

Android SDK 开发 FAQ 文档

目录

1	天翼 RTC 平台能力集成常见问题:	4
1.1	Q: 天翼 RTC 平台的定位是什么? 能够为用户提供哪些应用服务?	4
1.2	Q: 集成天翼 RTC 能力并发布应用, 需要什么前提条件?	4
1.3	Q: 开发者账号主要包括哪些内容?	4
1.4	Q: 如何成为开发者?	4
1.5	Q: 天翼 RTC 平台能力的开发流程是什么样的?	4
1.6	Q: 能够集成天翼 RTC 平台能力的客户端有哪些?	5
1.7	Q: 第三方应用服务器的具体功能有哪些:	5
1.8	Q: RTC 平台具有哪些功能?	5
1.9	Q: 客户端和应用服务器之间有哪些交互?	5
1.10	Q: 应用服务器和 RTC 平台之间有哪些交互?	6
1.11	Q: 客户端与 RTC 平台之间的交互有哪些?	6
2	天翼 RTC 平台鉴权及能力 Token 常见问题:	6
2.1	Q: 用户在天翼 RTC 平台的鉴权过程是怎么样的?	6
2.2	Q: 从天翼 RTC 平台获取的 Token 接口结构是什么样的? 请求方法是什么? ..	7
2.3	Q: 获取 Token 接口的 URL 格式是什么样的?	7
2.4	Q: 获取 Token 接口的请求消息体的请求消息参数有哪些? 响应消息体的响应消息参数有哪些?	7
2.5	Q: 在什么情况下应用需要向 RTC 平台重新获取 Token?	8
2.6	Q: 天翼 VV demo 是如何获取 Token 的?	9
3	天翼 RTC SDK 能力、接口及配置常见问题:	9
3.1	Q: 天翼 RTC SDK 具有哪些能力?	9
3.2	Q: SDK 能力的语音、视频通信采用了哪些编码技术? 音视频质量如何?	9
3.3	Q: 天翼 RTC SDK 能力的库文件包括哪些?	9
3.4	Q: Android SDK 的接口有哪些?	10
3.5	Q: SDK 接口之间存在什么样的依赖关系?	10
3.6	Q: 如何导入 SDK 所需的 jar 包?	11

3.7	Q: 如何配置 AndroidManifest.xml?	11
3.8	Q: 在创建 Device 时, 应用需要提供什么?	12
4	初始化常见问题:	12
4.1	Q: Android SDK 初始化流程是什么样的?	12
4.2	Q: SDK 初始化失败的情况有哪些? 应该如何处理?	12
4.3	Q: 如何设置视频清晰度? 可选的视频清晰度有哪些?	12
4.4	Q: 如何设置音视频编解码器? 可选的音视频编解码方案有哪些?	13
4.5	Q: 当通话双方选择的视频清晰度以及音视频编码方案不同时, SDK 如何处置?	13
4.6	Q: 如何设置视频日志文件路径?	13
4.7	Q: Android 系统时间不准确, 会影响 SDK 初始化么?	13
5	注册登录常见问题:	13
5.1	Q: 注册 Device 时, 需要发送给服务器的参数有什么规范要求?	13
5.2	Q: 注册 Device 时, kAccType 共有几种可选的终端类型?	14
5.3	Q: 注册 Device 时, 可选的账号体系共有几种?	14
5.4	Q: 注册 Device 时, 第三方应用的用户名有什么命名规范?	14
5.5	Q: 如何查询 Device 的注册结果及状态切换?	14
5.6	Q: 在认证登录中出现错误, 应该如何处理?	15
6	建立通话常见问题:	15
6.1	Q: 创建呼叫时, 需要向服务器发送的参数有什么规范要求?	15
6.2	Q: 创建呼叫时, 账号的格式是什么样的?	16
6.3	Q: 创建呼叫时的呼叫媒体类型有哪些?	16
6.4	Q: 创建呼叫时的可选项 kCallInfo 是什么?	16
6.5	Q: 是否支持同时创建两路以上的通话?	16
6.6	Q: 创建呼叫后, 呼叫会遇到哪些异常情况, 应该如何处理?	16
6.7	Q: 收到来电通知时, 需要注意什么?	17
6.8	Q: 被叫接听呼叫时, 如何设置接听类型?	17
6.9	Q: 如何获取呼叫信息?	17
6.10	Q: 挂断已有连接, 是否需要等待 SDK 释放?	18
7	通话过程中的常见问题:	18

