandup(String data, OnTaskRet taskRet)` 直播间举手

举手操作。当前客户端举手，`data` 没有实际意义，传入"" 就行。其他客户端响应为 4.2.15。

5.2.10 `roomHanddown(boolean isAll, OnTaskRet taskRet)` 直播间取消举手

取消举手。`isAll` 为 `false` 时仅当前客户端的举手状态，`isAll` 为 `true` 时将会取消直播间（课堂）的所有举手状态。

5.2.11 setVideoCallback(IVideoCallback videoCallback) 设置视频回调句柄

设置视频回调，一般在 join 之前设置。设置以后一旦有视频相关的消息通知就会回调给 app，app 可以做相应的处理。具体的回调参考 IVideoCallback。

5.2.12 setLocalVideoView(ILocalVideoView localVideoView) 设置本地视频采集的 UI

设置本地视频采集的 UI 句柄 localVideoView。如果只做观看端，请忽略掉该接口，如果需要进行视频交流，一定得调用并传递一个 LocalVideoViewEx。

5.2.13 displayVideo(long userId, OnTaskRet onTaskRet) 观看某用户视频

观看 userId 的视频。当 userId 的视频（摄像头）打开以后，所有客户端包括本客户端可以观看它（userId）的视频数据。不调用观看，是不会显示该用户的视频的。

5.2.14 unDisplayVideo(long userId, OnTaskRet onTaskRet) 取消观看某用户视频

取消对 userId 视频的观看，服务端不发送 userId 的视频数据，本质上减少视频数据的网络流量，与 displayVideo 成对调用。

5.2.15 setAsCallback(IAsCallback asCallback) 设置桌面共享回调句柄

设置桌面共享回调。一般在 join 之前调用。若有与桌面共享相关的消息，则会通过回调通知 app，app 根据自身功能处理回调。详情参考 IAsCallback。

5.2.16 setLodCallback(ILodCallback lodCallback) 设置插播（媒体共享）回调句柄

设置插播回调，一般在 join 之前调用。若有与插播相关的消息，则会通过回调通知 app，app 自行处理。详情参考 ILodCallback。

5.2.17 setAudioCallback(IAudioCallback audioCallback) 设置音频回调句柄

设置音频回调，一般在 join 之前调用。当有音频相关的消息时，sdk 通过回调通知 app，app 自行处理。详情参考 IAudioCallback。

5.2.18 audioOpenMic(OnTaskRet taskRet) 打开设备 mic

打开本客户端的音频设备 mic。一种情况是本客户端需要进行语音交流时调用打开 mic；另一种情况是被要求打开，如直播中被组织者请求打开 mic 的时候需要本客户端调用该函数来打开 mic。

5.2.19 audioCloseMic(OnTaskRet taskRet) 关闭设备 mic

关闭本客户端的音频设备 mic，和 openSelfAudio 成对调用，客户端自身关闭或被请求关闭的时候调用。与 audioOpenMic 成对调用。

```
如：public void onClick(View v) {
        if (isMicOpened) {
            //调用返回基本上为 true，要通过 onTaskRet 来判断最终结果
            if (!IrtSdk.audioCloseMic(new OnTaskRet()) {
                @Override
                public void onTask(boolean ret, int id, String desc) {
```

```

        if (ret) {
            isMicOpened = false;
            toast("关闭 mic 成功");
        } else {
            toast("关闭 mic 失败");
        }
    }
}))) {
    toast("关闭 mic 失败");
}
} else {
    if (!IrtSdk.audioOpenMic(new OnTaskRet() {
        @Override
        public void onTask(boolean ret, int id, String desc) {
            if (ret) {
                isMicOpened = true;
                toast("打开 mic 成功");
            } else {
                toast("打开 mic 失败");
            }
        }
    }))) {
        toast("打开 mic 失败");
    }
}
}
}

```

5.2.20 audioOpenSpeaker(OnTaskRet taskRet)打开设备扬声器

打开当前设备的扬声器, sdk 默认是开启扬声器的。如果用户要调节音量, 由 app 自行通过 android sdk 相关 api 来控制媒体音量。由 taskRet 返回真正操作结果。

5.2.21 audioCloseSpeaker(OnTaskRet taskRet)关闭扬声器

关闭当前设备的扬声器。与打开对应, 一开一关, 由 taskRet 返回真正操作结果。

如: public void onClick(View v) {

```

    if (isSpeakerOpened) {
        // 关闭扬声器 调用返回基本上为 true, 要通过 onTaskRet 来判断最终结果
        if (!IrtSdk.audioCloseSpeaker(new OnTaskRet() {
            @Override
            public void onTask(boolean ret, int id, String desc) {
                if (ret) {
                    isSpeakerOpened = false;
                    toast("关闭扬声器成功");
                } else {
                    toast("关闭扬声器失败");
                }
            }
        }))) {
            // 失败处理
        }
    }
}

```

```

    }
    ))) {
        toast("关闭扬声器失败");
    }
} else {
    // 打开扬声器
    if (!rtSdk.audioOpenSpeaker(new OnTaskRet() {
        @Override
        public void onTask(boolean ret, int id, String desc) {
            if (ret) {
                isSpeakerOpened = true;
                toast("打开扬声器成功");
            } else {
                toast("打开扬声器失败");
            }
        }
    }))) {
        toast("打开扬声器失败");
    }
}
}
}

