

Android AliMeetingUISDK 使用文档-1.2.0.1

1 概述

2 环境配置

3 SDK使用流程

3.1 SDK初始化

3.1.1 应用启动时初始化

3.1.2 进入会议前全局初始化

3.1.2.1 UT埋点相关

3.1.2.2 UI 配置

3.1.2.3 UI高级配置专项

3.2 获取入会信息

3.3 构建入会配置

3.3.1 必须配置

3.3.2 可选配置

3.3.2.1 日志输出配置

3.3.2.2 会议关键回调配置

3.4 加入会议

4 ReleaseNote

1.2.0.1

1.1.0.8

1.1.0.7

1.1.0.6

1.1.0.5

1.1.0.4

1.1.0.0

1.0.0.6

5 常见问题FAQ

1. SDK 对 Android 各种环境支持补充说明

1. 如何对入会成功率进行监听

2. 遇到crash如何进行反馈

3、遇到会议问题如何排查

附录1： UISDK 关键接口补充说明

AMUIMeetingDetail 会议详情

悬浮框相关

AMUIEdgeInsets 悬浮框显示约束

AMUIFloatingShowConfig 悬浮框显示大小/位置定义

IAMUIFloatingLayout 悬浮框UI接口

人员列表相关

IAMUIMemberListController 人员列表控制

1 概述

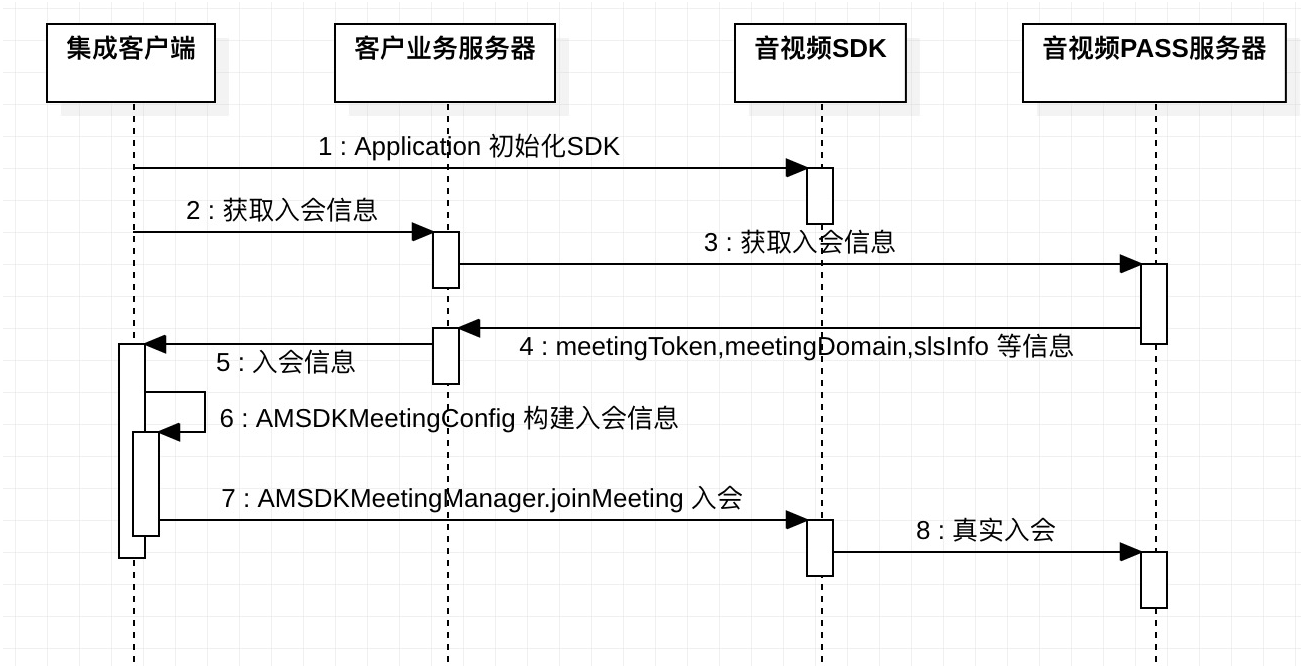
云视频会议AliMeetingUISDK 提供了一套用于加入视频会议的接口集合，开发者可以通过调用AliMeetingUISDK（以下简称 SDK），可以在自己应用中快速集成阿里音视频会议功能。