7.1	Q: 通话建立后, 如何获取通话时长?	18
7.2	Q: 通话过程中, 发现视频画面出现绿栅或者花屏现象, 应该怎么办?	18
7.3	Q: 如何切换前后摄像头?	18
7.4	Q: 视频窗口的界面从隐藏到显示, 是否需要重置视频窗口?	18
7.5	Q: 视频通话中, 截屏功能保存的图片路径在哪里设置?	18
7.6	Q: 如何设置摄像头旋转?	19
7.7	Q: SDK 是否支持视频录制功能?	19
7.8	Q: 通话过程中可能出现的错误代码有哪些? 分别代表什么意思?	19
7.9	Q: 通话过程中能否接听 SIM 卡来电或者拨打传统电话?	19
8	常见的网络问题:	19
8.1	Q: SDK 初始化时发生网络异常, 应该如何处理?	19
8.2	Q: 网络状态发生变化时, 如何及时获取 Device 的状态变化?	20
8.3	Q: 创建呼叫时, 因为网络问题导致的呼叫连接异常的情况有哪些? 应该如何处理?	20
8.4	Q: 针对网络突然中断的情况, SDK 是如何处理的?	20
8.5	Q: 网络发生异常时, 应用是否需要调用 SDK 接口连接 RTC 平台?	20
8.6	Q: 开发者应该如何处理常见的呼叫失败?	20
8.7	Q: 手机锁屏是否会导致 SDK 通信断开?	21
9	终端与浏览器互通的常见问题:	21
9.1	Q: 如何实现手机终端与浏览器互通?	21
10	终端适配常见问题:	21
10.1	Q: 回声问题的适配是在 SDK 中完成的么?	21
10.2	Q: 免提和听筒切换、音量调节的适配是在 SDK 中完成的么?	21
11	其他常见问题:	21
11.1	Q: 别的应用可以使用和天翼 VV 相同的广播代码么?	21
11.2	Q: 如果采用 WAP 接入网络, 该如何进行处理?	22

1 天翼 RTC 平台能力集成常见问题：

1.1 Q: 天翼 RTC 平台的定位是什么？能够为用户提供哪些应用服务？

A: 天翼 RTC 平台运行在公共互联网上，定位于纯互联网方式提供 OTT 实时通信服务的通信能力平台，未来将与中国电信的 IMS/软交换/PSTN 等网络通信系统互通。为客户端应用提供应用内、应用间的语音、视频和数据实时通信沟通(初期主要提供语音和视频沟通)。

1.2 Q: 集成天翼 RTC 能力并发布应用，需要什么前提条件？

A: 开发者要在应用中集成 RTC 能力并发布应用，前提条件是要拥有开发者账号，也就是要拥有 AppKey、ApplicationID。目前开发者需要将注册信息提交至 RTC 平台进行注册，平台自动为开发者生成相关 ID，和开发者所拥有能力相匹配的能力 Token。

1.3 Q: 开发者账号主要包括哪些内容？

A: 开发者账号主要包括 AppKey、ApplicationID 以及 CapbilityToken。其中：

- AppKey 是开发者提交成功注册信息后，系统为开发者账号生成的密钥。AppKey 很重要，不能轻易暴露；
- ApplicationID 是开发者创建的应用后，系统为所创建应用生成的平台唯一 ID，该 ID 在 SDK 配置时使用；
- CapbilityToken（能力 Token）用来标识开发者所拥有的 RTC 能力，用以能力鉴权。只有当客户端所调用能力鉴权通过时，客户端才能完全使用该能力。

1.4 Q: 如何成为开发者？

A: 在 RTC 开放者门户或者集成 RTC 能力的开放平台门户上注册开发者资料，提交并接受平台审核，审核通过将成为开发者并获得开发者账号，系统将自动分配 AppKey 给开发者。

1.5 Q: 天翼 RTC 平台能力的开发流程是什么样的？

A: RTC 平台基于 IP 的实时通信能力，通过开放 API 以及 SDK 的方式提供给开发者和商家，协助开发者快速、高效、低成本打造集成通信沟通能力的业务应用产品。具体的开发流程：

- 1) 客户端登录，向应用服务器请求账号信息；
- 2) 第三方应用服务器向 RTC 平台请求能力 Token，第三方服务器和 RTC 平台各存一份；
- 3) 客户端携带账号信息、能力 Token 向 RTC 平台发起注册、呼叫。

1.6 Q: 能够集成天翼 RTC 平台能力的客户端有哪些？

A: 天翼 RTC 能力能够提供跨平台的通信能力服务，适用于 Android 手机客户端程序、IOS 手机客户端程序、PC 客户端程序以及浏览器，并且能够实现不同客户端之间的通信服务。客户端只需集成 SDK 或使用 REST API，编写很少的代码，便可轻松完成对 RTC 通信能力的集成。

1.7 Q: 第三方应用服务器的具体功能有哪些：

A: 第三方应用服务器是指开发者的业务服务器，常用于提供业务应用逻辑服务以及控制自有的业务和用户数据，如：用户注册、用户登录、业务展示等。第三方应用服务器与 RTC 平台交互，通过调用 RTC 平台的 REST 接口获取能力 Token 和进行 RTC 业务的统计查询。

1.8 Q: RTC 平台具有哪些功能？

A: RTC 平台为开发者提供了开放接口和 SDK，并面向不同平台提供不同语言的 Demo，方便开发者顺利上手，快速集成语音、视频通信能力。

1.9 Q: 客户端和应用服务器之间有哪些交互？

A: 客户端到应用服务器，要完成应用内部的注册、登录等内部业务，SDK 不做任何参与。在这一过程中，有很重要的一个环节是客户端向应用服务器请求能力 Token，客户端调用应用服务器的私有接口，请求能力 Token，并保存在内存中，在初始化 SDK 的时候使用。