```

5.2.22 openUserAudio(long userId, OnTaskRet taskRet) 打开用户音频

请求打开 userId 对应用户的音频，即音频设备 mic。

5.2.23 closeUserAudio(long userId, OnTaskRet taskRet) 关闭用户音频

请求关闭 userId 对应用户的音频，即音频设备 mic。

5.2.24 videoOpenCamera(OnTaskRet taskRet)打开本地摄像头

打开本地相机进行视频采集，一般由本客户端主动调用或者组织者要求打开的时候调用，两种情况都需要客户端调用 videoOpenCamera 来进行视频采集，前提是 LocalVideoViewEx 已经设置好。

5.2.25 videoCloseCamera(OnTaskRet taskRet)关闭本地摄像头

关闭相机停止视频采集，与 videoOpenCamera 同样的两种情况下都需要调用 videoCloseCamera 来关闭视频。返回值基本上为 true，要根据 taskRet 来准确判断操作的结果，与 videoOpenCamera 成对调用。

如： public void onClick(View v) {
 if (isVideoOpened) {
 if (!rtSdk.videoCloseCamera(new OnTaskRet() {
 @Override
 public void onTask(boolean ret, int id, String desc) {
 if (ret) {
 toast("关闭本地视频成功");
 isVideoOpened = false;
 } else {


```

        toast("关闭本地视频失败");
    }
}
))) {
    toast("关闭本地视频失败");
}
} else {
    if (!IrtSdk.videoOpenCamera(new OnTaskRet() {
        @Override
        public void onTask(boolean ret, int id, String desc) {
            if (ret) {
                isVideoOpened = true;
                toast("打开本地视频成功");
            } else {
                toast("打开本地视频失败");
            }
        }
    })) {
        toast("打开本地视频失败");
    }
}
}
}
}

```

5.2.26 videoActive(long userId, boolean isActive, OnTaskRet taskRet)设置直播视频

在直播（webcast）中用户视频打开的情况下，可以将该用户设置为直播视频此时 `isActive` 传值 `true`，相反，取消直播视频传 `false`。

在 `training` 中，如果老师上讲台打开视频的时候将老师的视频进行 `active` 作为直播视频，此时 `isActive` 传值是 `true`；当老师关闭视频的时候，应当取消直播视频，此时 `isActive` 传值为 `false`。

5.2.27 setChatCallback(IChatCallBack chatCallBack)设置聊天回调

设置聊天回调。设置后 `sdk` 收到聊天消息后会以回调通知 `app` 新消息。详情参考 `IChatCallBack`。

5.2.28 chatWithPublic(ChatMsg msg, OnTaskRet taskRet)发送公共聊天消息

发送公共聊天消息，房间里面所有人都能收到。`chatText`（`ChatEditText.getChatText()`）为要发送的文本消息内容，`richText`（`ChatEditText.getRichText()`）为要发送的富文本消息内容。如下构造一个聊天消息：

```

String msgId = UUID.randomUUID().toString();//消息 id,即加上 uuid
//公聊消息 msgId, chatText, richText 都不为空才有效
ChatMsg msg = new ChatMsg(chatText, richText, ChatMsg.CHAT_MSG_TYPE_PUBLIC, msgId);

```

注：聊天输入与显示需要添加聊天资源库 `RTRoutineLib`，并在聊天消息要显示或输入前初始化聊天资源：
`ChatResource.initChatResource(Context context)`，调用一次即可。

5.2.29 chatWithPersion(ChatMsg msg, OnTaskRet taskRet)发送私人聊天消息

向指定用户发送私人消息。**chatText** (ChatEditText.getChatText()) 为要发送的文本消息内容, **richText**(ChatEditText.getRichText())为要发送的富文本消息内容。**userId** 接收消息的用户 Id。

```
String msgId = UUID.randomUUID().toString();//消息 id,即加上 uuid 即可
```

```
//私聊消息 msgId, chatText, richText 都不为空且接收人 id 正确才有效
```

```
ChatMsg msg = new ChatMsg(chatText, richText, ChatMsg.CHAT_MSG_TYPE_PRIVATE, msgId);
```

```
msg.setReceiverId(userId);//接收人的用户 id
```

注: 聊天输入与显示需要添加聊天资源库 RTRoutineLib, 并在聊天消息要显示或输入前初始化聊天资源: `ChatResource.initChatResource(Context context)`, 调用一次即可

5.2.30 setVoteCallback(IVoteCallBack voteCallBack)设置投票、试卷回调句柄

设置投票调查(试卷)回调。设置后 sdk 会将投票调查(试卷)相关的消息以回调通知 app, app 自行处理相关回调事件。

5.2.31 voteAdd(VoteGroup voteGroup, final OnTaskRet taskRet)添加一个投票

添加一个投票问卷(试卷), 其他客户端则会响应 4.8.2。

5.2.32 voteDel(String strVoteID, OnTaskRet taskRet)删除一个投票

删除一个投票调查, 其他客户端则响应 4.8.3。

5.2.33 votePublish(String strVoteID, OnTaskRet taskRet)发布投票

发布投票。答题开始, 其他客户端响应 4.8.4。

5.2.34 votePublishResult(String strVoteID, OnTaskRet taskRet)发布投票结果

发布投票结果。结果公布, 其他客户端响应 4.8.5。

5.2.35 voteSubmit(VoteGroup voteGroup, OnTaskRet taskRet)提交(参与)投票

提交答案。完成投票问卷或试卷的时候提交, 组织者或老师会响应 4.8.6。

5.2.36 voteDeadline(String strID, OnTaskRet taskRet)终止投票

终止答题。答题时间到, 所有答题的客户端会响应 4.8.7, app 终止答题。

5.2.37 votePostUrl(String strURL, OnTaskRet taskRet)第三方投票

发起第三方投票。strURL 为第三方投票地址。答题客户端响应 4.8.8。

5.2.38 cardSubmit(int[] itemIds, OnTaskRet taskRet)提交答题卡

当答题卡发布后, 通过 cardSubmit 提交, itemIds 是答题卡中被选择的选项组成的数组。

5.2.39 setDocView(GSDocView docView)设置文档显示的 UI

设置文档显示 UI, 如果需要显示文档, 则将 docView 传递给 sdk, app 可以自由布局和控制其大小。

5.2.40 setGSDocViewGx(GSDocViewGx docView) 设置支持 ppt 动画的显示文档的 UI

设置支持 ppt 动画的文档显示控件，与 setDocView 不同的是能够支持 ppt 动画。往后将逐步废除 GSDocView 的使用，而采用新的 GSDocViewGx。

