ALIBABA CLOUD

阿里云

实人认证
离线人脸识别SDK

文档版本：20220228

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.接入流程概述

阿里云实人认证提供离线人脸识别SDK，帮助您在弱网或离网环境下进行人脸认证。本文介绍了离线人脸识别SDK的接入流程。

接入说明

离线人脸识别SDK的实际应用效果与硬件配置和设备所处环境密切相关。如果您需要使用离线人脸识别SDK，请先联系我们评估是否可用。

关于如何调用API，请参见API调用方式。

准备工作

1. 登录阿里云官网注册账号。

   如果已有注册账号，请跳过此步骤。

2. 进行企业实名认证。

   如果已经是企业账号，请跳过此步骤。关于如何进行企业实名认证，请参见。

3. 开通实人认证服务。

   在实人认证产品页面，开通实人认证服务。

离线SDK接入

1. 调用CreateVerifySDK和DescribeVerfySDK下载离线人脸识别SDK。

   关于API的详细介绍，请参见CreateVerifySDK和DescribeVerifySDK。

2. 接入Android客户端，并集成离线人脸识别SDK。

   关于离线SDK接入的具体操作，请参见SDK端接入。

3. 调用服务端API。

   i. 获取AccessKey。

   阿里云API的调用需要使用AccessKey来校验请求数据的安全性。关于如何获取AccessKey，请参见获取AccessKey。

   > **说明** 因为阿里云账号的AccessKey具有所有云产品API的访问权限，一旦泄露将导致极大的安全风险，所以强烈建议您根据最小权限原则创建并使用RAM用户来进行API访问和控制台操作。具体操作，请参见RAM用户接入。

ii. 调用CreateAuthKey激活离线人脸识别SDK。

关于CreateAuthKey的具体内容，请参见CreateAuthKey。

如果您需要更新或者查询设备信息，请参考如下API。

API	描述
ModifyDeviceInfo	更新设备相关信息，包括延长授权有效期等。
DescribeDeviceInfo	查询设备信息，包括设备授权的有效期等。

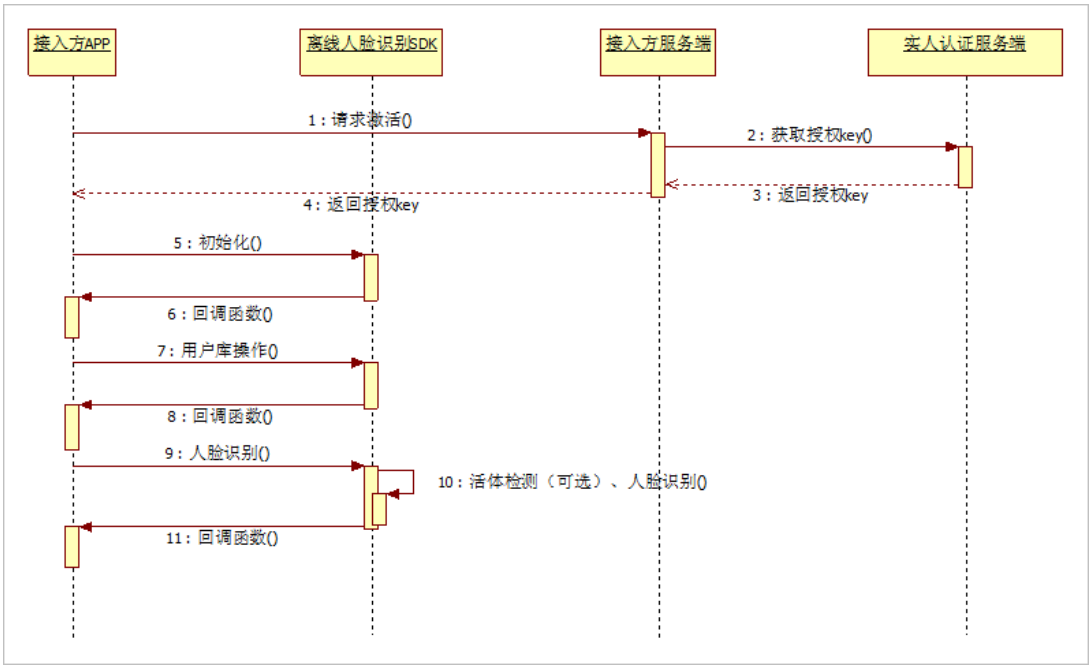
您在调用API接口时，推荐使用我们提供的服务端SDK（包含对应的示例代码），便于您接入和调试。

您可以在阿里云OpenAPI开发者门户上调试实人认证API。

2.接入时序图

介绍离线人脸识别SDK的接入流程。

离线人脸识别SDK接入时序图



时序图说明：

- Step1~Step4：可选。若设备未激活（如首次使用SDK，或授权有效期过期等），则需要调用实人认证服务端接口CreateAuthKey获得授权key，再进行初始化。若设备已激活，则直接从Step5开始，传入授权key进行SDK初始化。
- Step5~Step6：调用SDK的initWithToken函数，传入授权key进行初始化，SDK会回调初始化结果。
- Step7~Step8：可选。若使用人脸1:N检索，则需要进行用户库操作。若使用人脸1:1比对，则不需要该步骤。
- Step9~Step11：进行人脸比对或检索，其中活体检测（翻拍和红外活体）根据接入方需要可开启或关闭，比对/检索结果SDK会回调告知。

3.SDK端接入

本文介绍离线人脸识别SDK接入方式。

授权方式说明

离线人脸识别SDK目前支持在线联网激活，以获得SDK使用授权，被授权的设备和App，在有效期内可以运行SDK，离线完成包括人脸库管理、人脸检测、人脸比对、人脸检索、活体判断等功能。授权结束之后，将无法使用任何功能，需要重新激活获得授权。

授权有效，不会消耗新的授权，仅需联网重新拉取授权：

- 删除SDK
- 设备刷机
- 清除数据
- 恢复出厂设置

授权无效，需要联网激活，且会消耗新的授权：

- 授权有效期到期
- 新设备首次使用

免费测试：我们为每个接入方提供2个免费测试授权，用于SDK接入试用，测试授权在激活后的1个月内有效。

正式购买：客户根据业务需要购买相应时长的SDK使用授权，具体定价可联系阿里云客户经理咨询。

步骤一：SDK下载

离线人脸识别SDK可通过服务端接口进行下载。关于服务端接口的具体内容，请参见CreateVerifySDK和DescribeVerifySDK。

步骤二：在工程中导入SDK

解压离线人脸识别SDK包，将以下Android依赖包引入到您的应用工程中：

- lib-verify-xxx-release.aar
- SecurityBodySDK-external-release.aar
- SecurityGuardSDK-external-release.aar

1. 设定依赖包所在的路径：

```
apply plugin: 'com.android.application'
 repositories {
     flatDir {
         dirs '../libs'
     }
 }
```

2. 设定引入的本地库所在路径，将需要引入的依赖包都放在*../libs*目录，包含所有需要的库。
3. 在*gradle*文件中引入以下需要的库依赖：

```
implementation (name:'lib-verify-xxx-release',ext:'aar')
implementation (name:'SecurityGuardSDK-external-release-5.4.121',ext:'aar')
implementation (name:'SecurityBodySDK-external-release-5.4.79',ext:'aar')
implementation "com.squareup.okhttp3:okhttp:3.12.0" // 版本可以和业务工程的okhttp版本保持一致。
```