1.10 Q: 应用服务器和 RTC 平台之间有哪些交互?

A: 在开发者账号生成初期, 应用服务器调用 REST 接口 (获取能力令牌) 向 RTC 平台请求一次能力 Token, 并保存下来供客户端获取。在以后的呼叫请求中, 客户端会携带功能 Token, RTC 平台检查该 Token 是否有效 (是否与平台的一致), 只有当 Token 有效时, 客户端的呼叫请求才会被转发。

1.11 Q: 客户端与 RTC 平台之间的交互有哪些?

A: 客户端与 RTC 平台间的交互主要是 RTC 能力调用, 比如呼叫、接听、群组业务等等。RTC 能力调用分两种情况, 本地化 SDK 和 REST API。以 SDK 方式调用时, 客户端发起的每次呼叫中都会携带上能力 Token, 供 RTC 平台检查。

2 天翼 RTC 平台鉴权及能力 Token 常见问题:

2.1 Q: 用户在天翼 RTC 平台的鉴权过程是怎么样的?

A: 用户在天翼 RTC 平台的鉴权过程:

- 1) 当客户端用户请求登录 RTC 平台时, 第三方应用客户端首先向第三方应用平台申请鉴权使用的 RTC 平台的用户名和 token;
- 2) 首先第三方客户端在第三方应用平台完成身份鉴权;
- 3) 第三方应用平台需要在天翼 RTC 平台完成鉴权 (ApplicationID, AppKey);
- 4) 第三方应用平台先向 RTC 平台获取鉴权使用的 RTC 平台的用户名和 Token, 请求中需要携带该用户在第三方应用的用户 ID 和需要的能力集;
- 5) RTC 平台返回对应的用户名和 token, 当前 RTC 平台的用户名和第三方平台的用户名;
- 6) 第三方应用平台将用户名和 token 分发给第三方应用客户端;
- 7) 用户通过第三方客户端向 RTC 平台发起鉴权请求 (携带用户名和 Token);
- 8) RTC 平台检查用户名和之前颁发的 token, 为用户进行鉴权。

2.2 Q: 从天翼 RTC 平台获取的 Token 接口结构是什么样的？请求方法是什么？

A: 获取 Token 接口主要包括 URL、认证头和消息体。URL 可以是 HTTP 或者 HTTPS，认证头包括 AppID 及 AppKey，消息体包含了用户账户、设备 ID 等信息。获取 Token 采用的请求方法是 POST。

2.3 Q: 获取 Token 接口的 URL 格式是什么样的？

A: 接口的 URL 格式为：

https://{serverRoot}/{APIversion}/ApplicationID/{InApplicationID}/CapabilityToken。其中大括弧中为需要填充的对应的值，字体颜色与以下值对应：

http://rest.chinartc.com:8090/RTC/ws/1.0/ApplicationID/123/CapabilityToken

https://rest.chinartc.com:442/RTC/ws/1.0/ApplicationID/123/CapabilityToken

2.4 Q: 获取 Token 接口的请求消息体的请求消息参数有哪些？响应消息体的响应消息参数有哪些？

A: Token 接口的请求消息参数包括：

参数名	是否 必选	参数类型 【长度】	含义描述
authType	是	Integer	授权类型： 0：向应用整体授权，所有该应用用户获得相同的权限； 1：向应用内用户授权
appAccountID	是	字符串【256】	当授权类型是应用内用户授权时，为第三方应用的用户 帐号，注意加账号体系前缀如 10-
userTerminalType	否	String	获取令牌的终端类型，取值： TV、Pad、Phone、PC、Browser、.....
userTerminalSN	否	String	获取令牌的终端特征序列号，唯一标识某个终端（如 IMEI，MAC 地址等）

grantedCapabiltID	是	String	标识被授权的能力，RTC 平台为每个能力分配不同的能力 ID，初期对于每个能力的调用权限不再细分，二期等后续参考 OAuth 的能力 Scope 细化定义。 100: 点对点语音; 200: 点对点视频; 300: 多人会话(301 是多人会话对讲, 302 是语音群聊, 303 是多人两方); 400: 微直播; 可以被授予多个能力，每个能力 ID 之间以 “<”号区分，如授予“100<200<301”则赋予三个能力.
callbackURL	否	String	用户状态更新通知回调地址

Token 接口的响应消息参数包括:

参数名	是否必选	参数类型	含义描述
requestid	是	String	请求的对应标识
rtcaccountid	否	String	RTC 分配的用户账号 为空则第三方应用按照账号对应规则产生 RTC 账号。
capabilitytoken	否	String	能力令牌（包含密码）
CurrentUserTerminalType	否	String	当前用户已在线状态时的帐号终端类型 取值： TV、Pad、Phone、PC、Browser、..... 若用户帐号未在线，则无返回
CurrentUserTerminalSN	否	String	当前用户已在线状态时的帐号终端令牌特征序列号，唯一标识某个终端（如 IMEI，MAC 地址等） 若用户帐号未在线，则无返回
code	是	Integer	接口调用结果返回码，HTTP 状态码的补充与扩展结果返回码
reason	否	String	支持失败原因描述