2 环境配置

请参考SDK集成文档

3 SDK使用流程

- 入会交互流程



3.1 SDK初始化

SDK初始化分为两个步骤，分别是

3.1.1 应用启动时初始化

用于监控应用生命周期，后续可以正常显示悬浮窗。调用方法

```
1 AliMeetingUIManager.initManager(app:Application)
```

3.1.2 进入会议前全局初始化

初始化会议日志、UI等全局性内容的定制。无需要每次入会都初始化，可以仅进行一次初始化

3.1.2.1 UT埋点相关

```
1 AliMeetingUIManager.pageTracker = object: IAMUIPageTrackerAdapter{
2 }
```

3.1.2.2 UI 配置

UI配置分为两个部分，一是静态资源的替换，二是用户交互控制

1 静态资源替换

- Activity 样式定制

请在app中覆盖style AliMeetingLight.MeetingDetail

- 定制指定图片资源

png 需要覆盖 drawable-hdpi, drawable-xhdpi, drawable-xxhdpi, drawable-xxxhdpi

- 1 定制屏幕共享用户小窗头像，请覆盖资源 ic_cloud_meeting_screen_share_default_avatar.png
- 2 定制会议室默认小窗头像，请覆盖资源 ic_cloud_meeting_room_default_avatar.png
- 3 定制用户默认头像，请覆盖资源 ic_cloud_meeting_user_default_avator.png

2. 用户交互控制

```
1 AliMeetingUIManager.uiController = object : AliMeetingUIManager.IAMUIMeetingUIController()
```

可定制内容：悬浮框显示位置、悬浮框显示样式、邀请人、会议详情、会议评价页面跳转、是否显示个人详情页、头像显示、反馈入口等。具体请参考接口定义。

AlimeetingUIManager.IAMUIMeetingUIController 的具体定义如下：

```
1
2 abstract class IAMUIMeetingUIController {
3
4     /**
5      * 获取悬浮窗距离屏幕边缘的距离，默认为上下留出状态栏与底部虚拟按键的
      距离
6      */
7     @MainThread
8     open fun getFloatingEdgeInsets(floatingType: AMUIFloatingType): AMUIEdgeInsets
9
10    /**
11     * 获取悬浮框显示信息，包括大小、圆角半径、默认显示位置
12     */
13    @MainThread
14    open fun getFloatingViewShowConfig(context: Context, floa
```

```

tingType: AMUIFloatingType): AMUIFloatingShowConfig
15
16
17     /**
18      * 创建悬浮框实例，移动逻辑内部统一处理
19      */
20     open fun createFloatingView(context: Context): IAMUIFloat
ingLayout
21
22
23     /**
24      * 会中点击邀请人操作，接入方在外部自己处理邀请逻辑
25      * 返回false 表示外部未处理，将显示默认UI
26      */
27     @MainThread
28     open fun onInviteAction(
29         view: View, activity: FragmentActivity, detail: A
MUIMeetingDetail?
30     ): Boolean
31
32     /**
33      * 会中点击会议标题，显示会议信息信息
34      * 返回false 表示外部未处理，将显示默认UI
35      */
36     @MainThread
37     open fun showMeetingDetail(
38         view: View, activity: FragmentActivity, detail: A
MUIMeetingDetail?
39     ): Boolean
40
41     /**
42      * 加载人员头像，
43      * view 默认类型为 ImageView，用户可以自定义View进行处理，通过替换
布局文件 layout_meeting_avatar 来进行
44      * 返回false 表示外部未处理，将显示默认UI
45      */
46     @MainThread
47     open fun loadAvatar(
48         view: AMUIAvatarLayout, url: String?, client: AMS
DKMeetingClient, userName: String?

```

```

49         ): Boolean
50
51
52         /**
53          * 聊天页面回调, chatPermissionGranted 表示当前会议是否有聊天权限
54          * 返回false 表示外部未处理, 将显示默认UI
55          */
56         open fun loadMeetingChat(
57             view: View, activity: FragmentActivity, chatPermi
58             ssionGranted: Boolean
59         ): Boolean
60
61         /**
62          * 获取人员列表显示控制器, 可以控制人员列表显示
63          */
64         open fun getMemberListController(): IAMUIMemberListContro
65             ller?
66
67         open fun showClientProfile(client: AMSDKMeetingClient)
68
69         /**
70          * maxOnlineClients 当前入会端在线期间, 会中最大在线人数
71          */
72         open fun startRateMeeting(meetingUuid: String, memberUui
73             d: String,
74             rateLink: String?, maxOnlineCli
75             ents: Int): Boolean = false
76
77         /** 用户手动改变是否结束后进入会议
78          * **/
79         open fun onRateMeetingConfigChanged(enableRateMeeting: Bo
80             olean)
81
82         /**
83          * 当前client 是否有详情信息, 如果有, 则在个人页面里面显示详情信息入
84          口, 否则隐藏入口
85          */
86         open fun hasClientProfileDetail(client: AMSDKMeetingClie
87             nt): Boolean = false
88
89
90
91