> **说明** 未来各依赖库的版本号会有所变化，实际版本号以下载到的SDK为准。

SDK签名图片

- 解压已下载的离线人脸识别SDK包，获得*yw_1222_*.jpg*签名图片文件，该文件在SDK授权时需要使用。
- 把该图片文件导入到工程中*res\drawable*目录，如果没有这个文件夹，请先在工程中创建，否则将无法正常工作。
- 如果在安卓工程打包时启用了shrinkResources true，还需要在*keep.xml*文件中添加以下内容：

```
<resources xmlns:tools="http://schemas.android.com/tools" tools:keep="@drawable/yw_1222_*" />
```

- 离线人脸识别SDK会与App包名（package name）+签名（keystore）绑定，所以若App的package name或keystore有变化都需要重新下载SDK，若无变化，则不需要重新下载。

ABI类型

离线人脸识别SDK目前支持armeabi-v7a、arm64-v8a，请接入方按需在*build.gradle*中增加abiFilters配置。例如，接入方仅需要支持其中armeabi-v7a和arm64-v8a，则配置如下：

```
defaultConfig {
    ndk {
        abiFilters "armeabi-v7a", "arm64-v8a"
    }
}
```

关于gradle（重要）

离线人脸识别SDK目前只支持gradle plugin版本在3.4.1版本及以下。

步骤三：SDK初始化及相关配置

接入方只需关注SDK提供的VerifySDKManager类，上层与SDK的交互都是通过VerifySDKManager进行调用。

在使用SDK前必须通过setNeedDeviceManage开启设备管理功能，且通过initWithToken方法进行初始化。

- **设置是否开启设备管理功能**

  接口名：setNeedDeviceManage

  接口描述：使用initWithToken初始化方法，需要开启设备管理功能。

  setNeedDeviceManage参数说明

名称	类型	描述
needDeviceManage	Boolean	是否开启设备管理。 ○ **true**：表示开启设备管理。 ○ **false**：表示不开启设备管理。

示例代码：

```
VerifySDKManager.getInstance().setNeedDeviceManage(true);
```

- **使用授权key进行初始化**

  接口名：initWithToken

  接口描述：使用授权key，初始化结果通过回调告知。

  initWithToken参数说明

名称	类型	描述
context	String	当前activity的上下文。
token	String	激活使用的授权key，从实人认证服务端申请获得。
initWithTokenCallback	InitWithTokenCallback	回调函数，获取初始化结果。

> **说明** 如果设备目录*/sdcard/verify/verifysdk/ab.bin*存在且有效，可以使用 `token=""` 直接进行本地断网初始化。如果本地初始化失败，需要联网在线初始化，激活使用的授权key，从实人认证服务端CreateAuthKey接口获得，在线联网初始化会将最新的授权文件ab.bin下载到本地，完成SDK授权。

示例代码：

```
// 初始化结果字段。
private int regCode = -1;
// 回调函数。
private OnInitListener initWithTokenCallback = new OnInitListener(){
    @Override
    public void onInitResult(int code) {
        regCode = code;
        Log.i("initData","return code: " + code);
    }
};
// activity onCreate方法中进行SDK初始化。
VerifySDKManager.getInstance().setNeedDeviceManage(true);
VerifySDKManager.getInstance().initWithToken(getApplicationContext(),"",initWithTokenCallback);
if(regCode !=0){
      // 调用接入方服务端，服务端通过调用实人认证服务端接口CreateAuthKey获取。
      // String token =
VerifySDKManager.getInstance().initWithToken(getApplicationContext(),token,initWithTokenCallback);
  }
```

SDK提供相关设置函数。例如，设置人脸检测的最小尺寸、是否开启翻拍检测、是否开启红外活体检测、翻拍检测阈值、红外活体阈值、人脸检索匹配阈值等，接入方可以按需设置。

- **设置最小检测人脸**

接口名：setMinFaceDetectSize

接口描述：设置最小检测人脸，图像中的人脸大小超过该值时，才能检测到人脸，否则无法检测到人脸。

setMinFaceDetectSize参数说明

名称	类型	描述
minFaceDetectSize	Float	人脸检测的最小尺寸，取值范围0-1，定义为占图像min（宽，高）的比例，默认值*0.1*。

示例代码：

```
VerifySDKManager.getInstance().setMinFaceDetectSize(0.05f);
```

- **设置人脸位置有效区域边界**

  接口名：setBorders

  接口描述：设置人脸位置有效区域边界。

  setBorders参数说明

名称	类型	描述
leftBorder	Float	人脸位置有效区域的左边界，定义为占旋转后图像宽度的比例，默认值*0.1*。
rightBorder	Float	人脸位置有效区域的右边界，定义为占旋转后图像宽度的比例，默认值*0.9*。
topBorder	Float	人脸位置有效区域的上边界，定义为占旋转后图像高度的比例，默认值*0.1*。
bottomBorder	Float	人脸位置有效区域的下边界，定义为占旋转后图像高度的比例，默认值*0.9*。

  示例代码：

```
VerifySDKManager.getInstance().setBorders(0.1f,0.9f,0.1f,0.9f);
```

- **设置是否开启离线翻拍检测**

  接口名：setNeedRecapCheck

  接口描述： 设置是否开启离线翻拍检测。

  setNeedRecapCheck参数说明

名称	类型	描述
needRecapCheck	Boolean	是否开启离线翻拍检测。 ○ **true**：表示开启离线翻拍检测。 ○ **false**：表示不开启离线翻拍检测。

  示例代码：

```
VerifySDKManager.getInstance().setNeedRecapCheck(true);
```

- **设置是否开启红外防彩色图片翻拍检测**

  接口名：setNeedNirSeniorRecapCheck

接口描述：设置是否开启红外防彩色图片翻拍检测。

setNeedNirSeniorRecapCheck参数说明

名称	类型	描述
needNirSeniorRecapCheck	Boolean	是否开启红外防彩色图片翻拍检测。 ○ **true**：表示开启红外防彩色图片翻拍检测。 ○ **false**：表示不开启红外防彩色图片翻拍检测。

示例代码：

```
VerifySDKManager.getInstance().needNirSeniorRecapCheck(true);
```

- **设置是否需要红外活体能力**

  接口名：setNeedNirLiveness

  接口描述：设置是否需要红外活体能力，若设置需要红外活体能力，那么setNirFrameW、setNirFrameH、setNirAngle、setRgbAngle为必要参数，最后再调用init方法进行生效。

  setNeedNirLiveness参数说明

名称	类型	描述
needNirLiveness	Boolean	是否需要红外活体能力。 ○ **true**：表示需要红外活体能力。 ○ **false**：表示不需要红外活体能力。

  示例代码：

```
VerifySDKManager.getInstance().needNirLiveness(true).setNirAngle(0).setNirFrameH(480).setNirFrameW(640).setRgbAngle(0).init(getApplicationContext());
```