5.2.41 getDocModule()获取文档模块操作句柄

获取文档通信句柄，返回 IDocModule 句柄。App 可以使用该句柄来操作文档，如翻页、同步、拍照上传等。

5.2.42 setQACallback(IQACallback qaCallback)设置问答回调

可以根据需要设置问答回调，使得 app 可以接收新提问、提问回复、提问发布等。

5.2.43 qaAddQuestion(String question, OnTaskRet taskRet) QA 提问

提问，即发送一个问题给组织者，由组织者分配问题的回复。question 为提出的问题。taskRet 可传空，需要监听发送结果则需要一个有效的实例。

5.2.44 qaAddAnswer(String questId, String answer, OnTaskRet taskRet) QA 回复

提问回复，当有人提问后，组织者对提问进行文字回复或者文字解答，即发送回复。questId 为问题 id，answer 为回复内容。taskRet 可传空，需要监听发送结果则传有效实例。

5.2.45 qaPublish(String questId,boolean isPublish,OnTaskRet taskRet) QA 发布

发布提问，当有人提问后，组织者把问题发布给 web 用户。questId 问题 id，isPublish 是否发布。taskRet 可传空，需要监听执行结果则传有效实例。

5.2.46 qaPush2Panelist(String questId, OnTaskRet taskRet) QA 推送嘉宾

推送给嘉宾，当有人提问后，组织者把问题推送给嘉宾。questId 问题 id。taskRet 可传空，需要监听执行结果则传有效实例。

5.2.47 qaAssign2Panelist(String questId, long userId, OnTaskRet taskRet) QA 指派给嘉宾

指派给嘉宾，当有人提问后，组织者把问题指派给嘉宾进行回复。questId 问题 id，userId 为被指派嘉宾的用户 id，taskRet 可传空，需要监听执行结果则传有效实例。

5.2.48 qaSetReplyByVoice(String questId, OnTaskRet taskRet) 将提问置为语音回复

将提问置为语音回复，当有人提问后，回复者可以将该问题设置为语音回复。questId 问题 id，taskRet 可传空，需要监听执行结果则传有效实例。

5.2.49 qaNowReplyingByVoice(String questId, boolean isNowReplying,OnTaskRet taskRet) 将问题置为语音回复状态

将问题置为语音回复状态，回复者在语音回复的时候设置回复状态。questId 问题 id，isNowReplying true 为“正在进行语音回复”，false 为“取消语音回复”即回复完成，taskRet 可传空，需要监听执行结果则传有效实例。

5.2.50 qaNowReplyingByText(String questId, int duration, OnTaskRet taskRet) 将问题置为正在进行文本回复

将问题置为正在进行文本回复，回复者在文字回复的时候设置回复状态。questId 问题 id，duration 回复时间，一般客户端不使用这个值，askRet 可传空，需要监听执行结果则传有效实例。

5.2.51 **qaSetState(boolean isAutoPushQuestion, boolean isAutoPublishQuestion, boolean isQAEnable, OnTaskRet taskRet)** 问答权限通知

组织者对问答操作的控制，isAutoPushQuestion 自动推送问答，组织者需要关注，如果为 true 则自动将收到的提问进行推送，否则在适当时间推送，isAutoPublishQuestion 自动发布问答，isQAEnable 提问允许 true，允许提问，false 禁止提问。组织者调用后，其他客户端响应 IQACallback 的 onQaSettingUpdate 函数。