```

```

82         /** 用户打开我要反馈入口 */
83         open fun openFeedBack(meetingUuid: String, memberUuid: String, feedbackUrl: String): Boolean = false
84     }

```

3.1.2.3 UI高级配置专项

1. 未入会成员操作配置

通过实现IAMUIMeetingUIController.getMemberListController() 接口，提供自定义成员管理控制，实现未入会成员的呼叫功能，具体实现如下

```

1  AliMeetingUIManager.uiController = DemoUIController()
2
3  private class DemoUIController : AliMeetingUIManager.IAMUIMeeting
   UIController() {
4
5      override fun getMemberListController(): IAMUIMemberListCo
   ntroller? {
6          return DemoMemberListController()
7      }
8  }
9
10 private class DemoMemberListController : IAMUIMemberListContr
   oller() {
11
12     companion object {
13         private const val MENU_CALL_CLIENT = 11
14     }
15
16     private val offlineListener = object : IAMUIClientClickLi
   stener {
17         override fun onClientClicked(view: View, client: AMSD
   KMeetingClient) {
18
19             //自定义展示逻辑，可以利用UISDK的组件进行实现
20             val menuItems = arrayListOf<AMUIOperateMenu>()
21             menuItems.add(
22                 AMUIOperateMenu(

```

```

23             MENU_CALL_CLIENT,
24             view.context.getString(R.string.cloud_meeting_member_menu_call_him),
25             ContextCompat.getColor(
26                 view.context, R.color.cloud_meeting_operate_menu_high_light
27             ),
28             AMUIOperateMenuType.TYPE_TEXT
29         )
30     )
31     AMUIMemberOperateMenuFragment().setData(client, menuItems)
32     .setOnClickListener(object : IAMUIClientMenuClickListener {
33         override fun onClientMenuClicked(
34             view: View,
35             client: AMSDKMeetingClient,
36             menu: AMUIOperateMenu,
37             newConfigValue: Boolean
38         ) {
39             if (menu.menuId == MENU_CALL_CLIENT){
40                 //开始自定义的呼叫动作
41             }
42         }
43     })
44     .safeShow(activity, "operate_dialog")
45 }
46
47
48 }
49
50 override fun bindOfflineClient(
51     context: Context,
52     viewHolder: RecyclerView.ViewHolder,
53     client: AMSDKMeetingClient,
54     showOfflineAction: Boolean,
55     listener: IAMUIClientClickListener?
56 ) {
57     super.bindOfflineClient(context, viewHolder, client,
58         true, offlineListener)

```



```

58         }
59     }

```

2. 自定义头像组件

UISDK可以自定义头像加载组件，具体方式

- 覆盖 *layout_cloud_meeting_avatar.xml* 布局文件，注意保持 `android:id="@+id/avatarView"` 不变
- 设置 `AliMeetingUIManager.uiController`，实现 `IAMUIMeetingUIController.loadAvatar()` 接口，实现内部逻辑即可

```

1 private class DemoUIController : AliMeetingUIManager.IAMUIMeeting
  UIController() {
2
3     override fun loadAvatar(
4         view: AMUIAvatarLayout, url: String?, client: AMSDKMe
5         etingClient, userName: String?
6     ): Boolean {
7         val customView = view.contentView() //获取到 R.id.ava
8         tarView 对应的View 对象
9         //实现自己的头像加载逻辑
10        //注意返回值为true，否则将进入默认加载头像的逻辑
11        return true
12    }
13 }

```

3. 如何配置悬浮框在应用外展示

注意：默认为悬浮框不在应用外展示，现有逻辑是悬浮框显示时，点击悬浮框会跳转到会中页面，同时悬浮框消失，但是如果再应用外展示，由于现在很多机器限制唤起后台应用，会导致可能点击后无法跳转到会中页面，然后悬浮框消失，给用户带来不好体验。