- **设置红外图像的帧高度**

  接口名：setNirFrameH

  接口描述：设置红外图像的帧高度，默认值为720。

  setNirFrameH参数说明

名称	类型	描述
nirFrameH	Integer	红外图像的帧高度，默认值为*720*。

- **设置红外图像的帧宽度**

  接口名：setNirFrameW

  接口描述：设置红外图像的帧宽度，默认值为720。

  setNirFrameW参数说明

名称	类型	描述
nirFrameW	Integer	红外图像的帧宽度，默认值为*720*。

- **设置红外图像的旋转角度**

  接口名：setNirAngle

  接口描述：设置红外图像的旋转角度，默认值为0。

  setNirAngle参数说明

名称	类型	描述
nirAngle	Integer	红外图像的旋转角度，默认值为*0*。

- **设置RGB图像的旋转角度**

  接口名：setRgbAngle

  接口描述：设置RGB图像的旋转角度，默认值为0。

  setRgbAngle参数说明

名称	类型	描述
rgbAngle	Integer	RGB图像的旋转角度，默认值为*0*。

- **设置翻拍检测阈值**

  接口名：setRecapThreshold

  接口描述：设置翻拍检测阈值，目前翻拍默认分值是80f，取值范围0~100，翻拍得分大于等于80f则认为是翻拍，小于80f则认为是真人。接入方可根据业务实际需要进行阈值调整。

  setRecapThreshold参数说明

名称	类型	描述
recapThreshold	Float	设置翻拍检测阈值，默认值*80f*。

  示例代码：

```
VerifySDKManager.getInstance().setRecapThreshold(80f);
```

- **设置红外活体阈值**

  接口名：setNirScoreThreshold

  接口描述：设置红外活体阈值，目前默认分值是0f，取值范围0~1，接入方可根据业务实际需要进行阈值调整。

  setNirScoreThreshold参数说明

名称	类型	描述
nirScoreThreshold	Float	设置红外活体阈值，默认值*0f*。

示例代码：

```
VerifySDKManager.getInstance().setNirScoreThreshold(0f);
```

- **设置人脸匹配阈值**

  接口名：setFaceMatchThreshold

  接口描述：设置人脸匹配阈值，目前默认分值是0.44f，取值范围0~1，接入方可根据业务实际需要进行阈值调整。

  setFaceMatchThreshold参数说明

名称	类型	描述
faceMatchThreshold	Float	设置人脸匹配阈值，默认值*0.44f*。

  示例代码：

```
VerifySDKManager.getInstance().setFaceMatchThreshold(0f);
```

- **设置支持口罩识别能力**

  接口名：setNeedSupportMaskVerify

  接口描述：默认口罩识别能力是关闭的，如果需要使用，需要手动设置打开。打开后，在底照入库时，一个用户需要同时添加不带口罩照片和戴口罩照片，可以达到最好的戴口罩识别效果。

  示例代码：

```
VerifySDKManager.getInstance().setNeedSupportMaskVerify(true);
```

步骤四：SDK用户库操作

初始化成功后，如果要使用SDK的人脸1：N检索能力，需要进行相应用户库操作，向人脸库中添加人脸图片，否则人脸检索将无法匹配。

如果要使用SDK的人脸1:1比对能力，则可以不关注用户库操作，请参见人脸识别部分的说明。

- **加载用户库**

  接口名：loadUserLib

  接口描述：用于初始时加载全量用户库。

  loadUserLib参数说明

名称	类型	描述
listener	VerifyLibEventListener	用户库操作回调。

  示例代码：

```
// 加载用户数据到内存。
VerifySDKManager.getInstance().loadUserLib(verifyLibEventListener);
```

- **添加本地用户照片**

  接口名：addUser

  接口描述：添加本地用户照片到用户库，操作结果以回调方式通知，完成后不必重新loadUserLib。添加后，用户照片数据会存储在设备本地数据库，适用于人脸底库数量不大或设备存储空间相对充裕的情况。

  > **说明** 使用该函数添加用户照片，优势是当人脸算法模型更新升级时，接入方不需要再重新入库一次用户照片，SDK会根据本地已有的照片数据重新提取特征值，支持在新模型下的运行；劣势是照片数据会占用一定的设备存储空间。

  addUser参数说明

名称	类型	描述
isSync	Boolean	是否同步调用。 ○ **true**：表示建议上层调用入库自己管理线程。 ○ **false**：表示不需要上层调用入库自己管理线程。
id	String	用户ID，需要保证唯一性。
type	Integer	生物特征类型，目前只支持*BiologyType.BIOLOGY_FACE*。
featureData	byte[]	用户照片的byte源数据，支持JPG、PNG格式。
verifyLibEventListener	VerifyLibEventListener	用户库操作回调。

  示例代码：

```
VerifySDKManager.getInstance().addUser(true, String.valueOf(i), BiologyType.BIOLOGY_FACE,
FileUtil.getFileBuffer(path),verifyLibEventListener);
```

  接口名：addUserWithoutFeatureData

  接口描述：添加本地用户照片，操作结果以回调方式通知。添加后，用户照片数据不会存储在设备本地数据库，适用于人脸底库数量大或设备存储空间有限的情况。

  > **说明**
  > ○ 使用该函数添加用户照片，优势是不占用设备存储空间；劣势是当人脸算法模型更新升级时，本地已有的特征库无法适配新模型，SDK会运行失败，需要接入方再重新入库一次用户照片，以便新模型重新提取特征值，确保SDK正常运行。
  > ○ 使用该函数的接入方，需要在onUserLibLoaded回调函数中处理errorCode=104的错误码，详见onUserLibLoaded调用示例代码。

  addUserWithoutFeatureData参数说明

名称	类型	描述

名称	类型	描述
isSync	Boolean	是否同步调用。 ○ **true**：表示建议上层调用入库自己管理线程。 ○ **false**：表示不建议上层调用入库自己管理线程。
id	String	用户ID，需要保证唯一性。
type	Integer	生物特征类型，目前只支持*BiologyType.BIOLOGY_FACE*。
featureData	byte[]	用户照片的byte源数据，支持JPG、PNG格式。
verifyLibEventListener	VerifyLibEventListener	用户库操作回调。

示例代码：

```
VerifySDKManager.getInstance().addUserWithoutFeatureData(true, String.valueOf(i), BiologyType.BIOLOGY_FACE,FileUtil.getFileBuffer(path),verifyLibEventListener);
```

- **删除指定用户生物数据**

  接口名：removeUser

  接口描述：删除指定用户生物数据，操作结果以回调方式通知，无需重新loadUserLib。

  removeUser参数说明

名称	类型	描述
isSync	Boolean	是否同步调用。 ○ **true**：表示建议上层调用入库自己管理线程。 ○ **false**：表示不需要上层调用入库自己管理线程。
id	String	用户ID。
verifyLibEventListener	VerifyLibEventListener	用户库操作回调。

  示例代码：

```
VerifySDKManager.getInstance().removeUser(false,id,verifyLibEventListener);
```

- **清空所有用户数据**

  接口名：clearUserLib

  接口描述：清除本地用户库。

  clearUserLib参数说明

名称	类型	描述
isSync	Boolean	是否同步调用。 ○ **true**：表示建议上层调用入库自己管理线程。 ○ **false**：表示不建议上层调用入库自己管理线程。
verifyLibEventListener	VerifyLibEventListener	用户库操作回调。

示例代码：

```
VerifySDKManager.getInstance().clearUserLib(false,verifyLibEventListener);
```

- **单用户数据更新回调**

  接口名：onSingleUserLibUpdate

  接口描述：用户库操作回调，单用户数据更新。

  onSingleUserLibUpdate参数说明

名称	类型	描述
id	String	用户ID。
errorCode	Integer	错误码，0为正确。

- **用户库加载完成回调**

  接口名：onUserLibLoaded

  接口描述：loadUserLib操作回调，用户库加载完成。

  onUserLibLoaded参数说明

名称	类型	描述
errorCode	Integer	错误码，0为正确。 **说明** 使用addUserWithoutFeatureData函数添加用户的接入方，需要特殊处理该回调函数errorCode=104的错误码，详见下方示例代码中说明。