注：isQAEnable 为 false 的时候，app 要禁止用户提问，当为 true 的时候允许用户提问。

5.2.52 **loadLibrarys()** 加载动态库（提高加入速度）

优先加载动态库，在程序启动的适当时机优先加载动态库，可以提高直播的加入速度。

5.2.53 **roomRollcall(int duration, OnTaskRet taskRet)** 发起点名

本接口针对老师或组织者的，用于向直播间发起点名。duration 为点名时间，即 duration 秒后结束本次点名。

5.2.54 **roomRollcallAck(OnTaskRet taskRet)** 点名应答

当收到组织者或老师发起的点名时候，在点名的时间范围内进行应答，超时后请不要再应答。将被认为没有答到。

5.2.55 **roomSaveRollcallResult(OnTaskRet taskRet)** 发布点名统计

本接口针对老师或组织者的，点名完成后发布的点名统计。

5.2.56 **roomPublish(byte state, OnTaskRet taskRet)** 直播控制

控制类型：开启直播、暂停直播、恢复直播、停止直播。

例如：开启直播 State.S_RUNNING（恢复直播）

暂停直播 State.S_PAUSED

停止直播 State.S_STOPPED

//开启/恢复直播

```
rtSdk.roomPublish(State.S_RUNNING.getValue(), null);
```

5.2.57 **roomRecord(byte state, OnTaskRet taskRet)** 录制控制

控制类型：开启录制、暂停录制、恢复录制、停止录制。

例如：开启录制 State.S_RUNNING（恢复录制）

暂停录制 State.S_PAUSED

停止录制 State.S_STOPPED

//开启/恢复录制

```
rtSdk.roomRecord (State.S_RUNNING.getValue(), null)
```

5.2.58 **roomRename(long userId, String newName, OnTaskRet taskRet)**修改用户名

修改用户名，userId 为对应要修改的，一般情况下 app 提供修改自己用户名称的功能。

5.2.59 roomGrantHost(long userId, OnTaskRet taskRet) 设置组织者（老师）

将用户（userId）设置（授权）为组织者或老师，不适用于小班课。

5.2.60 roomGrantPresentor(long userId, OnTaskRet taskRet)设置主讲

将用户（userId）设置为主讲，不适用于小班课。

5.2.61 roomPushUserCount2Web(boolean isPush, OnTaskRet taskRet)向 web 用户推送在线人数

调用后可以让 web 加入的用户显示在线人数。

5.2.62 roomPushUserList2Web(boolean isPush, OnTaskRet taskRet)向 web 用户推送用户列表

调用后可以让 web 加入的用户显示用来列表。

5.2.63 roomNotifyBroadcastMsg(String msg, boolean isNeedCache, OnTaskRet taskRet) 发送广播消息

发送系统广播消息，msg 消息内容，isNeedCache 是否缓存。缓存，消息发后加入的 web 用户会收到，不缓存收不到。

5.2.64 roomSetData(String key, long value, OnTaskRet taskRet) training 课堂席位、状态等控制

只用在 training 站点小班课当中，调用之后所有用户会响应 onRoomData(String key, long value)。控制讲台、提问席、聊天等操作。其中 key 的取值定义在 RTRoomDataKey 当中：

```
String KEY_USER_ROSTRUM = "user.rostrum";    //讲台
String KEY_USER_ASKER = "user.asker";        //提问席 1
String KEY_USER_ASKER_1 = "user.asker.1";    //提问席 2
String KEY_USER_ASKER_2 = "user.asker.2";    //提问席 2
String KEY_USER_ASKER_3 = "user.asker.3";    //提问席 3
String KEY_CLASS_MODE = "class.mode";        //班级类别（小班）一般用不着
String KEY_CHAT_MODE = "chat.mode";          //聊天模式 学生一般用不着
```

例:key 为 KEY_USER_ROSTRUM 代表讲台操作,对应的value 不为0 则表示用户 id 为 value 的用户上讲台, 为0 则表示此前讲台上的用户下讲台。同理提问席 1-4 以此类推。

5.2.65 roomEjectUser (long userId, boolean isForbidIp, OnTaskRet taskRet)踢人

直播间或课堂踢人操作，userId 为被踢人 id。isForbidIp true 代表踢人且封禁 ip，封禁后不能再次加入，false 普通踢人，可再次加入。

5.2.66 getIDCs() 优选网络选择列表

做网络优选功能使用，取到可以选择的列表 IDCInfo[]，然后进行界面上的展示。

```
IDCInfo {
    private String id;        //网络 id
    private String name;      //网络名称
}
```

5.2.67 `getCurIDC()`获取当前正在使用的网络 id

做网络优选列表时候匹配当前使用的网络，只返回正在使用的网络 id。

5.2.68 `setCurIDC(String idcId, OnTaskRet taskRet)`设置新的网络 id

选择新的网络后进行调用，idcId 请保证是从 getIDCs 里面取到的 idcId 且有效。taskRet 可为 null。
设置新的网络后会重连（onRoomReconnecting）

5.2.69 `roomOpenUserVideo(long userId, OnTaskRet taskRet)`打开他人视频命令

一般组织者或老师才具备向他人发出打开他人视频的命令，和 openUserAudio 都属于直播间内的命令。注意仅向他人发出这个命令，需要他人具备能打开的条件。

A 打开 B 的视频：

- 1、B 需要事先设置 localvideoview 给 sdk，随时待命被打开；
- 2、A 调用 `roomOpenUserVideo(B.getId(), null)`;
- 3、B 会响应 `onVideoCameraOpened` `onVideoJoin(B)`
- 4、A 同时也会响应 `onVideoJoin(B)`
- 5、A 在 4 的条件下订阅 B 的视频（`displayVideo(B.getId(), null)`），A 用 `GSVideoView` 就能看到 B 的视频。

5.2.70 `roomCloseUserVideo(long userId, OnTaskRet taskRet)`关闭他人视频命令

同样是组织者或老师才具备调用条件或者其他特殊场景时调用。和 `roomOpenUserVideo` 是对应关系，需要在他人视频已经打开的情况下操作。

5.2.71 `getLiveInfo()`获取直播简介、主讲、议程

app 已经调用过 `initWithParam` 过后，如果需要这些信息，可以调用 `getLiveInfo` 来获得这几个信息，信息通过 `onLiveInfo` 返回。

5.2.72 `initMedalPraise(IMedalPraiseCallback callback, int defMedal, int defFavour)`点赞勋章初始化