- 1.0.0.6 版本增加了参数 `enableFloatingOutsideApp` 配置来控制强制应用外展示，
- 1.1.0.5 后取消了改参数，通过 `AliMeetingUIManager.IAMUIMeetingUIController` 来配置，具体配置方法如下

```

1 AliMeetingUIManager.uiController = object : AliMeetingUIManager.IA
  MUIMeetingUIController() {
2
3     override fun createFloatingView(context: Context): IAMUIFloat
      ingLayout {
4         //关键!!!，这里在创建好 IAMUIFloatingLayout 后，设置默认允许在
      应用外显示，用户也可以实现自己的 IAMUIFloatingLayout
5         return AMUIVideoFloatingLayout(context).setFloatingOutSide(tr
      ue)
6     }
7 }

```

3.2 获取入会信息

开发者向自己的服务请求入会信息，获取入会的必须信息。

```

1 data class AMUIMeetingDetailConfig(
2     val meetingAppId: String = "", //必填
3     val meetingToken: String = "", //必填
4     val meetingDomain: String = "", //必填
5     val meetingCode: String = "", //必填
6     val meetingUUID: String = "", //必填
7     val memberUUID: String = "", //必填
8 )

```

3.3 构建入会配置

开发者通过 AliMeetingJoinConfig.Builder 构建入会信息，示例：

- Java 代码

```

1 public void joinMeeting(Context context,
2                         String meetingCode,
3                         String userId,
4                         String userName,
5                         MeetingInfo meetingInfo,
6                         String appTag,

```

```

7             int appNotifyRes,
8             boolean openBeautifyDefault,
9             boolean closeCameraDefault,
10            boolean muteAudioDefault,
11            AMUIMeetingDetail detail) {
12
13        AMUIMeetingJoinConfig.Builder builder = new AMUIMeetingJoinConfig.Builder()
14            .setMeetingCode(meetingCode)
15            .setUserId(userId)
16            .setUserName(userName)
17            .setMeetingDetailConfig(
18                new AMUIMeetingDetailConfig(
19                    meetingInfo.clientAppId, meetingInfo.meetingToken,
20                    meetingInfo.meetingDomain, meetingInfo.meetingCode,
21                    meetingInfo.meetingUUID, meetingInfo.memberUUID,
22                    meetingInfo.slsInfo
23                )
24            )
25            .setOpenBeautifyDefault(openBeautifyDefault)
26            .setAppIdentifier(appTag)
27            .setAppVersion(getShortVersionName())
28            .setAppNotifyIconRes(appNotifyRes)
29            .setOpenCameraDefault(!closeCameraDefault)
30            .setMuteAudioDefault(muteAudioDefault)
31            .setMeetingDetail(detail);
32        AliMeetingUIManager.joinMeeting(context, builder.builder());
33    }

```

- Kotlin代码

```

1 val builder = AMUIMeetingJoinConfig.Builder()
2     .setMeetingCode(meetingInfo.meetingCode)
3     .setUserId(userId)
4     .setUserName(userName)

```

```

5         .setMeetingDetailConfig(
6             AMUIMeetingDetailConfig(
7                 meetingInfo.clientAppId, meetingInfo.meetingT
oken,
8                 meetingInfo.meetingDomain, meetingInfo.meetin
gCode,
9                 meetingInfo.meetingUUID, meetingInfo.memberUU
ID
10            )
11        )
12        .setOpenBeautifyDefault(openBeautifyDefault)
13        .setAppIdentifier(appTag)
14        .setAppVersion(Platform.getShortVersionName())
15        .setAppNotifyIconRes(appRes)
16        .setOpenCameraDefault(!closeCameraDefault)
17        .setMuteAudioDefault(muteAudioDefault)
18        .setMeetingDetail(detail)

```

具体配置说明：

3.3.1 必须配置

配置变量	最低支持版本	说明
appNameRes	1.0.1.6	应用名字符串资源，状态栏消息会显示
appNotifyIconRes	1.0.1.6	应用图标，状态栏消息显示
userId	1.0.0.0	用户Id
meetingCode	1.0.0.0	会议口令
meetingDetailConfig	1.0.0.0	入会配置信息，包含入会域、Token、sls等信息