- **用户库清除回调**

  接口名：onUserLibEmpty

  接口描述：clearUserLib操作回调，用户库清除完成。

  onUserLibEmpty参数说明

名称	类型	描述
errorCode	Integer	错误码，0为正确。

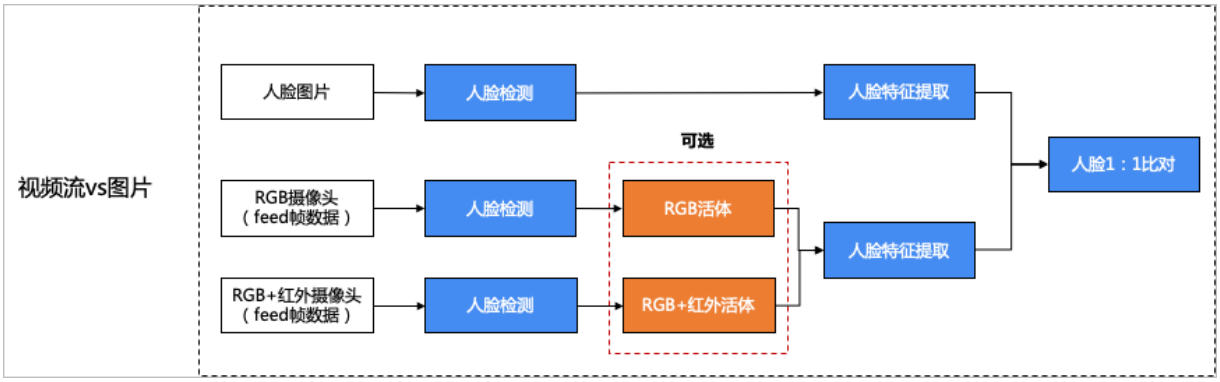
示例代码：

```
// 用户库操作回调。
final OnVerifyLibEventListener verifyLibEventListener = new OnVerifyLibEventListener() {
        @Override
        public void onSingleUserLibUpdate(String id, final int errorCode) {
        // 单用户更新操作完成后处理。
        }
        @Override
        public void onBatchUserLibUpdate(int errorCode) {
        // 批量用户更新，暂时不支持。
        }
       @Override
        public void onUserLibLoaded(int errorCode) {
        // loadUserLib操作完成后处理。
        //特殊说明：使用addUserWithoutFeatureData函数添加用户的接入方，需要在该回调中特殊处理errorCode=104的错误码，操作步骤如下：
        //第一步：调用clearUserLib清除本地数据。
        //第二步：接入方重新入库用户数据。
        }
        @Override
        public void onUserLibEmpty(int errorCode) {
        // clearUserLib操作完成后处理。
        }
};
```

步骤五：SDK人脸识别

人脸1：1比对说明

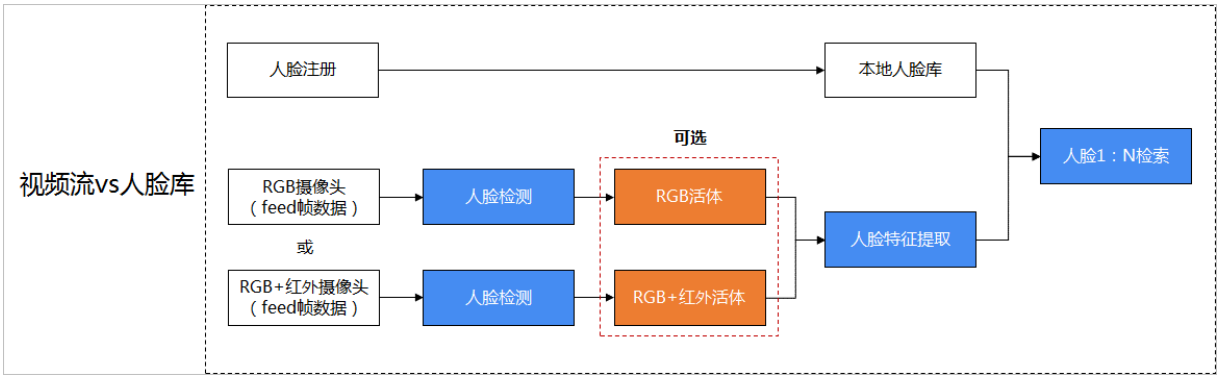
SDK支持实时视频流和人脸图片进行比对，这是最为常见的1：1对比类型。针对一张事先获取的图片（通常为身份证芯片照、证件照片等），与摄像头实时采集的符合条件的人脸图片进行比对。通常适用于有人值守的场景。



人脸1：N检索

将需要识别的人脸图片注册到本地人脸库中，当有用户需要识别身份时，从视频流中实时采集符合条件的人脸图片，与人脸库中的人脸集合比对，得到检索结果。如果是无人值守的情况，建议可以开启翻拍检测或红外活体保障安全性。

如果在初始化时设置开启了翻拍检测或红外活体检测，则摄像头采集的人脸图片必须同时通过活体检测，才能进入人脸检索环节，任一活体检测未通过，都不会进行人脸检索。



- 人脸1:1比对调用
  - 接口名：doFaceMatchWithImage

    接口描述：RGB摄像头帧数据与人脸图片进行1：1比对，结果以回调方式通知。适用于设备上只有RGB单目摄像头，或者有双目摄像头但未开启红外活体检测的情况，可以使用该接口进行1：1比对。

    doFaceMatchWithImage参数说明

名称	类型	描述
data	byte[]	检测到合格人脸时的RGB帧数据。
width	Integer	帧数据宽度。
height	Integer	帧数据高度。
imageFormat	Integer	帧数据格式，Android默认选择ImageFormat.NV21。
imageAngle	Integer	读取到合格人脸时的图片旋转角度，取值0/90/180/270，从摄像头的回调接口中获取。
cameraRotation	Integer	读取到合格人脸时的相机旋转角度，取值0/90/180/270，从摄像头的回调接口中获取。
targetBitmap	Bitmap	待比对人脸图片。
listener	FaceMatchWithImageListener	1:1比对回调。

    示例代码：

```
// 当未开启红外活体功能时，使用OnFaceMatchWithImageListener类型做回调。
// 1:1比对的回调
OnFaceMatchWithImageListener faceMatchWithImageListener = new
OnFaceMatchWithImageListener() {
    @Override
    @param faceMatchResult可以通过faceMatchResult.getMatchItems().get(0).getScore()获取对比分数。
    public void onMatchResult(FaceMatchResult faceMatchResult) {
     // 与目标图片匹配后的处理。
}
@Override
public void onRecapDetected(byte[] data,
int width, int height, int imageAngle, int cameraRotation, float recapScore){
  // 检测到人脸翻拍。
}
};
// 1:1比对调用。
VerifySDKManager.getInstance().doFaceMatchWithImage(data,width,height, ImageFormat.NV21
, degree, cameraRotation, targetBitmap, faceMatchWithImageListener)
```