初始化勋章或点赞需要按需求以及后台的启用禁用来初始化，如果没有点赞和勋章的需求，就无需初始化。如果后端禁用了勋章或点赞功能，那么对应功能也是初始化不成功的。

初始化时机，加入直播或课堂成功以后。

callback:勋章和点赞的回调实现。

defMedal:初始化勋章数目，-1 取默认值，-2 不初始化勋章，非负数为自定义初始化的数值。

defFavour:初始化点赞数目，-1 取默认值，-2 不初始化点赞，非负数为自定义初始化的数值。

初始化勋章后会响应 callback 的 `onPraiseInfo` 函数（4.12.1）

isFavourEnable isMedalEnable 的取值是从 IRoomCallbck 的 onSettingSet 里面来的：

```
public void onSettingSet(String key, int val) {  
    //勋章功能是否开启  
    if (RTSharedPref.KEY_MEDAL_ENABLE.equals(key)) {  
        isMedalEnable = val != 0;  
    }  
    //点赞功能是否开启  
    else if (RTSharedPref.KEY_FAVOUR_ENABLE.equals(key)) {  
        isFavourEnable = val != 0;  
    }  
}
```

```
}
```

例如:

```
//如果有勋章点赞的需求, 才根据后台开关初始化点赞和勋章
if (isFavourEnable || isMedalEnable) {
    //-1 为取默认值, -2 代表不初始化该功能
    rtsdk.initMedalPraise(medalPraiseCallbck, isMedalEnable ? -1 : -2, isFavourEnable ?
-1 : -2);
}
```

5.2.73 sendPraise(String praiseType, long toUserId, String toUserName, String comment, OnTaskRet taskRet)发送勋章或点赞

发送勋章或点赞, praiseType 代表勋章 or 点赞。

toUserId 接收者用户 id

toUserName 接收者用户名

comment 附言说明, 可以为空 (“ ”)

调用 (发送) 后, 所有 rtsdk 用户都会收到通知 (4.12.3)

5.2.74 getPraiseRecvList(String praiseType, int maxUser, OnTaskRet taskRet)获取勋章排行

如果需要显示排行榜, 调用该接口后, 响应排行榜信息 (4.12.2)

praiseType 勋章类型或点赞类型 (一般是勋章)

maxUser 排行榜前多少的意思。例如 10, 即 top10, 将返回最多 10 个数据。

5.3 GSVideoView

GSVideoView 是视频显示的 UI, app 可以通过布局文件布局, 也可以用代码创建。也可以任意改变其大小。可以显示的内容有: 视频、插播视频、桌面共享。其中画面会自动适应其大小显示。

5.3.1 setDefColor(int color)视频显示设置

设置视频底色, 其中默认是黑色。

5.3.2 onReceiveFrame(byte[] data, int width, int height)视频显示 UI 接收数据

接收并显示相关数据 data, width 为视频宽, height 为视频高, 这些参数都由 IVideoCallBack 和 IAsCallBack 回调给出, app 只需将数据往 GSVideoView 的该函数填入即可。

5.3.3 renderDefault()视频 UI 清屏

清除视频画面使用的, 一般 app 希望清除视频最后一帧的时候可以调用, 而有的 app 则期望保留最后一帧。因此清除画面后视频会显示为 5.3.1 所给的颜色。

5.3.4 setRenderMode(RenderMode renderMode)设置视频显示模式

设置视频显示模式:

```
enum RenderMode{
    RM_FILL_XY,    //填充显示即充满窗口
    RM_ADPT_XY,    //适应窗口显示
    RM_CENTER,     //居中显示, 这种只有视频分辨率小于窗口本身的时候设置有效果
```

RM_FILL_CENTER_CROP //以小边放大或缩小 并裁剪后充满窗口(一般用将不同分辨率统一成一定分辨率的情况)

```
}
```

5.3.5 getRenderMode()获取视频显示模式

和 setRenderMode(RenderMode renderMode)配合使用。

5.3.6 renderDrawble(Bitmap defBitmap,boolean isFillXY)渲染默认图片

一般为默认图片即初始图片的渲染，一旦有真正的视频渲染后失效。

5.3.7 getVideoWidth()获取视频宽度

返回视频宽度。

5.3.8 getVideoHeight()获取视频高度

返回视频高度。

5.4 GSDocViewGx (GSDocView)

GSDocViewGx 是用来显示文档的。默认为观看模式。可以在布局文件中使用，也可以在代码中创建。最后将其传递给 SDK。若需要做文档标注，如画直线、矩形、自由曲线等，就要调用 GSDocViewGx 的相关 api 来切换操作类型；若只是接收其他客户端的文档显示，则只要将其传递给 SDK 即可。文档的显示方式（内容与 view 的适应）可根据需要选取或 app 自由变换，已达到想要的显示效果。**注意 GSDocViewGx 与 GSVideoView(SurfaceView)重叠（画中画）显示会有屏幕共用的现象，因此请尽量避免两个重叠显示的设计，若 GSVideoView 显示的时候总在 GSDocViewGx Z 方向上方显示的话，这种画中画是可以支持的。**

5.4.1 setPaintColor(int color)设置文档画笔工具的颜色

设置文档即将要使用的画笔颜色，颜色为 argb 位序。

5.4.2 setStrokeWidth(LINE_SIZE width)设置文档画笔工具的线粗

设置文档即将使用的画笔粗细。

其中 enum LINE_SIZE {
L, //1 像素 已经废弃
M, //3 像素
H, //6 像素
}

5.4.3 setAnnoMakeType(DrawMode mode)设置即将开始的文档标注类型

设置文档的标注类型。注意文档标注只有组织者（直播）、嘉宾（直播）、老师（training）、助教（training）这些角色或经批准的其他角色才有权限进行操作。

其中 enum DrawMode {
PEN, //画笔画图模式
ERASE, //橡皮擦模式
ERASE_ALL, //全部清除
DOC_TIP, //文档指示(圆圈或鼠标箭头)
RECT, //矩形


```

        LINE,        //直线
        HLIGHT;     //荧光笔
    }