2.5 Q: 在什么情况下应用需要向 RTC 平台重新获取 Token?

A: 当 Device 监听接口收到 CallCode_Forbidden、ReLoginNetwork、DeviceEvt_KickedOff 时，可以由用户选择是否继续（应用需要避免自动选择而进入死循环），想继续使用 CallCode_Forbidden 和 DeviceEvt_KickedOff 就需要重新获取 token。当重新获取 token 成功后，跟 ReLoginNetwork 一

样重新登录，应用需要先 `release` 然后再 `createDevice`。其他错误 SDK 内部会不同周期的自动尝试重连，不需要重新获取 `token` 重新登录。状态异常时应用需要在用户呼叫前给出异常提示，避免异常状态使用 `connect` 接口。

2.6 Q: 天翼 VV demo 是如何获取 Token 的？

A: 当前 VV demo 自带了获取 `token` 的功能，并对 `token` 进行了缓存，当没有 `token` 或者失效后才会再次更新缓存。

3 天翼 RTC SDK 能力、接口及配置常见问题：

3.1 Q: 天翼 RTC SDK 具有哪些能力？

A: SDK 是 RTC 平台提供给手机端的 Native Apps 开发者的快速集成 RTC 能力的一种便捷开发工具集，SDK 提供了纯 IP 的点对点语音、点对点视频通信、多人语音聊天室（包括多人会话群聊、多人会话对讲、多人两方语音）、微直播通信能力，旨在为第三方开发者在应用内快速、高效、低成本集成语音视频实时通信沟通提供了一站式的服务。

3.2 Q: SDK 能力的语音、视频通信采用了哪些编码技术？音视频质量如何？

A: 天翼 RTC 的语音支持 `opus\ilbc` 三种编码、视频支持 `vp8\h.264` 编码，可以由业务应用层通过接口设置具体的编码。RTC 的语音、视频质量通过研发优化和现网检验，已经达到业内一流水准，满足众多应用通过互联网实时沟通的需求，并且作为电信运营商的互联网通信能力产品，后续将结合自身的智能管道提供网络带宽保障，提供更高品质的音视频质量，打造 Telco-OTT 融合服务。

3.3 Q: 天翼 RTC SDK 能力的库文件包括哪些？

A: SDK 库文件包括：`media-native.jar`、`rtcsdk.jar`、`libapi-native.so`、`libmedia-native.so`。

3.4 Q: Android SDK 的接口有哪些？

A: Android SDK 的接口包名为 `rtc.sdk.iface`。在使用 SDK 各项功能之前，首先要实例化 SDK，获取 Client 的单例对象，并完成一些配置工作。Android SDK 的接口及其实现功能：

接口名	实现功能
RtcClient	SDK 实例
Device	授权，能力集成
Connection	连接管理
GroupMgr	会议管理
ClientListener	SDK 初始化状态回调接口
DeviceListener	设备注册状态回调接口
ConnectionListener	连接状态回调接口
GroupCallListener	会议群组的回调接口

3.5 Q: SDK 接口之间存在什么样的依赖关系？

A: SDK 接口之间的依赖关系：

- 1) 只有 SDK 初始化成功后才能创建 Device，而呼叫接口的使用依赖于 Device 的状态是否正常。当 SDK 销毁后，那么其他所有接口都无法使用了。
- 2) RtcClient 初始化成功后，创建 Device 却失败，不需要重新初始化。同样的，呼叫失败不需要重新创建 Device。
- 3) 进行呼叫前必须要检查 Device 的状态，如果是限制呼叫，则不需要调用 Connect 接口。
- 4) 由于当前 SDK 只支持一路呼叫，所以在上一个呼叫未释放前切忌进行第二次呼叫。

3.6 Q: 如何导入 SDK 所需的 jar 包?

A: 获取最新版 SDK 的 zip 包并解压, 然后

- 1) 将其中的 libs 文件夹合并到本地工程 libs 子目录下。Eclipse 用户右键工程根目录, 选择 Properties -> Java Build Path -> Libraries, 然后点击 Add External JARs... 选择指向 jar 的路径, 点击 OK, 即导入成功, 注意: Eclipse ADT 17 以上版本用户, 请在工程目录下建一个文件夹 libs, 把 jar 包直接拷贝到这个文件夹下, 再在 Eclipse 里面刷新一下工程就好了, 不要通过上述步骤手动添加 jar 包引用;
- 2) 复制 aidl 文件, 将 src 下的 rtc\sdk\aidl 目录全部复制到 Android 项目的 src 目录下即可, 注意: rtc 目录的 aidl 文件用于跨进程通信, 必须要复制到开发目录下, 开发者请勿修改, sdk 使用的编码为 utf-8;

3.7 Q: 如何配置 AndroidManifest.xml?

A: 本 SDK 所支持的最低 API level=9, 需要网络支持。打开 AndroidManifest.xml, 添加 SDK 需要的权限到<manifest>标签下:

- 1) android.permission.WRITE_EXTERNAL_STORAGE
- 2) android.permission.ACCESS_NETWORK_STATE
- 3) android.permission.ACCESS_WIFI_STATE
- 4) android.permission.INTERNET
- 5) android.permission.RECORD_AUDIO
- 6) android.permission.MODIFY_AUDIO_SETTINGS
- 7) android.permission.CAMERA
- 8) android.hardware.camera
- 9) android.hardware.camera.autofocus
- 10) android.permission.WAKE_LOCK

添加服务:

```
<service android:name="rtc.sdk.impl.SdkService" >
    <intent-filter>
        <action android:name="rtc.sdk.service" />
    </intent-filter>
```

3.8 Q: 在创建 Device 时，应用需要提供什么？

A: 在创建 Device 时，应用需要给出：应用 user、能力 Token、应用 ID (ApplicationID)。

4 初始化常见问题：

4.1 Q: Android SDK 初始化流程是什么样的？

A: 开发者调用 RtcClient 类中的接口来初始化 SDK，需要遵守如下初始化流程：

- 1) 调用 `public int initialize (Context inContext, ClientListener inListener)` 接口初始化 SDK，参数 `inContext` 是 Android 应用上下文对象，用于在 SDK 中调用上层应用资源；
- 2) 调用 `ClientListener` 类中的 `void onInit(int result)` 接口监听初始化结果。如果接口参数 `result=0`，说明 SDK 初始化成功；
- 3) SDK 初始化成功后，调用 `public void setAudioCodec(int codec)` 接口和 `public void setVideoCodec(int codec)` 分别设置优先选用的音频编解码和视频编解码方案；
- 4) 调用 `public void setVideoAttr(int wh)` 接口设置视频清晰度。

4.2 Q: SDK 初始化失败的情况有哪些？应该如何处理？

A: `ClientListener` 类中的 `void onInit(int result)` 接口的参数 `result` 不为 0 时表示 SDK 初始化失败。常见的错误如 `result=9003`，一般是因为终端网络不可用所导致的，应通知用户选择是否继续，继续则应先 `release RtcClient` 对象，重建 `RtcClient` 对象并调用 `initialize`。初始化成功后，应用退出销毁前，其他接口失败情况不需要再次初始化。

4.3 Q: 如何设置视频清晰度？可选的视频清晰度有哪些？

A: 开发者可以调用 `RtcClient` 类中的 `public void setVideoAttr(int wh)` 接口设置视频清晰度，其中参数 `wh` 的可选值有 `RtcConst.Video_SD`、

RtcConst.Video_FL、RtcConst.Video_HD，分别表示标清、流畅、高清。

4.4 Q: 如何设置音视频编解码器？可选的音视频编解码方案有哪些？

A: 开发者可以调用 RtcClient 类中的 `public void setAudioCodec(int codec)` 接口和 `public void setVideoCodec(int codec)` 分别设置音频编码方案和视频编码方案，其中音频编码参数 `codec` 可选的编码方案有 RtcConst.ACodec_OPUS、RtcConst.ACodec_ILBC；视频编码参数 `codec` 可选的编码方案有 RtcConst.VCodec_VP8、RtcConst.VCodec_H264。

4.5 Q: 当通话双方选择的视频清晰度以及音视频编码方案不同时，SDK 如何处置？

A: 当通话双方选择的视频清晰度不同时，实际生效的为较低的视频清晰度；当通话双方选得的音视频编码方案不同时，实际生效的为被叫所选择的方案。

4.6 Q: 如何设置视频日志文件路径？

A: 开发者可以调用 RtcClient 类中的 `public void enableLog(String sFile)` 接口设置日志文件。参数 `sFile` 存储日志文件的绝对路径，`null` 或者 `""` 表示取消日志。

4.7 Q: Android 系统时间不准确，会影响 SDK 初始化么？

A: Android 系统时间需要准确，否则时间异常会导致初始化过程中 HTTP 接口失败。

5 注册登录常见问题：

5.1 Q: 注册 Device 时，需要发送给服务器的参数有什么规范要求？

A: 注册 Device 时需要发送给平台的参数在 `public Device createDevice(String parameters, DeviceListener inListener)` 接口的 `parameters` 实现。参数 `parameters` 必填的信息包括：kAccType/kAccUser/kAccAppID/kAccPwd，可

以先调用 `SdkSettings.defaultDeviceSetting()` 获取默认设置。其中：

- 1) `kAccType`: 其值表示终端类型，要与从 RTC 平台获取 Token 时传递的终端类型值一致；
- 2) `kAccUser`: 其值表示第三方应用的用户名；
- 3) `kAccAppID`: 其值是应用的唯一标示，由 RTC 生成；
- 4) `kAccPwd`: 其值表示注册密码/能力令牌；
- 5) `kAccSrtp`: 可选参数，表示是否加密媒体流，默认值为 1，为 2 时可与浏览器互通。