- 接口名：doFaceMatchWithImageWithNir

  接口描述：RGB和红外摄像头帧数据与人脸图片进行1：1比对，结果以回调方式通知。适用于设备上有双目红外的摄像头，在开启红外活体检测的情况下，可以使用该接口进行1：1比对。

  doFaceMatchWithImageWithNir参数说明

名称	类型	描述
nirData	byte[]	检测到合格人脸时的红外帧数据。
rgbData	byte[]	检测到合格人脸时的RGB帧数据。
width	Integer	RGB帧数据宽度。
height	Integer	RGB帧数据高度。
imageFormat	Integer	帧数据格式，Android默认选择ImageFormat.NV21。
imageAngle	Integer	读取到合格人脸时的图片旋转角度，取值0/90/180/270，从摄像头的回调接口中获取。
cameraRotation	Integer	读取到合格人脸时的相机旋转角度，取值0/90/180/270，从摄像头的回调接口中获取。
targetBitmap	Bitmap	待比对人脸图片。

名称	类型	描述
nirFaceDetectListener	FaceMatchWithImageListener	人脸检测、检索比对、翻拍检测回调。

示例代码：

```
// 当开启红外检测时，使用OnFaceMatchWithImageNirListener类型做回调。
final OnFaceMatchWithImageNirListener faceMatchWithImageNirListener = new OnFaceMatchWithImageNirListener() {
       @Override
       @param faceMatchResult可以通过faceMatchResult.getMatchItems().get(0).getScore()获取对比分数。
    public void onMatchResult(FaceMatchResult faceMatchResult) {
     // 与目标图片匹配后的处理。
}
        @Override
        public void onRecapDetected(byte[] data, int width, int height, int imageAngle, int cameraRotation, float recapScore) {
    // 检查到人脸翻拍。
        }
        @Override
        public void onFaceDetectedInNIR(byte[] nirData, int width, int height, int nirAngle, NirFaceInfo nirFaceInfo){
            // 红外摄像头检查到人脸。
        }
        @Override
        public void onNofaceDetectedInNIR(byte[] nirData, int width, int height, int nirAngle){
    // 红外摄像头没有检查到人脸。
        }
    };
//  1：1比对接口调用，带红外活体。
VerifySDKManager.getInstance().doFaceMatchWithImageWithNir(nirData,data,width, height,ImageFormat.NV21,degree,cameraRotation, faceMatchWithImageNirListener);
```

- 人脸1：1比对结果回调
  - 接口名：onMatchResult

    接口描述： 人脸1：1比对结果回调，可获得相应比对分。

    onMatchResult参数说明

名称	类型	描述
result	FaceMatchResult	人脸1:1比对结果。

示例代码：见人脸1：1比对调用代码示例。

- 接口名：onRecapDetected

  接口描述：检测到翻拍，且初始化时开启了翻拍检测时，会执行这个回调。

  接口的参数说明可以参考1：N检索中的该回调函数。

- 接口名：onFaceDetectedInNIR

  接口描述：红外帧数据中检测到人脸时，会回调该函数，调用方可根据此回调做相应处理。

  接口的参数说明可以参考1：N检索中的该回调函数。

- 接口名：onNofaceDetectedInNIR

  接口描述：红外帧数据中未检测到人脸。

  接口的参数说明可以参考1：N检索中的该回调函数。

- **人脸1：N检索调用**
  - 接口名：feedPreviewFrame

    接口描述：RGB摄像头帧数据与人脸库进行1:N检索，结果以回调方式通知。适用于设备上只有RGB单目摄像头，或者有双目摄像头但未开启红外活体检测的情况，可以使用该接口进行检索比对。

    feedPreviewFrame参数说明

名称	类型	描述
data	byte[]	检测到合格人脸时的RGB帧数据。
width	Integer	帧数据宽度。
height	Integer	帧数据高度。
imageFormat	Integer	帧数据格式，Android默认选择*ImageFormat.NV21*。
imageAngle	Integer	读取到合格人脸时的图片旋转角度，取值0/90/180/270，从摄像头的回调接口中获取。
cameraRotation	Integer	读取到合格人脸时的相机旋转角度，取值0/90/180/270，从摄像头的回调接口中获取。
faceDetectWithMatchListener	FaceDetectWithMatchListener	人脸检测、检索比对、翻拍检测回调。

    示例代码：

```
// 当未开启红外活体功能时，使用OnFaceDetectWithMatchListener类型做回调。
final OnFaceDetectWithMatchListener
faceDetectListener = new OnFaceDetectWithMatchListener() {
        @Override
        public void onFaceDetected(byte[] data, final int width, final int height, int
imageFormat, int imageAngle, int cameraRotation, final FaceInfo faceInfo) {
    // 识别到人脸时。
        }
        @Override
        public void onNofaceDetected(byte[] data, int width, int height, int imageForma
t, int imageAngle, int cameraRotation) {
    // 没有识别到人脸时。
        }
        @Override
        public void onFaceMatched(byte[] data, int width, int height, int imageAngle, i
nt cameraRotation, final FaceMatchResult matchResult, final long costTime) {
    // 识别到人脸并在用户库中匹配用户。
        }
        @Override
        public void onRecapDetected(byte[] data, int width, int height, int imageAngle,
int cameraRotation, float recapScore) {
    // 检查到人脸翻拍。
        }
        @Override
        public void onFaceMoving(boolean isMoving) {
    // 检查到人脸移动。
        }
// 1：N检索接口调用，不带红外活体。
VerifySDKManager.getInstance().feedPreviewFrame(data,width, height,ImageFormat.NV21,deg
ree,cameraRotation, faceDetectWithMatchListener);
```

- 接口名：feedPreviewFrameWithNir

  接口描述：RGB和红外摄像头帧数据与人脸库进行1：N检索，结果以回调方式通知。适用于设备上有双目红外的摄像头，在开启红外活体检测的情况下，可以使用该接口进行检索比对。

  feedPreviewFrameWithNir参数说明

名称	类型	描述
nirData	byte[]	检测到合格人脸时的红外帧数据。
rgbData	byte[]	检测到合格人脸时的RGB帧数据。
width	Integer	RGB帧数据宽度。
height	Integer	RGB帧数据高度。
imageFormat	Integer	帧数据格式，Android默认选择*ImageFormat.NV21*。

名称	类型	描述
imageAngle	Integer	读取到合格人脸时的图片旋转角度，取值0/90/180/270，从摄像头的回调接口中获取。
cameraRotation	Integer	读取到合格人脸时的相机旋转角度，取值0/90/180/270，从摄像头的回调接口中获取。
nirFaceDetectListener	FaceMatchWithImageListener	人脸检测、检索比对、翻拍检测回调。

示例代码：

```
// 当开启红外检测时，使用OnNirFaceDetectListener类型做回调。
final OnNirFaceDetectListener nirFaceDetectListener = new OnNirFaceDetectListener() {
        @Override
        public void onFaceDetected(byte[] data, final int width, final int height, int
imageFormat, int imageAngle, int cameraRotation, final FaceInfo faceInfo) {
    // 识别到人脸时。
        }
        @Override
        public void onNofaceDetected(byte[] data, int width, int height, int imageForma
t, int imageAngle, int cameraRotation) {
    // 没有识别到人脸时。
        }
        @Override
        public void onFaceMatched(byte[] data, int width, int height, int imageAngle, i
nt cameraRotation, final FaceMatchResult matchResult, final long costTime) {
    // 识别到人脸并在用户库中匹配用户。
        }
        @Override
        public void onRecapDetected(byte[] data, int width, int height, int imageAngle,
int cameraRotation, float recapScore) {
    // 检查到人脸翻拍。
        }
        @Override
        public void onFaceMoving(boolean isMoving) {
    // 检查到人脸移动。
        }
        @Override
        public void onFaceDetectedInNIR(byte[] nirData, int width, int height, int nirA
ngle, NirFaceInfo nirFaceInfo){
            // 红外摄像头检查到人脸。
        }
        @Override
        public void onNofaceDetectedInNIR(byte[] nirData, int width, int height, int ni
rAngle){
    // 红外摄像头没有检查到人脸。
        }
    };
// 1：N检索接口调用，带红外活体。
VerifySDKManager.getInstance().feedPreviewFrameWithNir(nirData,data,width, height,Image
Format.NV21,degree,cameraRotation,nirFaceDetectListener);
```