```

5.4.4 setCtrlMode(CtrlMode mode)设置文档 UI 的控制模式

设置文档操作控制模式。控制有两种，“观看”和“编辑”。默认为观看模式，可以缩放和拖动文档。编辑则是指进行标注的添加修改，调用 setAnnoMakeType 会自动切换为编辑模式，当标注完成或角色失去标注的功能后需要调用 setCtrlMode (CtrlMode.SIGHT) 恢复到观看模式。

```

enum CtrlMode {
    EDIT,    //编辑模式，拖拉无效，标注有效
    SIGHT    //观看模式，拖拉有效，标注无效
}

```

5.4.5 setOnDocViewClickedListener(OnDocViewEventListener l)

设置文档 view 的事件监听，具体的事件参考 OnDocViewEventListener。

5.4.6 undo(long userId)文档注释撤销

撤销自身添加的标注。

5.4.7 showFillView()文档内容充满 docview 显示

文档显示方式，文档内容填满文档 view。如果文档长宽比例和文档 view 长宽比例不一致的情况会有不同程度的拉伸。

5.4.8 showAdaptView()文档内容适应 docview 显示

文档显示方式，文档内容会根据文档 view 的大小来适合显示，这种方式能够保证内容等比拉伸，而不会变形。

5.4.9 showSourceScall()文档内容原始大小显示

文档显示方式，文档内容按照原始大小显示，如果内容过大，则只显示部分，可以随意拖动。

5.4.10 showAdaptViewWidth()文档内容适应 docview 宽度显示

文档显示方式，文档内容充满文档 view 的宽度显示，如果高度比过大则可上下拖动，否则 Y 方向居中显示。

5.4.11 showAdaptViewHeight()文档内容适应 docview 高度显示

文档显示方式，文档内容充满文档 view 的高度显示，如果宽度比过大则可左右拖动，否则 X 方向居中显示。

5.4.12 setBackgroundColor(int color)设置文档背景色

文档背景色设置，在文档不足的地方（即文档内容与 docview 之间的地方）显示。color 为 argb 位序，如果是定义资源文件中则需要通过转换，不能传入 resId。

5.4.13 forbidZoomGestrue(boolean isFobid)手势禁用

如果不需要文档 view 的缩放或拖拽，可以禁用掉手势。

5.5 IDocModule 文档操作句柄

SDK 里面文档模块的通信操作句柄（RtSdk 的 `getDocModule()` 获取），加入直播（课堂）成功后可以使用。可以得到当前打开的文档和已经打开的文档，并可以切换文档的显示。

PduDoc: 代表一个 ppt、word、png、pdf 等任意类型的文档，主要调用函数如下：

- `getDocName()` //返回文档的名称
- `getDocType()` //返回文档类型
- `getPageCount()` //返回文档总页数
- `getPages()` //返回文档内所有页的信息列表(List<PduPage>)
- `getCurPage()` //返回这个文档正在显示的那一页(PduPage)，默认是第一页。
- `getCurrentPageIndex()` //返回这个文档当前显示页的页码
- `getNextPage()` //返回当前页对应的下一页(PduPage),用于翻页 `gotoPage`，返回 null 说明当前是最后一页（此时可以找下一个文档 PduDoc 的当前页并翻到该页）
- `getPrePage()` //返回当前页对应的上一页(PduPage),用于翻页 `gotoPage`，返回 null 说明当前是第一页

PduPage: 代表一个文档中一页。`getPageName()` 返回该页的名称。

5.5.1 getDocs()获取文档列表

获取当前已经打开的文档。这个的作用一般是做界面使用。返回 List<PduDoc>。

5.5.2 getCurrentDoc() 获取当前显示的文档

获取当前正在显示的文档。用处一般是显示文档标题。返回 PduDoc。

5.5.3 isCurrentDoc(PduDoc doc)

判断 doc 是否为当前正在显示的文档。

5.5.4 gotoPage(PduPage page,boolean syn,OnTaskRet taskRet)翻页

切换文档。page 为要切换到的文档页，即为任意文档的任何一页。syn 为 true 会同步切换其他客户端，即直播间里面其他人会一起翻页，false 只在本地切换显示，其他人不会一起翻页。syn 注意，一般是有限权(老师、组织者、主讲)的操作者才能传 true，例如 traning 里面上讲台的人需要同步自己的当前文档。

- 1、当前文档取出对应的下一页，翻到下一页 `page = currentDoc.getNextPage()`。
- 2、当前文档取出对应的上一页，翻到上一页 `page = currentDoc.getPrePage()`。

5.5.5 openDoc(String name, Bitmap bitmap)打开、上传文档

打开文档。主要做拍照、相册上传使用的，name 为文档名称如 xxx.拍照(1).jpeg（拍照：规则为 userName + .拍照 + （拍照计数）+ .jpeg；相册：规则为文件原是名称包含扩展名），注意：图片的大小限制在 512000 个像素以内。

如果是老接口 `openDoc(String name, byte[] data, int w, int h)` data 为图片 jpeg 格式的数据，w 为图片宽，h 为图片高。

5.5.6 synCurrentDoc()同步文档

同步文档。当得到做标注权限的时候调用。如 training 中插播完成、老师、助教上讲台。

5.5.7 getPhotoCount()图片、照片计数

拍照计数。配合拍照上传使用。

5.6 RtSimpleImpl

RtSimpleImpl 是一个抽象类，实现了音、视频相关的回调，并将所有视频（直播视频、媒体共享（插播）视频、桌面共享）处理为一路视频。如果只做直播观看或听课的功能，那么这个是最佳选择，因为只需要简单的调用和实现就能够达到目的，参数也和 RtSdk 的一样。