3.3.2 可选配置

配置变量	最低支持版本	说明
openCameraDefault	1.0.0.0	是否默认开启本地摄像头，默认为开启

muteAudioDefault	1.0.0.0	是否默认关闭麦克风，默认为不关闭
openSpeakerDefault	1.0.0.0	是否默认开启扬声器，默认为开启
openBeautifyDefault	1.0.1.6	是否默认开启美颜，默认为开启
enableVideo	1.0.0.0	当前会议是否支持视频，默认为支持，设置为false则变成音频会议
enableSaveTrafficMode	1.1.0.0	入会时是否默认开启省流量模式，如果开启，则openCameraDefault 参数不生效，会自动关闭摄像头。默认不开启省流量模式
loadingFinishShowDelayMills	1.0.1.8	进入会议时，loading 页面上面的结束会议按钮显示延时时间，默认为0，小于或等于0时不延迟显示
canHostUnMuteClientDirectly	1.1.0.0	主持人是否可以直接取消他人静音状态，默认为false
onlyHostCanMuteAudio	1.0.0.0	是否只允许发起人静音他人。 在非主持人会议模式下生效 ，默认为 false 即所有人都可以进行静音他人操作；设置为 true 则表示只有发起人可以进行静音他人操作。
onlyHostCanHangUpClient	1.0.0.0	是否只允许发起人挂断他人。 仅在非主持人模式下生效 ，默认为 false 即表示所有人都可以进行挂断他人操作；设置为 true 则只有发起人可以进行静音他人操作
meetingDetail	1.0.0.0	会议详情，在会中显示会议信息，邀请人入会时使用
meetingCallBack	1.0.0.0	会议关键事件回调，包括用户主动离会、会议结束，用户状态改变等事件，请参考3.3.2.2
actionPlugins	1.1.0.0	更多操作里面的扩展功能，可以添加自己的实现，如果要替换默认的扩展项，可以配合 disableDefaultActionPlugin 函数使用
feedBackUrl	1.1.0.0	在更多弹框中，是否显示用户反馈入口，如果 feedBackUrl 不为空，则显示反馈入口
enableManualDisableRateMeeting	1.1.0.0	是否允许手动关闭弹出评分页面，默认为True，当meetingDetail#rateLink 传空时，该参数无效 用户修改配置后，会回调到IAMUIMeetingUIController#onRateMeetingConfigChanged
allowsParticipantA	1.2.0.1	主持人会议中，在主持人未入会设置了不允许开启摄像头或者麦

3.3.2.1 日志输出配置

UISDK默认仅在Debug包中输出日志到Console，开发者可以自定义会中日志输入，在发布包中将日志存储到文件，方便后续问题排查。用户可以定义日志输出方式，如果线上需要收集日志，用户可以设置 `AMUILoggerLevel` 为 `AMUILoggerLevel.INFO`，防止日志过大

```
1 AMUIMeetingJoinConfig.Builder#setLoggerLevel(AMUILoggerLevel.VERBOSE, SDKDemoLogger())
```

SDKDemoLogger 的定义

```
1 private class SDKDemoLogger : AMUILoggerPrinter {
2     override fun printInfo(tag: String, msg: String) {
3         //开发者自定义日志输出，可以输出到文件保存
4         Log.d(tag, msg)
5     }
6
7     override fun printDebug(tag: String, msg: String) {
8         //开发者自定义日志输出，可以输出到文件保存
9         Log.d(tag, msg)
10    }
11
12    override fun printWarn(tag: String, msg: String) {
13        //开发者自定义日志输出，可以输出到文件保存
14        Log.w(tag, msg)
15    }
16
17    override fun printError(tag: String, msg: String, throwable:
18        Throwable?) {
19        //开发者自定义日志输出，可以输出到文件保存
20        Log.e(tag, msg, throwable)
21    }
22
23    override fun onMeetingKeyEvent(
24        progress: String, isStart: Boolean, time: Long, result: Boolean, extra:String?
```

```

24     ) {
25         //入会过程关键步骤输出
26         Log.e("DemoMeeting", "onMeetingKeyEvent $progress isStar
            t:$isStart time:$time $result")
27     }
28 }