- **人脸1：N检测到人脸回调**

- 接口名：onFaceDetected

  接口描述：RGB帧数据中检测到人脸时，会回调该函数，调用方可根据此回调绘制人脸框。

  onFaceDetected参数说明

名称	类型	描述
data	byte[]	RGB帧数据。
width	Integer	RGB帧数据宽度。
height	Integer	RGB帧数据高度。
imageFormat	Integer	帧数据格式，Android默认选择*ImageFormat.NV21*。
imageAngle	Integer	图片旋转角度。
cameraRotation	Integer	相机旋转角度。
faceInfo	FaceInfo	检测到的人脸信息在帧中的坐标，宽高。

  示例代码：见1：N检索调用代码。

- 接口名：onFaceDetectedInNIR

  接口描述：红外帧数据中检测到人脸时，会回调该函数，调用方可根据此回调做相应处理。

  onFaceDetectedInNIR参数说明

名称	类型	描述
nirData	byte[]	红外帧数据。
width	Integer	红外帧数据宽度。
height	Integer	红外帧数据高度。
nirAngle	Integer	图片旋转角度。
nirFaceInfo	NirFaceInfo	检测到的红外人脸信息在帧中的坐标，宽高。

  示例代码：见1：N检索调用代码。

- **人脸1：N未检测到人脸回调**

- 接口名：onNofaceDetected

  接口描述：RGB帧数据中未检测到人脸，可以在这个回调中更新UI，移除之前绘制的人脸框。

  onNofaceDetected参数说明

名称	类型	描述
data	byte[]	RGB帧数据。
width	Integer	RGB帧数据宽度。
height	Integer	RGB帧数据高度。
imageFormat	Integer	帧数据格式，Android默认选择*ImageFormat.NV21*。
imageAngle	Integer	图片旋转角度。
cameraRotation	Integer	相机旋转角度。

  示例代码：见1：N检索调用代码。

- 接口名：onNofaceDetectedInNIR

  接口描述：红外帧数据中未检测到人脸。

  onNofaceDetectedInNIR参数说明

名称	类型	描述
nirData	byte[]	红外帧数据。
width	Integer	红外帧数据宽度。
height	Integer	红外帧数据高度。
nirAngle	Integer	图片旋转角度。

  示例代码：见1：N检索调用代码。

- **人脸1：N检索结果回调**

  接口名：onFaceMatched

  接口描述：人脸检索匹配结果回调。

  onFaceMatched参数说明

名称	类型	描述
data	byte[]	RGB帧数据。
width	Integer	RGB帧数据宽度。
height	Integer	RGB帧数据高度。
imageAngle	Integer	图片旋转角度。
cameraRotation	Integer	相机旋转角度。
matchResult	FaceMatchResult	匹配结果，极端情况下可能存在多个人的情况，比如双胞胎。
costTime	Long	人脸检索比对耗时，性能分析调试用。

示例代码：见1：N检索调用代码。

- **检测到翻拍回调**

  接口名：onRecapDetected

  接口描述：检测到翻拍，且初始化时开启了翻拍检测时，会执行这个回调。

  onRecapDetected参数说明

名称	类型	描述
data	byte[]	RGB帧数据。
width	Integer	RGB帧数据宽度。
height	Integer	RGB帧数据高度。
imageAngle	Integer	图片旋转角度。
cameraRotation	Integer	相机旋转角度。
recapScore	Float	翻拍得分。

  示例代码：见1：N检索调用代码。

- **人脸移动时的回调**

  接口名：onFaceMoving

  接口描述：检测到人脸在移动时，会回调该函数。

  onFaceMoving参数说明

名称	类型	描述
isMoving	Boolean	人脸是否移动。

示例代码：见1：N检索调用代码。

SDK错误码

错误码	错误信息	错误描述
100	VERIFYSDK_ERR_CODE_BAD_PARAM	参数错误。
101	VERIFYSDK_ERR_CODE_ACCESS_WORK_FAILED	访问工作路径错误。
104	VERRORCODE_NEED_MANUAL_UPDATE	人脸特征库与算法不匹配，SDK启动失败。
1000	VERIFYSDK_ERR_CODE_INVALID_PARAM	人脸引擎参数无效。
1001	VERIFYSDK_ERR_CODE_SDK_NOT_INIT	人脸引擎未初始化。
1002	VERIFYSDK_ERR_CODE_INIT_MULT_TIME	人脸引擎初始化多次。
1100	VERIFYSDK_ERR_CODE_FILE_NOT_EXIST	人脸模型文件不存在。
1101	VERIFYSDK_ERR_CODE_FILE_SIZE_ERROR	人脸模型文件大小错误。
1102	VERIFYSDK_ERR_CODE_VERSION_NOT_MATCH	人脸模型版本不匹配。
11020	VERIFYSDK_ERR_CODE_VERSION_NOT_MATCH_FD	人脸模型版本不匹配子错误。
11021	VERIFYSDK_ERR_CODE_VERSION_NOT_MATCH_LDM	人脸模型版本不匹配子错误。
11022	VERIFYSDK_ERR_CODE_VERSION_NOT_MATCH_LDC	人脸模型版本不匹配子错误。
11023	VERIFYSDK_ERR_CODE_L_VERSION_NOT_MATCH_FE	人脸模型版本不匹配子错误。
11024	VERIFYSDK_ERR_CODE_VERSION_NOT_MATCH_FEENC	人脸模型版本不匹配子错误。
1103	VERIFYSDK_ERR_CODE_CANCEL	取消。
1104	VERIFYSDK_ERR_CODE_BUSY	数据库并发操作。
1201	VERIFYSDK_ERR_CODE_FEATURE_SIZE_ERROR	人脸特征大小错误。
5000	VERIFYSDK_ERR_CODE_LICENCE	授权错误。