即：`abstract class RtSimpleImpl implements IRoomCallBack, IVideoCallBack, ILodCallBack, IAsCallBack, IAudioCallBack`。除了音视频的其他回调通过 RtSdk `rtSdk = rtSimpleImpl.getRtSdk()`后，用 `rtSdk` 进行设置，例如聊天：`rtSdk.setChatCallback(chatCallback)`。

5.6.1 setVideoView(GSVideoView)设置视频显示的 UI

设置显示视频的 GSVideoView，布局大小可随意变更，显示内容和 web 是一样的。

5.6.2 setDocView(GSDocView docView)设置文档显示的 UI

设置文档显示 UI。

5.6.3 setGSDocViewGx(GSDocView docView) 设置支持 ppt 动画的文档显示控件

设置文档显示 UI，支持 ppt 动画，3.1 过后启用。

5.6.4 join(String logFileName,String joinParam)加入直播间

加入直播（课堂），`logFileName` 可以填为“”，`joinParam` 就是 4.1.1 响应的结果。可以重写 RtSimpleImpl 的 `onInit` 来确认加入状态。

5.6.5 leave(boolean isClose)退出直播间

退出直播（课堂），`isClose` 为 `true` 结束，`false` 仅离开退出。

5.6.6 onVideoStart()视频开始，视频加载成功

视频开始播放，抽象函数。

5.6.7 onVideoEnd()视频结束

视频停止播放，抽象函数。

5.6.8 onGetContext()

获取 `appcontext`，抽象函数。而且必须返回。

5.6.9 RtSdk getRtSdk()获取 RtSdk 句柄

获取 `rtSdk` 句柄，可以做其他的操作。

5.7 ChatEditText 聊天输入框

继承自 `EditText` 聊天输入框主要是针对聊天表情显示使用的，如果不需要聊天或不需要表情功能的请忽略。前提要加入表情资源库的引用且初始化聊天的表情资源。

5.7.1 insertAvatar(String avatar, int resId)向输入框插入一个表情

在表情选择完成后，将表情字符串和资源 id 传入即可在输入框内显示，实例：

```
public void selectAvatar(String avatar, int resId) {  
    chatEditText.insert(avatar, resId);  
}
```

5.8 MyTextViewEx 聊天消息富文本显示

继承自 TextView，主要支持聊天表情的显示，如果不需要聊天或不需要表情功能的请忽略。前提要加入表情资源库的引用且初始化聊天的表情资源。

5.8.1 setRichText(String richText)富文本显示

设置富文本消息内容，其中 richText 来自 IChatCallback 的响应,例如 onChatWithPerson(UserInfo userInfo, String msg, String richText)。

5.9 GSLocalVideoView(LocalVideoViewEx)视频采集预览 UI

这些 view 是为用户进行视频采集预览使用的。如果有打开摄像头的需求，这个 view 必须要用上，且要优先设置给 Sdk。注意，如果 targetSdkVersion >= 23 的情况下，app 应当要在调用 sdk 加直播接口之前申请好视频录制相关权限（android.permission.CAMERA android.permission.RECORD_VIDEO）。

GSLocalVideoView 相对于 LocalVideoViewEx 的特性：

- 1、采集时可以做到美颜效果，同时多了美颜的开关函数
- 2、对设备要求更高，
- 3、内存占用要高

5.9.1 doCameraSwitch 切换前后摄像头

在视频打开的情况下，可以进行前后摄像头切换。

5.9.2 setOrientation(int orientation)设置视频采集模式

默认是横屏采集，如果应用处于竖屏时设置相关的采集模式。

orientation 取值：

ILocalVideoView. **ORIENTATION_PORTRAIT** //竖屏采集会裁剪（app 竖屏），这种模式与 gensee 其他客户端配套使用时设置。宽比高要大

ILocalVideoView. **ORIENTATION_LANDSCAPE** //横屏采集（app 横屏）

ILocalVideoView. **ORIENTATION_PORTRAIT_UNCROP** //竖屏采集不裁剪（app 横屏），这种效果是宽比高要小。纯移动端发直播和观看的场景比较合适。

ILocalVideoView. **ORIENTATION_PORTRAIT_9_16** //竖屏采集裁剪 9: 16。采集时会采集到的图像以中心为参考，最大化的裁剪成为 9: 16，app 请保持竖屏采集，至于分辨率可以设置为 640*480 效果最好，最终图像是旋转之后采集成为 360*640。（只有 GSLocalVideoView 支持此模式）

ILocalVideoView. **ORIENTATION_PORTRAIT_16_9** //横屏采集裁剪 16: 9。采集时会采集到的图像以中心为参考，最大化的裁剪成为 16: 9，app 请保持横屏采集，至于分辨率可以设置为 640*480 效果最好，最终图像是裁剪成为 640*360。（只有 GSLocalVideoView 支持此模式）

5.9.3 setVideoSize(int width,int height)设置采集分辨率

设置视频采集的分辨率。注意：若正在采集（视频已打开）的情况下，需要先 videoCloseCamera 之后再 setVideoSize, 接着 videoOpenCamera 才有效。不要视图在视频已经打开的情况下随意 setVideoSize；不同的设备采集的能力不一样，请根据实际情况选择合适的分辨率。

一般入参为 320*240, 640*480, 1280*720，不同的设备实际的大小不一定是设置的大小。若需要，可通过 setOnCameraInfoListener 监听实际采集的尺寸。

