

# 人脸识别服务

## SDK 参考

文档版本

01

发布日期

2024-03-05



版权所有 © 华为技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

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

---

# 目录

---

<b>1 简介.....</b>	<b>1</b>
<b>2 Java SDK.....</b>	<b>3</b>
2.1 入门指导.....	3
2.2 SDK 接口示例.....	4
2.3 SDK & API 对应关系.....	4

# 1 简介

人脸识别服务软件开发工具包（FRS SDK）是对人脸识别服务提供的REST API进行的封装，以简化用户的开发工作。FRS SDK目前支持Java版本。

## 接口与 API 对应关系

人脸识别接口与API对应关系请参见[表1-1](#)。

表 1-1 接口与 API 对应关系表

接口		API
人脸检测		POST /v2/{project_id}/face-detect
人脸比对		POST /v2/{project_id}/face-compare
人脸搜索		POST /v2/{project_id}/face-sets/{face_set_name}/search
活体检测	动作活体检测	POST /v1/{project_id}/live-detect
人脸库资源管理	创建人脸库	POST /v2/{project_id}/face-sets
	查询所有人脸库	GET /v2/{project_id}/face-sets
	查询人脸库	GET /v2/{project_id}/face-sets/{face_set_name}
	删除人脸库	DELETE /v2/{project_id}/face-sets/{face_set_name}
人脸资源管理	添加人脸	POST /v2/{project_id}/face-sets/{face_set_name}/faces
	查询人脸	GET /v2/{project_id}/face-sets/{face_set_name}/faces?offset=xxx&limit=xxx
	更新人脸	PUT /v2/{project_id}/face-sets/{face_set_name}/faces

接口		API
	删除人脸	DELETE /v2/{project_id}/face-sets/{face_set_name}/faces?field_name=field_value
	批量删除人脸	DELETE /v2/{project_id}/face-sets/{face_set_name}/faces/batch

# 2 Java SDK

## 2.1 入门指导

### 环境准备

1. 开通人脸识别服务：登录[人脸识别管理控制台](#)，选择对应的子服务，单击右侧的“开通服务”。  
服务开通一次即可，后续使用时无需再开通。
2. 获取人脸识别服务的终端节点（Endpoint），请参见[地区和终端节点](#)。
3. 获取并了解人脸识别API，请参见《[人脸识别API参考](#)》。
4. 获取华为云账号的AK/SK。  
如果之前没有生成过AK/SK，可登录“我的凭证”“界面，选择“管理访问密钥 > 新增访问密钥”来获取。
5. 已经安装好JDK，Java SDK适用于JDK1.8+。

#### 说明

- 人脸识别服务提供的SDK暂不支持访问重试，如果访问异常，SDK会直接返回本次的请求结果，所以需要客户端自行建立重试机制。
- 安卓系统使用人脸识别的Java SDK时，不支持主线程直接调用，建议使用子线程。
- 使用java SDK中的Demo时，因readAllBytes()为JDK1.9+中方法，建议您使用JDK1.9+。
- 关于SDK中Demo示例的使用方法，请参考demoV2()。

### SDK 获取和安装

以Eclipse为例，创建Java工程后，执行以下步骤，导入JAR文件到新建的工程中：

1. 下载[SDK的JAR文件](#)。
2. 将下载的JAR文件复制到工程文件夹中。
3. 在Eclipse中打开工程，右键单击该项工程，选择“Properties”。
4. 在弹出的对话框中，单击“Java Build Path”，然后在“Libraries”页签下单击“Add JARs”，添加下载的JAR文件。
5. 单击OK。

本文中涉及的服务使用同一个JAR文件。

## 2.2 SDK 接口示例

### SDK 示例

#### 1. 构造服务客户端

```
String ak = System.getenv("HUAWEICLOUD_SDK_AK");
String sk = System.getenv("HUAWEICLOUD_SDK_SK");
String projectId = "projectId";
//Proxy info, 当您需要使用代理才可以访问到服务时, 可在此处增加代理信息
ProxyHostInfo proxyHostInfo = new ProxyHostInfo("proxy",
8080,
"UserName",
"Password"
);
//Service info
String url = "https://face.ap-southeast-1.myhuaweicloud.com";
String region = "ap-southeast-1";
AuthInfo authInfo = new AuthInfo(url, region, ak, sk);
FrsClient frsClient = new FrsClient(authInfo, projectId);
//FrsClient frsClient = new FrsClient(authInfo, projectId, proxyHostInfo);//当您需要代理才可以访问到服务时, 在第三个参数增加代理信息即可
```

#### 2. 创建人脸集

```
CreateExternalFields createExternalFields = new CreateExternalFields();
createExternalFields.addField("testInteger", FieldType.INTEGER);
CreateFaceSetResult createFaceSetResult =
frsClient.getV2().getFaceSetService().createFaceSet(faceSetName, faceSetCapacity,
createExternalFields);
```

#### 3. 添加人脸

```
String imageId = "image1";
File image = new File("data/face.jpg");
byte[] fileData = FileUtils.readFileToByteArray(image);
String imageBase64 = Base64.encodeBase64String(fileData);
AddExternalFields addExternalFields = new AddExternalFields();
addExternalFields.addField("testInteger", 123);
AddFaceResult addFaceResult = frsClient.getV2().getFaceService().addFaceByBase64(faceSetName,
imageId, imageBase64, addExternalFields);
```

#### 4. 调用人脸检测服务

```
String obsUrl = "/bucket/face.jpg";
String attr = "1,2";
DetectFaceResult detectFaceResult = frsClient.getV2().getDetectService().detectFaceByObsUrl(obsUrl,
attr);
```

#### 说明

示例中ak、sk即访问密钥，请登录[我的凭证](#)界面，选择“管理访问密钥 > 新增访问密钥”获取。

projectId为项目ID，获取方式请参见[获取项目ID](#)。

## 2.3 SDK & API 对应关系

接口	方法	API
CompareService	compareFaceByBase64	POST /v2/{project_id}/face-compare
	compareFaceByFile	

接口	方法	API
	compareFaceByObsUrl	
DetectService	detectFaceByBase64	POST /v2/{project_id}/face-detect
	detectFaceByFile	
	detectFaceByObsUrl	
FaceService	addFaceByBase64	POST /v2/{project_id}/face-sets/{face_set_name}/faces
	addFaceByFile	
	addFaceByObsUrl	
	getFaces	GET /v2/{project_id}/face-sets/{face_set_name}/faces?offset=xxx&limit=xxx
	getFace	GET /v2/{project_id}/face-sets/{face_set_name}/faces?face_id={face_id}
	deleteFaceByFacelId	DELETE /v2/{project_id}/face-sets/{face_set_name}/faces ?external_image_id=imageID
	deleteFaceByExternalImageId	DELETE /v2/{project_id}/face-sets/{face_set_name}/faces ?face_id=faceID
	deleteFaceByFieldId	DELETE /v2/{project_id}/face-sets/{face_set_name}/faces ?id=home
FaceSetService	createFaceSet	POST /v2/{project_id}/face-sets
	getAllFaceSets	GET /v2/{project_id}/face-sets
	getFaceSet	GET /v2/{project_id}/face-sets/{face_set_name}
	deleteFaceSet	DELETE /v2/{project_id}/face-sets/{face_set_name}
LiveDetectService	liveDetectByBase64	POST /v2/{project_id}/live-detect
	liveDetectByFile	
	liveDetectByObsUrl	
SearchService	searchFaceByBase64	POST /v2/{project_id}/face-sets/{face_set_name}/search
	searchFaceByFile	
	searchFaceByObsUrl	
	searchFaceByFacelId	