错误码	错误信息	错误描述
5001	VERIFYSDK_ERR_CODE_LICENCE_INPUT	输入参数错误。
5002	VERIFYSDK_ERR_CODE_LICENCE_DATA_FORMAT	授权文件格式错误。
5003	VERIFYSDK_ERR_CODE_LICENCE_SIGN	签名校验错误。
5004	VERIFYSDK_ERR_CODE_LICENCE_APKPKG	包名校验错误。
5005	VERIFYSDK_ERR_CODE_LICENCE_PUBKEY	公钥校验错误。
5006	VERIFYSDK_ERR_CODE_LICENCE_EXPIRE	授权过期。
5007	VERIFYSDK_ERR_CODE_LICENCE_NOT_CHECK	未进行授权检查。
5009	VERIFYSDK_ERR_CODE_LICENCE_CLIENT_ID	终端标识校验错误。
5010	VERIFYSDK_ERR_CODE_LICENCE_EXPIRE_DATE_MODIFIED	授权失效时间被修改。
6000	VERIFYSDK_ERR_CODE_NOT_SUPPORT_DEVICE	不支持的硬件型号。
6001	VERIFYSDK_ERR_CODE_LOAD_LIBARY_FAILED	加载so失败。
6002	VERIFYSDK_ERR_CODE_LICENCE_NOT_EXIST	授权文件不存在。
7001	VERIFYSDK_ERR_CODE_SECURITY_TOKEN_SOCKET_TIMEOUT	获取激活key超时，即初始化失败或未完成。
7003	VERIFYSDK_ERR_CODE_GET_STATIC_DATA_STORE_COMP	SDK安全组件获取appkey失败。
7004	VERIFYSDK_ERR_CODE_GET_SECURE_SIGNATURE_COMP	SDK安全组件获取加签失败。
7002	VERIFYSDK_ERR_CODE_SG_SECEXCEPTION	SDK安全组件异常。
8001	VERIFYSDK_ERR_CODE_GET_LICENSE_INFO_TOKEN_NULL	带授权key激活时，key为空。

错误码	错误信息	错误描述
8002	VERIFYSDK_ERR_CODE_GET_LICENSE_INFO_NATIVE_FAILED	获取授权失败。
8003	VERIFYSDK_ERR_CODE_GET_LICENSE_CHARSETNAME_EXCEPTION	授权key编码转换出错。
9001	VERIFYSDK_ERR_CODE_USER_LIB_NOT_LOADED	人脸库还未初始化完成。
9999	VERIFYSDK_ERR_CODE_UNKNOWN	人脸模块未知错误。
10001	VERIFYSDK_ERR_CODE_ACTIVATE_LICENSE_REQUEST_FAILED	请求失败，激活SDK失败。
10002	VERIFYSDK_ERR_CODE_ACTIVATE_LICENSE_DEVICE_NOT_AUTH	未授权，激活SDK失败。
10003	VERIFYSDK_ERR_CODE_ACTIVATE_LICENSE_DEVICE_AUTH_EXPIRED	授权到期失效，激活SDK失败。
10004	VERIFYSDK_ERR_CODE_ACTIVATE_LICENSE_DATA_NULL	授权文件为空，激活SDK失败。

4.服务端接入

4.1. CreateVerifySDK

调用CreateVerifySDK提交离线人脸识别SDK下载任务。

请求方法：支持以HTTPS POST和GET方法发送请求。

接口描述：提交无线应用，异步生成离线人脸识别SDK，一般可在1分钟内生成完成。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CreateVerifySDK	要执行的操作。取值：**CreateVerifySDK**。
AppUrl	String	是	https://www.aliyundoc.com	接入方无线应用程序的可访问链接。实人认证服务端会从该地址下载无线应用，进行SDK生成，接入方务必保证该地址可访问，否则SDK会生成失败。
Platform	String	否	IOS	操作系统类型，可选值： • **ANDROID** • **IOS**

返回数据

名称	类型	示例值	描述
RequestId	String	B506328A-D84B-4750-82C7-6A207C585CF1	本次请求的ID。
TaskId	String	vsdk.g.4fcMqTTf4i76NhlH8wYMYi-1s7Nxt	生成SDK的任务ID，用于查询SDK生成结果，一般生成可以在1分钟内完成。

示例

请求示例

```
https://cloudauth.aliyuncs.com/?Action= CreateVerifySDK
&AppUrl=https://www.aliyundoc.com
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<CreateVerifySDKResponse>
      <TaskId>vsdk.g.4fcMqTTf4i76NhIH8wYMYi-1s7Nxt</TaskId>
      <RequestId>B506328A-D84B-4750-82C7-6A207C585CF1</RequestId>
</CreateVerifySDKResponse>
```

`JSON` 格式

```
{
 "TaskId": "vsdk.g.4fcMqTTf4i76NhIH8wYMYi-1s7Nxt",
 "RequestId": "B506328A-D84B-4750-82C7-6A207C585CF1"
}
```

错误码

访问错误中心查看更多错误码。

4.2. DescribeVerifySDK

调用DescribeVerifySDK获取离线SDK下载地址。

请求方法：支持以HTTPS POST和GET方法发送请求。

接口描述：根据生成离线人脸识别SDK的任务ID获取SDK生成结果。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeVerifySDK	要执行的操作。取值：**DescribeVerifySDK**。
TaskId	String	是	1KQMcnLd4m37LN2D0F0WCD	生成SDK的任务ID。

返回数据

名称	类型	示例值	描述
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E	本次请求的ID。

名称	类型	示例值	描述
SdkUrl	String	https://www.xxx.com	SDK下载地址。不为空时，表示生成完成。

示例

请求示例

```
https://cloudauth.aliyuncs.com/?Action=DescribeVerifySDK
&TaskId=1KQMcnLd4m37LN2D0F0WCD
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<DescribeVerifySDKResponse>
      <Data>
            <Url>https://www.xxx.com</Url>
      </Data>
      <Success>true</Success>
</DescribeVerifySDKResponse>
```

`JSON` 格式

```
{
  "Data": {
    "SdkUrl":"https://www.xxx.com"
  },
  "Success": true
}
```

错误码

访问错误中心查看更多错误码。

4.3. CreateAuthKey

调用CreateAuthKey获取授权key，用于离线人脸识别SDK激活。

请求方法：支持以HTTPS POST和GET方法发送请求。

> **说明** 授权key在30分钟内有效，且不可重复使用，建议在每次激活前重新获取。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CreateAuthKey	要执行的操作。取值：**CreateAuthKey**。
BizType	String	否	FACE_TEST	业务类型。不超过64字符。可用于对具体业务进行备注，例如可以取接入方不同的人脸使用场景，或者待交付的客户标识等。建议传入该参数。
UserDeviceId	String	否	3iJ1AY$oHcu7mC69	用户设备ID。不超过64字符。可用于标识具体设备，建议可以使用设备物理编号。建议传入该参数。
Test	Boolean	否	false	测试标识。为true表示使用测试授权，授权时长默认为30天；为false，则授权时长根据AuthYears进行授权。
AuthYears	Integer	否	1	当Test标识为false或空时，AuthYears必填，单位为年，范围为[1,100]，取值100表示永久授权。

返回数据

名称	类型	示例值	描述
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E	本次请求的ID。
AuthKey	String	auth.1KQMcnLd4m37LN2D0F0WCD-1qtQI$	可用于授权激活的key。授权key在30分钟内有效，且不可重复使用，建议在每次激活前重新获取。

示例

请求示例

```
https://cloudauth.aliyuncs.com/?Action=CreateAuthKey
&BizType=FACE_TEST
&UserDeviceId=3iJ1AY$oHcu7mC69
&Test=false
&AuthYears=1
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<CreateAuthKeyResponse>
      <Data>
            <AuthKey>auth.1KQMcnLd4m37LN2D0F0WCD-1qtQI$</AuthKey>
      </Data>
      <Success>true</Success>
      <Code>1</Code>
</CreateAuthKeyResponse>
```

`JSON` 格式

```
{
  "Data": {
    "AuthKey":"auth.1KQMcnLd4m37LN2D0F0WCD-1qtQI$"
  },
  "Success": true
}
```