5.9.4 switchBeauty(boolean isOpen) 美颜切换

这个函数是美颜效果的开关切换，只有 GSLocalVideoView 才有这个函数。默认就是开美颜效果的。

5.10 GenseeLog GenseeSDK 日志工具

GenseeLog 是 sdk 中的日志工具类，其中包括日志的处理和上传。注意工具类能做的是 sdk 内部流程日志，至于 app 的日志以及 crash 的日志需要由 app 自行处理和收集。

5.10.1 reportDiagonse(Context context, String desc,ServiceType serviceType)上传日志

该函数将 sdk 的日志上传到服务端。返回值是上传后日志的下载地址。已经有日志文件的情况下可以在任何时候进行上传

context:app context,日志的描述

desc: 该日志的描述或表现描述

serviceType: 和加入一样，webcast 站点填写 ServiceType.WEBCAST,training 站点填写

ServiceType.TRAINING

返回值: 日志的下载路径

6 代码实例

说明：这部分代码只供参考，具体还需要根据具体业务调整。

6.1 Domain 等参数初始化代码实例

```
InitParam p = new InitParam();  
//domain  
p.setDomain("demo.gensee.com");  
//编号（直播间号）  
p.setNumber("93857580");  
//站点认证帐号，根据实际情况可以填"  
p.setLoginAccount("站点认证帐号");  
//站点认证密码，根据实际情况可以填"  
p.setLoginPwd("站点认证密码");  
//昵称，供显示用  
p.setNickName("昵称");  
//加入口令，没有则填"  
p.setJoinPwd("加入口令");  
//站点类型 ServiceType.ST_CASTLINE 直播，  
//ServiceType.ST_MEETING 会议，  
//ServiceType.ST_TRAINING 培训  
p.setServiceType(ServiceType.ST_TRAINING);  
p.setK("你们的 K 值");//启用第三方认证的时候必须有效，没启用时忽略  
RtComp comp = new RtComp(getApplicationContext(),callback);  
comp.initWithGensee(p);
```

6.2 响应处理

6.2.1 RtComp.Callback onInit

```
// RtComp.Callback 响应，得到 joinParam 后进行 rtsdk 的初始化  
@Override  
public void onInit(String joinParam) {  
    //直播（课堂）初始化  
    rtSdk.initWithParam("", joinParam, roomCalBack);  
}
```

6.2.2 IRoomCallBack onInit

```
//直播（课堂）初始化响应  
@Override  
public void onInit(boolean result) {  
    RTLog.d(TAG, "OnInit = " + result);  
    //this 是实现了各个 callback 的，因此以 this 作为参数
```

```

if (result) {
    rtSdk.setVideoCallBack(this);
    rtSdk.setAudioCallBack(this);
    rtSdk.setLodCallBack(this);
    rtSdk.setChatCallBack(this);
    rtSdk.setVoteCallBack(this);
    rtSdk.setAsCallBack(this);
    //调用加入
    rtSdk.join(new OnTaskRet() {

        @Override
        public void onTask(boolean ret, int id, String desc) {
            RTLog.i("join ret = " + ret);
        }

    });
} else {
    //加入失败
}
}

```

6.2.3 IVideoCallBack onVideoJoin

```

@Override
public void onVideoJoin(UserInfo user) {
    if (user != null) {
        long userId = user.getId();
        // 插播视频打开
        if (UserInfo.LOD_USER_ID == userId) {
            rtSdk.displayVideo(userId, null);
        }
    }
}

```

6.2.4 IVideoCallBack onVideoJoin

```

@Override
public void onVideoLeave(long userId) {
    // 插播视频关闭
    if (UserInfo.LOD_USER_ID == userId) {
        rtSdk.unDisplayVideo(userId, null);
    }
}

```

6.2.5 IVideoCallBack onVideoDataRender

```

@Override
public void onVideoDataRender(long userId, int width, int height,

```

```

        int frameFormat, float displayRatio, byte[] data) {
//插播视频
    if (UserInfo.LOD_USER_ID == userId){
        videoViewLod.onReceiveFrame(data, width, height);
    }
//用户视频
    else {
        videoViewCast.onReceiveFrame(data, width, height);
    }
}
}

```

6.2.6 IAsCallBack onAsData

```

@Override
public void onAsData(byte[] data, int width, int height) {
//桌面共享
    videoViewAs.onReceiveFrame(data, width, height);
}

```

6.3 RtSimpleImpl

```

/**
 * 初始化一个 RTSDK 简单实现，重新定义一个类继承 RtSimpleImpl 也是可以的，
 * 并实现其 3 个抽象函数，其中 onGetContext 必须要返回 app context，音视频最佳选择。
 * 注：这里是在 onCreate 中创建的 simpleImpl，如果有屏幕变化引起 simpleImpl 重新被
 * 创建需要自行处理；回调线程为非 UI 线程，更新 ui 请在 UI 线程中处理。
 */

```

```

simpleImpl = new RtSimpleImpl() {

```

```

    @Override
    public Context onGetContext() {
        return getBaseContext();
    }

```

```

    @Override
    protected void onVideoStart() {
        GenseeLog.d(TAG, "onVideoStart");
    }

```

```

    @Override
    protected void onVideoEnd() {
        GenseeLog.d(TAG, "onVideoEnd");
    }
}

```



```

/**
 * result true 表示加入房间（直播间、会议室、课堂）成功
 */
@Override
public void onRoomJoin(boolean result, UserInfo self) {
    super.onRoomJoin(result, self);
    GenseeLog.d(TAG, "onRoomJoin");
}

/**
 * 直播状态 s.getValue() 0 默认直播未开始 1、直播中， 2、直播停止， 3、直播
暂停
 */
@Override
public void onRoomPublish(State s) {
    super.onRoomPublish(s);
    byte castState = s.getValue();
}
};

```