```

3.3.2.2 会议关键回调配置

UISDK提供了会议关键状态回调供开发者进行监控，主要包括两个方面

- 入会关键流程回调

通过实现 3.3.2.1 节的 `AMUILoggerPrinter#onMeetingKeyEvent` 接口进行监听

- 人员状态、会议状态关键流程回调

通过 `AMUIMeetingJoinConfig.Builder#setMeetingCallBack(callBack: AMUIMeetingCallBack)` 进行监听

```

1  abstract class AMUIMeetingCallBack {
2
3      /**
4       * 加入会议成功之后，用户状态变化
5       */
6      open fun onClientStatusChanged(
7          client: AMSDKMeetingClient, event: AMUIClientStatusEv
8          ent
9      ) {
10
11      /**
12       * 用户主动离开会议，会同时触发 onMeetingFinished 回调
13       */
14      open fun onUserLeaveMeeting() {
15      }
16
17      /**
18       * 用户主动结束会议，会同时触发 onMeetingFinished 回调
19       */
20      open fun onUserFinishMeeting() {

```

```

21     }
22
23     /**
24      * 加入会议成功
25      */
26     abstract fun onMeetingJoined()
27
28     /**
29      * 加入会议成功后，会议中断或正常挂断
30      */
31     abstract fun onMeetingFinished(code: AMUIFinishCode, msg: String?)
32
33     /**
34      * 加入会议未成功之前，加入异常
35      */
36     abstract fun onJoinMeetingError(code: AMUIErrorCode, msg: String?)
37 }

```

3.4 加入会议

调用AliMeetingUIManager#joinMeeting(context: Context, config: AMUIMeetingJoinConfig) 即可入会

4 ReleaseNote

1.2.0.1

【新增】暴露 allowsParticipantActionWithoutHost 参数，支持用户控制主持人未入会时，其他参会人是否可以根据权限操作音视频开关

【新增】UISDK 在接收共享流时，在全屏状态下支持手势缩放

【修改】大方数会议持续性能优化

【修改】Toast 显示优化，离会后立即停止，后续将继续优化

1.1.0.8

- 【新增】大方会支持批量订阅
- 【修改】针对服务出现404/405错误时内部重新入会
- 【修改】内部日志优化
- 【增加】输出的SDK包中，增加meetingsdk-api的source.jar，仅用于调试

1.1.0.7

- 【修改】优化被挂断是内部数据清理逻辑

1.1.0.6

- 【修改】大方会议持续优化
- 【修改】不支持SLS日志对外暴露
- 【增加】输出的SDK包中，UISDK增加t28/t29两个文件，用户可根据不同的targetVersion选择对应的版本进行集成

1.1.0.5

- 【增加】支持targetVersion=29版本
- 【增加】对外暴露SLS日志接口
- 【修改】未入会成员呼叫功能实现修改，具体实现方式参考 3.1.2.3 节
- 【修改】去掉入会时 enableFloatingOutSideApp 配置，默认不在应用外展示，如果需要在应用外显示悬浮框，请参考 3.1.2.3 节

1.1.0.4

- 【修改】支持MCU会议

1.1.0.0

- 【修改】重构UISDK，重构UI交互逻辑
- 【增加】增加会中共享功能
- 【增加】增加会议评价，我要反馈相关内容
- 【修改】悬浮窗对MCU会议进行适配
- 【修改】统一资源命名方式

升级注意点：

替换资源名称需要更新，具体参考 3.1.2.2 节静态资源替换相关内容

1.0.0.6

- 【修改】入会时增加了sIsInfo信息，通过如下代码设置

```
1 AMUIMeetingJoinConfig.Builder#setMeetingDetailConfig(detailConfig:
    AMUIMeetingDetailConfig)
```

【修改】入会时增加 enableFloatingOutSideApp 参数，由于部分手机无法处理后台打开页面的权限，默认不支持在应用外显示会议悬浮框，如果需要在应用外显示悬浮框，可以通过设置为true控制悬浮框在应用外显示