5.2 Q: 注册 Device 时，`kAccType` 共有几种可选的终端类型？

A: `kAccType` 的值表示终端类型，主要分为：`UEType_Any`（任意终端）、`UEType_TV`（电视终端）、`UEType_Phone`（手机终端）、`UEType_PC`（电脑终端）、`UEType_Browser`（浏览器终端）、`UEType_Pad`（平板终端）、`UEType_Other`（其他终端）。默认终端类型（`UEType_Current`）为 `UEType_Phone`。

5.3 Q: 注册 Device 时，可选的账号体系共有几种？

A: SDK 共定义了 4 种账号体系：`UEAPPID_Self`（私有账号）、`UEAPPID_Tianyi`（天翼账号）、`UEAPPID_SinaWeiBo`（新浪微博账号）、`UEAPPID_QQ`（QQ 账号）。当前账号默认为私有账号（`UEAPPID_Current= UEAPPID_Self`）。这里需要注意的是账号体系 `RtcConst.UEAPPID_Current` 必须在获取 token 之前确定。

5.4 Q: 注册 Device 时，第三方应用的用户名有什么命名规范？

A: `kAccUser` 的值表示第三方应用的用户名，该值不可包含“~”、空格。

5.5 Q: 如何查询 Device 的注册结果及状态切换？

A: 在注册 Device 后，开发者可通过 `DeviceListener` 类中的回调接口 `public void onDeviceStateChanged(int state)` 监听 Device 的状态切换。其中接口参

数 state 为:

- 1) `CallCode_Success` 表示注册成功或注销成功, 需要进一步通过判断该 `Device` 是否为 `null` 来确定 state 状态为注册成功还是注销成功;
- 2) `RtcConst.NoNetwork` 表示网络不可用或服务器错误, 应限制呼叫。此时应销毁当前 `Device`;
- 3) `RtcConst.ChangeNetwork` 表示网络切换了, 可以继续呼叫;
- 4) `RtcConst.PoorNetwork` 表示网络闪断, 可以忽略, 不影响呼叫;
- 5) `RtcConst.ReLoginNetwork` 表示重连失败应用可以选择重新登录, 应限制呼叫;
- 6) `RtcConst.DeviceEvt_KickedOff` 表示同一账号在另一同类型终端登录, 应限制呼叫;
- 7) `RtcConst.DeviceEvt_MultiLogin` 表示同一账号在不同类型终端登录, 不影响呼叫, 不同类型终端的登录只需给出界面提示即可;
- 8) 除了上述 7 个状态外, 其他值均表示注册失败。

5.6 Q: 在认证登录中出现错误, 应该如何处理?

A: 在认证登录中出现错如, 应该进行如下处理:

- 1) 整个登录流程某一步出现问题时, 提示出错, 同时停止继续走下面的流程。
- 2) 首先要处理第三方在自己平台认证的错误。
- 3) 获取服务器信息错误。
- 4) 获取 token 失败。
- 5) rtc 平台认证失败。

以上每一步都有可能由网络原因造成, 如请求超时, 应提示用户网络不稳定。

6 建立通话常见问题:

6.1 Q: 创建呼叫时, 需要向服务器发送的参数有什么规范要求?

A : 开 发 者 调 用 `public Connection connect(String parameters,`

ConnectionListener inListener)接口创建一个新的连接呼叫,参数 parameters 里需要填写对方号码、呼叫类型等信息。其中必填项: kCallRemoteUri/ kCallType, 选填项为 kCallInfo。kCallRemoteUri 表示呼叫对端的用户, 包含了账号的各部分内容, 需要注意的是来电信息中的 kCallRemoteUri 则只有号码; kCallType 为呼叫媒体类型 (例如点对点语音、点对点视频)。

6.2 Q: 创建呼叫时, 账号的格式是什么样的?

A: 账号格式应该形如 “账号体系-号码~应用 ID~终端类型”。账号的各部分内容被存储在 Connection Keys 的 kCallRemoteUri 中。

6.3 Q: 创建呼叫时的呼叫媒体类型有哪些?

A: 呼叫媒体类型 kCallType 主要包括: CallType_Audio (纯音频)、CallType_Video (纯视频)、CallType_A_V (音频+视频)、CallType_A_V_M (音视频监控)、CallType_A_V_L (单点直播) 等。

6.4 Q: 创建呼叫时的可选项 kCallInfo 是什么?

A: kCallInfo 为呼叫业务信息, 携带应用层自定义的消息。这里需要注意的是 kCallInfo 内容不可包含逗号。

6.5 Q: 是否支持同时创建两路以上的通话?

A: SDK 目前暂不支持呼叫保持及两路以上通话, 所以有来电时不能再去呼叫, 呼叫时有新的来电要拒接。

6.6 Q: 创建呼叫后, 呼叫会遇到哪些异常情况, 应该如何处理?

A: 当呼叫中遇到异常, 将回调 onDisconnected(int reasoncode)接口, 并将异常状态 reasoncode 通知给上层。异常状态 (reasoncode) 主要包括:

- 1) RtcConst.CallCode_Timeout 请求超时 (暂时无法接通)
- 2) RtcConst.CallCode_Fail 对方不在线 (暂时无法接通)
- 3) RtcConst.CallCode_Forbidden 请求禁止 (暂时无法接通)