错误码

访问错误中心查看更多错误码。

4.4. ModifyDeviceInfo

调用ModifyDeviceInfo更新设备相关信息，比如设备授权有效期延长、更新业务方自定义的业务标识和设备ID等。适用于设备有效期续期等场景。

请求方法：支持以HTTPS POST和GET方法发送请求。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ModifyDeviceInfo	要执行的操作。取值：**ModifyDeviceInfo**。
DeviceId	String	是	wd.6ziUffspAeW5FVYbaqmexR-1qwNjM	实人认证服务端为接入方设备生成的唯一ID，只有在设备成功激活后才会生成，该ID可通过离线人脸识别SDK里的getLicenseExtraInfo函数获得。
UserDeviceId	String	否	3iJ1AY$oHcu7mC69	更新的用户设备ID。不超过64字符，由接入方自定义，可用于标识具体设备，建议可以使用设备物理编号。

名称	类型	是否必选	示例值	描述
BizType	String	否	FACE_TEST	更新的业务类型。不超过64字符。由接入方自定义，可用于对具体业务进行备注，例如可以取接入方不同的人脸使用场景，或者待交付的客户标识等。
Duration	String	否	1	延长设备到期的时间。单位：年，范围为[1,100]，取值100表示永久使用。一年按365天计算。
ExpiredDay	String	否	20190401	当前设备的失效时间。失效日期不正确（和实人认证服务端记录的失效时间相差超过一周），则报错。 **说明** 该失效时间可以通过DescribeDeviceInfo接口查询获得，失效日期不正确则报错，这个校验是为了确保业务方不因为一些误操作导致重复激活一台设备，误消耗授权。

返回数据

名称	类型	示例值	描述
RequestId	String	473469C7-AA6F-4DC5-B3DB-A3DC0DE3C83E	本次请求的ID。
DeviceId	String	wd.6ziUffspAeW5FVYbaqmexR-1qwNjM	对应于请求参数中的DeviceId。
UserDeviceId	String	3iJ1AY$oHcu7mC69	对应于请求参数中的UserDeviceId。
BizType	String	FACE_TEST	对应于请求参数中的BizType。
BeginDay	String	20190401	若请求参数中Duration不为空，则该字段表示延长设备有效期后，授权开始时间。Duration的1年按365天计算。示例：20180101。
ExpiredDay	String	20200330	若请求参数中Duration不为空，则该字段表示延长设备有效期后，授权到期时间。Duration的1年按365天计算。示例：20180101。

示例

请求示例

```
https://cloudauth.aliyuncs.com/?Action=ModifyDeviceInfo
&DeviceId=wd.6ziUffspAeW5FVYbaqmexR-1qwNjM
&UserDeviceId=3iJ1AY$oHcu7mC69
&BizType=FACE_TEST
&Duration=1
&ExpiredDay=20190401
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<ModifyDeviceInfoResponse>
      <Data>
            <DeviceId>wd.6ziUffspAeW5FVYbaqmexR-1qwNjM</DeviceId>
            <UserDeviceId>3iJ1AY$oHcu7mC69</UserDeviceId>
            <BizType>FACE_TEST</BizType>
            <BeginDay>20190401</BeginDay>
            <ExpiredDay>20200330</ExpiredDay>
      </Data>
      <Success>true</Success>
      <Code>1</Code>
</ModifyDeviceInfoResponse>
```

`JSON` 格式

```
{
  "Data": {
    "DeviceId":"wd.6ziUffspAeW5FVYbaqmexR-1qwNjM",
    "UserDeviceId":"3iJ1AY$oHcu7mC69",
    "BizType":"FACE_TEST",
    "BeginDay":"20190401",
    "ExpiredDay":"20200330"
  },
  "Success": true
}
```

错误码

访问错误中心查看更多错误码。

4.5. DescribeDeviceInfo

调用DescribeDeviceInfo查询设备相关信息，比如授权有效期、接入方自定义的业务标识和设备ID等。

请求方法：支持以HTTPS POST和GET方法发送请求。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeDeviceInfo	要执行的操作。取值：**DescribeDeviceInfo**。
PageSize	Integer	否	20	查询每页数目。
CurrentPage	Integer	否	1	当前查询页数。
DeviceId	String	否	wd.6ziUffspAeW5FVYbaqmexR-1qwNjM	实人认证服务端为接入方设备生成的唯一ID，只有在设备成功激活后才会生成，该ID可通过离线人脸识别SDK里的getLicenseExtraInfo函数获得。
BizType	String	否	FACE_TEST	业务类型。不超过64字符。
UserDeviceId	String	否	3iJ1AY$oHcu7mC69	不超过64字符，由接入方自定义，可用于标识具体设备。
ExpiredStartDay	String	否	20190401	查询的开始时间，即查询在ExpiredStartDay和ExpiredEndDay之间要过期的授权。
ExpiredEndDay	String	否	20200330	查询的结束时间。即查询在ExpiredStartDay和ExpiredEndDay之间要过期的授权。

返回数据

名称	类型	示例值	描述
RequestId	String	969434DF-926B-4997-9881-4DE94E39F805	本次请求的ID。
PageSize	Integer	20	每页数目。
CurrentPage	Integer	1	当前查询页数。
TotalCount	Integer	1	总数。

名称	类型	示例值	描述
DeviceInfoList	Array		设备信息数组。
DeviceId	String	wd.6ziUffspAeW5FVYbaqmexR-1qwNjM	对应于请求中的DeviceId。
UserDeviceId	String	3iJ1AY$oHcu7mC69	对应于请求中的UserDeviceId。
BizType	String	FACE_TEST	对应于请求中的BizType。
BeginDay	String	20180101	授权开始时间。
ExpiredDay	String	20180101	授权到期时间。

示例

请求示例

```
https://cloudauth.aliyuncs.com/?Action= DescribeDeviceInfo
&DeviceId=wd.6ziUffspAeW5FVYbaqmexR-1qwNjM
&BizType=FACE_TEST
&UserDeviceId=3iJ1AY$oHcu7mC69
&ExpiredStartDay=20190401
&ExpiredEndDay=20200330
&PageSize=20
&CurrentPage=1
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<DescribeDeviceInfoResponse>
         <PageSize>20</PageSize>
         <CurrentPage>1</CurrentPage>
         <TotalCount>1</TotalCount>
         <DeviceInfoList>
                   <DeviceInfo>
                            <DeviceId>wd.6ziUffspAeW5FVYbaqmexR-1qwNjM</DeviceId>
                            <UserDeviceId>3iJ1AY$oHcu7mC69</UserDeviceId>
                            <BizType>FACE_TEST</BizType>
                            <BeginDay>20190101</BeginDay>
                            <ExpiredDay>20200101</ExpiredDay>
                    </DeviceInfo>
               </DeviceInfoList>
        <RequestId>969434DF-926B-4997-9881-4DE94E39F805</RequestId>
</DescribeDeviceInfoResponse>
```

`JSON` 格式

```
{
    "TotalCount": 0,
    "PageSize": 20,
    "RequestId": "969434DF-926B-4997-9881-4DE94E39F805",
    "DeviceInfoList": {
        "DeviceInfo": [
            "DeviceId" : "wd.6ziUffspAeW5FVYbaqmexR-1qwNjM",
            "UserDeviceId": "3iJ1AY$oHcu7mC69",
            "BizType": "FACE_TEST",
            "BeginDay": "20190101",
            "ExpiredDay": "20200101"
        ]
    },
    "CurrentPage": 1
}
```

错误码

访问错误中心查看更多错误码。