【修改】补充完善会议日志

【新增】补充未入会成员在参会人列表中的显示，用于可以通过设置呼叫回调控制呼叫逻辑，具体使用参考 3.1.2.3 节

5 常见问题FAQ

1. SDK 对 Android 各种环境支持补充说明

- 暂不支持Android 暗黑模式
- 原则上不支持Android 5.0 以下版本

1. 如何对入会成功率进行监听

参考 3.3.2.2 节，通过设置 AMUIMeetingJoinConfig.Builder#setMeetingCallBack(callback: AMUIMeetingCallBack) 进行监控，对 onMeetingFinished 和 onJoinMeetingError 来统计

```
1 class MeetingCallBackDemo : AMUIMeetingCallBack() {
2
3     override fun onUserLeaveMeeting() {
4         //记录用户手动结束会议，会同步回调onMeetingFinished
5         Log.d("MeetingCallBackDemo", "onUserLeaveMeeting")
6     }
7
8     override fun onUserFinishMeeting() {
9         //记录用户手动结束会议，会同步回调onMeetingFinished
10        Log.d("MeetingCallBackDemo", "onUserFinishMeeting")
11    }
12
13    override fun onMeetingJoined() {
14        //用户入会成功
```

```

15     }
16
17     override fun onMeetingFinished(code: AMUIFinishCode, msg: String?) {
18         //会议结束
19         when (code) {
20             AMUIFinishCode.HANGUP_BY_SELF, AMUIFinishCode.NORMAL_
QUIT -> {
21                 //正常退会,
22             }
23             else -> {
24                 //非正常离会, 记录错误信息
25             }
26         }
27     }
28
29     override fun onJoinMeetingError(code: AMUIErrorCode, msg: String?) {
30         //入会失败, 记录错误信息
31         Log.d("MeetingCallBackDemo", "onJoinMeetingError $code $msg")
32     }
33 }

```

2. 遇到crash如何进行反馈

请提供：

- crash复现操作路径说明
- 出现crash的机器信息，包括机器厂商、机器型号、Android系统版本
- 集成环境的CompileSdkVersion、TargetSdkVersion、BuildToolsVersion
- crash日志

3、遇到会议问题如何排查

请提供：

- 问题复现操作路径说明
- 会议的 meetingCode，入会时传入的 meetingAppId（一个接入方会固定一个，不会随会议改变的），问题产生时间

- 集成环境的CompileSdkVersion、TargetSdkVersion、BuildToolsVersion

附录1: UISDK 关键接口补充说明

AMUIMeetingDetail 会议详情

```
1 public class AMUIMeetingDetail implements Serializable {
2     /**
3      * 会议标题，用于默认邀请界面显示邀请信息
4      */
5     public String subject;
6     /**
7      * 会议密码，可以为空，用于默认邀请界面显示邀请信息
8      */
9     public String password;
10    /**
11     * 分享信息，用于默认邀请界面分享链接
12     */
13    public String shareLink;
14    /**
15     * 会议开始时间，UTC时间，单位为毫秒
16     */
17    public long beginDate = 0;
18    /**
19     * 会议结束时间，UTC时间，单位为毫秒
20     */
21    public long endDate = 0;
22    /**
23     * 会议分享信息，用于默认邀请界面分享内容发送
24     */
25    public String shareMessage;
26 }
```

悬浮框相关

AMUIEdgeInsets 悬浮框显示约束

```
1 /**
2  * 悬浮窗显示区域
3  */
4 class AMUIEdgeInsets(
5     /**
6      * 距离屏幕 左侧 边缘距离
7      */
8     val left: Int,
9     /**
10     * 距离屏幕 上侧 边缘距离
11     */
12     val top: Int,
13     /**
14     * 距离屏幕 右侧 边缘距离
15     */
16     val right: Int,
17     /**
18     * 距离屏幕 底部 边缘距离
19     */
20     val bottom: Int
21 )
```

AMUIFloatingShowConfig 悬浮框显示大小/位置定义

```
1 /**
2  * 悬浮框显示参数
3  */
4 class AMUIFloatingShowConfig(
5     /**
6      * 悬浮框宽度
7      */
8     val width: Int,
9     /**
10     * 悬浮框高度
11     */
```

```

12     val height: Int,
13     /**
14      * 悬浮框圆角
15      */
16     val radius: Float,
17     /**
18      * 悬浮框默认显示的起始位置 x
19      */
20     val startX: Int,
21     /**
22      * 悬浮框默认显示的起始位置 y
23      */
24     val startY: Int
25 )

```

IAMUIFloatingLayout 悬浮框UI接口

```

1
2 interface IAMUIFloatingLayout {
3
4     /**
5      * 悬浮框的 rootView
6      */
7     fun contentView(): View
8     /**
9      * 初始化悬浮框显示
10      */
11     fun initLayout(floatingSize: AMUIFloatingShowConfig)
12
13     /**
14      * 显示某个client，只在初始化显示时会调用
15      */
16     fun showClient(client: AMSDKMeetingClient, eglBase: AMSDKEglBase)
17
18     /**
19      * 主讲人发生变化
20      */