- 4) `RtcConst. CallCode_NotFound` 对方号码不存在（呼叫的号码不存在）
- 5) `RtcConst. CallCode_Network` 服务器错误或网络不可用（暂时无法接通）
- 6) `RtcConst. CallCode_Reject` 被叫拒接（对方正忙）
- 7) `RtcConst. CallCode_Busy` 用户忙（对方正忙）
- 8) `RtcConst. CallCode_Cancel` 取消呼叫
- 9) `RtcConst. CallCode_Bye` 挂断呼叫

除以上常见的具体状态码之外，其他响应码可以分为 3 类，每一类有一个范围：大于等于 `RtcConst. CallCode_RequestErr` 而小于 `RtcConst. CallCode_ServerErr` 一般是请求失败（暂时无法接通）；大于等于 `RtcConst. CallCode_ServerErr` 而小于 `RtcConst. CallCode_GlobalErr` 一般是服务器错误（暂时无法接通）；大于等于 `RtcConst. CallCode_GlobalErr` 一般是全局错误（暂时无法接通）。

6.7 Q: 收到来电通知时，需要注意什么？

A: 收到来电通知时，为了防止 Activity 被 finish 而无法接收到通知，请在应用中注册 `public void onNewCall(Connection call)` 接口，只要应用存活就可以收到通知。需要注意的是：需立即调用 `setIncomingListener` 设置 Listener 对象，否则应用层可能失去某些通知事件。

6.8 Q: 被叫接听呼叫时，如何设置接听类型？

A: 被叫收到呼叫请求后，调用 `public void accept(int callType)` 接口接听。其中参数 `callType` 为接听类型。音视频呼叫可以音视频/纯音频接听。但是 `aidl` 方式只能音频接听。

6.9 Q: 如何获取呼叫信息？

A: 开发者可以调用 `Connection` 类中的 `public String info()` 接口获取呼叫信息。接口返回值包含呼叫信息：`kIdpID/kCallRemoteUri/kCallInfo/kCallType/kCallDir`，如果为会议呼叫，还包含有会议的相关信息。

6.10 Q: 挂断已有连接，是否需要等待 SDK 释放？

A: 挂断当前呼叫连接，挂断成功后不会回调 `onDisconnected()`，不需要等待 SDK 释放。

7 通话过程中的常见问题：

7.1 Q: 通话建立后，如何获取通话时长？

A: 通话时长只能够在通话结束时通过调用 `Connection` 类中的 `public long getCallDuration()` 接口获取。该接口的返回值单位为毫秒。

7.2 Q: 通话过程中，发现视频画面出现绿栅或者花屏现象，应该怎么处理？

A: 视频画面出现绿栅或者花屏的现象，是由于部分手机的固件存在 bug，采用 `OpenGL` 时图像有些问题，解决的方案是在调用 `public Object createVideoView(boolean isLocal, Context ctx, boolean bOpenGL)` 接口建立视频窗口时，置参数 `bOpenGL` 为 `false`，即不采用 `OpenGL`。此外应用层需要注意适配。

7.3 Q: 如何切换前后摄像头？

A: 开发者可以调用 `public void setCamera(int cameraIndex)` 接口来切换摄像头，参数 `cameraIndex=0` 表示采用后置摄像头，`=1` 表示采用前置摄像头。默认前置优先。

7.4 Q: 视频窗口的界面从隐藏到显示，是否需要重置视频窗口？

A: 界面从隐藏到显示，需要重置视频窗口，图像才能正常显示。开发者可以调用 `public void resetVideoViews()` 接口实现视频窗口重置。

7.5 Q: 视频通话中，截屏功能保存图片的路径在哪里设置？

A: 开发者可以调用 `public int takeRemotePicture (String path)` 接口实现截屏功能。接口参数 `path` 为保存图片的路径。接口返回值 `0` 表示截屏成功，`-1` 或其他表示失败。

7.6 Q: 如何设置摄像头旋转?

A: 摄像头旋转功能由接口 `public int setCameraAngle(int nAIndex)` 实现。接口参数 `nAIndex` 为角度索引: $0=0^{\circ}, 1=90^{\circ}, 2=180^{\circ}, 3=270^{\circ}$ 。

7.7 Q: SDK 是否支持视频录制功能?

A: SDK 支持视频通话时录制视频, 开发者可以调用 `public int callRecordStart(String folder)` 接口录制对方视频, 接口参数 `folder` 为存储音视频文件的文件夹绝对路径, 不包括文件名。调用 `public int callRecordStop()` 接口停止通话录制。

7.8 Q: 通话过程中可能出现的错误代码有哪些? 分别代表什么意思?

A: 通话中常见的错误代码有:

- 1) 408: 请求服务器超时或被呼叫方网络异常;
- 2) 480: 对方未登录, 或异常断开一段时间, 主叫或被叫 NAT 超时;
- 3) 403: 认证失效;
- 4) 503: 网络断开或服务器错误;
- 5) 404: 用户不存在;
- 6) 603: 对方正忙或拒接;