```

```

21     fun onMainSpeakerChanged(client: AMSDKMeetingClient, eglBase:
    AMSDKEglBase, onlineCounts: Int)
22
23     /**
24      * 用户状态发生变化
25      */
26     fun onClientStatusChanged(client: AMSDKMeetingClient, event:
    AMSDKStatusEvent, onlineCounts: Int)
27
28     /**
29      * 销毁悬浮框
30      */
31     fun release()
32 }

```

人员列表相关

IAMUIMemberListController 人员列表控制

```

1
2 /**
3  * 人员列表基于 RecyclerView 设计
4  */
5 abstract class IAMUIMemberListController {
6
7     companion object {
8         //ViewHolder 类型
9         const val VIEW_TYPE_CLIENT_ITEM = 1
10        const val VIEW_TYPE_BAR_ONLINE = 2
11        const val VIEW_TYPE_BAR_OFFLINE = 3
12    }
13
14    var clientClickListener: IAMUIClientClickListener? = null
15
16    /**
17     * 在没有未入会成员时，是否隐藏未入会成员的 Bar
18     */
19    open fun hideOfflineBarIfNoClient(): Boolean = true

```

```

20
21     /**
22      * 未入会成员呼叫监听器
23      */
24     open fun getOfflineCallActionListener(): IAMUIClientCallListe
ner? = null
25
26
27     /**
28      * 当前入会人是否有主持人权限
29      */
30     fun hasHostPermission() = AMUIMeetingConfigHelper.hasHostPerm
ission()
31
32     /**
33      * 当前会议是否已开启主持人模式
34      */
35     fun isHostEnabled() = AMUIMeetingConfigHelper.isHostEnabled()
36
37     /**
38      * 获取除了正常人员的其他类型，获取当前需要显示的类型，如已入会未入会的标题
栏，可以自定义 viewType，
39      * 后续在 getViewHolder 中可以根据 viewType 进行初始化，默认仅支持 V
IEW_TYPE_CLIENT_ITEM，
40      * VIEW_TYPE_BAR_ONLINE, VIEW_TYPE_BAR_OFFLINE 三种 viewType
41      * */
42     open fun getBarItem(): List<AMUIListItem>
43
44     /**
45      * 当出现 AMSDKStatusEvent 事件时，是否需要更新对应的 client 显示状
态，更具 View 显示内容判断
46      */
47     open fun needUpdateItemStatus(event: AMSDKStatusEvent): Boole
an
48
49     /**
50      * 对应 RecyclerView onCreateViewHolder
51      */
52     open fun getViewHolder(
53         inflater: LayoutInflater, parent: ViewGroup, viewType: In

```



```

t
54     ): RecyclerView.ViewHolder
55
56     /**
57      * 对应 RecyclerView onBindViewHolder
58      */
59     open fun bindBarItem(
60         context: Context, viewHolder: RecyclerView.ViewHolder,
61         viewType: Int, totalCount: Int, onlineCounts: Int, offlineCounts: Int
62     )
63
64     /**
65      * 对应 RecyclerView onBindViewHolder, 内部根据 client 是否在线,
66      * 再调用 bindOnlineClient 或 bindOfflineClient
67      */
68     open fun bindClientItem(
69         context: Context, viewHolder: RecyclerView.ViewHolder, client: AMSDKMeetingClient
70     )
71
72     /**
73      * 对应 RecyclerView onBindViewHolder, listener 为默认的 clickListener,
74      * 未入会成员默认无响应, 如果需要处理, 请自己替换listener, 不要使用默认的listener
75      * @param showOfflineAction 离线成员是否显示右侧操作入口
76      */
77     open fun bindOfflineClient(
78         context: Context, viewHolder: RecyclerView.ViewHolder,
79         client: AMSDKMeetingClient, showOfflineAction: Boolean, listener: IAMUIClientClickListener?
80     )
81
82     /**
83      * 对应 RecyclerView onBindViewHolder, listener 为默认的 clickListener,
84      * 如果需要处理, 请自己替换listener, 不要使用默认的listener
85      */
86     open fun bindOnlineClient(

```

```

87         context: Context, viewHolder: RecyclerView.ViewHolder,
88         client: AMSDKMeetingClient, listener: IAMUIClientClickListener?
89     )
90
91     /**
92     * 获取列表内容比较器，用来排序人员列表显示，默认为先显示已入会成员，再显示
    未入会成员
93     */
94     open fun getItemSortComparator(): Comparator<IAMUIListItem>
95
96     interface IAMUIClientCallListener {
97         fun callClient(view: View, client: AMSDKMeetingClient?)
98     }
99 }

```