通话相关错误除 403 之外只要给出用户语音或界面提示即可, 用户可继续下一次呼叫。

7.9 Q: 通话过程中能否接听 SIM 卡来电或者拨打传统电话?

A: 通话中时如果接听 SIM 卡来电或者去拨打传统电话, 建议将当前通话挂断。

8 常见的网络问题:

8.1 Q: SDK 初始化时发生网络异常, 应该如何处理?

A: 网络异常会导致 SDK 初始化失败, 此时应通知用户选择是否继续, 继续则应先 `release RtcClient` 对象, 重建 `RtcClient` 对象并调用 `initialize`。初始

化成功后，应用退出销毁前，其他接口失败情况不需要再次初始化。具体的监听接口可见问题 4.2。

8.2 Q: 网络状态发生变化时，如何及时获取 Device 的状态变化？

A: 开发者通过 DeviceListener 类的 `public void onDeviceStateChanged (int state)`回调接口实时监听 Device 的状态切换。当网络状态发送变化导致 Device 状态变化时，该接口的参数 `state` 会给出当前的 Device 状态。关于 `state` 代表的具体 Device 状态码，可见问题 5.5。

8.3 Q: 创建呼叫时，因为网络问题导致的呼叫连接异常的情况有哪些？应该如何处理？

A: 呼叫连接异常的情况及其处理方法见问题 6.6。

8.4 Q: 针对网络突然中断的情况，SDK 是如何处理的？

A: 目前 SDK 在网络断开时，比如 WIFI 被关闭，会通过 `void onDeviceStateChanged (int state)`监听接口通知上层，此时应用需要记住最新的状态，并在适当的时候提示用户无法使用呼叫功能，需要用户先恢复网络后才能继续使用。

8.5 Q: 网络发生异常时，应用是否需要调用 SDK 接口连接 RTC 平台？

A: 在网络异常时，应用一定不要死循环调用任何 SDK 接口去尝试连接 RTC 平台，因为 SDK 内部已经在周期性的尝试，而且在网络恢复时会立即自动重连，应用只需要记录 Device 最新状态或者再更新下界面提示即可，除问题 2.5 所述的情况之外，不需要调用 SDK 的任何接口。

8.6 Q: 开发者应该如何处理常见的呼叫失败？

A: 在调用 SDK 的接口后，监听接口一定会有结果返回，对于网络异常时的呼叫，最长需要 32 秒才会收到响应。对于常见的呼叫失败，首先要检查两端的 Device 状态和网络状态，只有这两者正常时呼叫成功率才会高。

尤其是被叫终端 WIFI 的高级设置里,在休眠状态下是否保持 WLAN 连接,否则锁屏一会后被叫手机网络自动断开了,呼叫就无法成功了。

8.7 Q: 手机锁屏是否会导致 SDK 通信断开?

A: SDK 会在手机锁屏之后仍然保持通信机制,不会因为手机锁屏而中断网络通信。除非在手机 WLAN 高级设置中选择选择休眠时断开 WLAN,这种情况下,手机锁屏后由于 WLAN 断开,SDK 检测到网络中断,会执行问题 8.4 的解决流程。

9 终端与浏览器互通的常见问题:

9.1 Q: 如何实现手机终端与浏览器互通?

A: 手机终端在注册 Device 时,在 `public Device createDevice(String parameters, DeviceListener inListener)`接口的参数 `parameters` 里添加可选参数 `kAccSrtp`,设置该值为 2,即可实现与浏览器互通。

天翼 VV 想要实现和浏览器互通,需要在“设置”界面上点击“支持呼叫浏览器”,否则无法呼叫成功。

10 终端适配常见问题:

10.1 Q: 回声问题的适配是在 SDK 中完成的么?

A: 回声等问题在 VV 应用层有代码适配的,光集成 SDK 是不够的,需要把 VV 适配的代码也加上,适配代码包含在扬声器的设置接口里。

10.2 Q: 免提和听筒切换、音量调节的适配是在 SDK 中完成的么?

A: 免提和听筒切换、音量调节的适配是在天翼 VV 的代码里完成的,并不在 SDK 中。

11 其他常见问题:

11.1 Q: 别的应用可以使用和天翼 VV 相同的广播代码么?

A: 手机上如果存在别的应用使用和天翼 VV 相同的广播代码, 使得天翼 VV 的广播被该应用处理掉, 会导致天翼 VV 登录失败。

11.2 Q: 如果采用 WAP 接入网络, 该如何进行处理?

A: WAP 接入点处理, 当前 WAP 测试电信和移动可以支持 https, 联通 uniwap 和 3gwap 则不支持, 对应代码的开关 `RtcConst.bWapNetSupport`, 检测 WAP 的接口为 `int NetWorkUtil.checkNetworkType(Context mContext)`, 电信 WAP 返回值为 `NetWorkUtil.TYPE_CT_WAP`, 移动、联通 WAP 返回值为 `NetWorkUtil.TYPE_CM_CU_WAP`, 建议用户在电信 WAP 接入点时可继续使用, 否则提示用户切换接入点。