



人脸识别

## SDK 参考

文档版本 02

发布日期 2019-01-30

版权所有 © 华为技术有限公司 2019。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默示的声明或保证。

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 Java SDK</b> .....	<b>1</b>
1.1 入门指导.....	1
1.2 SDK 接口示例.....	3
1.3 SDK & API 对应关系.....	3
<b>2 Python SDK</b> .....	<b>5</b>
2.1 入门指导.....	5
2.2 SDK 接口示例.....	6
2.3 SDK & API 对应关系.....	6
<b>3 Go SDK</b> .....	<b>9</b>
3.1 入门指导.....	9
3.2 SDK 接口示例.....	10
3.3 SDK & API 对应关系.....	11
<b>4 CSharp SDK</b> .....	<b>13</b>
4.1 入门指导.....	13
4.2 SDK 接口示例.....	14
4.3 SDK & API 对应关系.....	14
<b>A 修订记录</b> .....	<b>16</b>

# 1 Java SDK

## 1.1 入门指导

### 环境准备

1. 获取API相关文档，人脸识别服务的API请见《[人脸识别API参考](#)》。  
访问[华为云开放API的产品界面](#)，可获取其他云服务的API参考。
2. 获取人脸识别服务的Endpoint，请参见[地区和终端节点](#)。
3. 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。
4. 获取华为云账号的AK/SK。  
如果之前没有生成过AK/SK，可登录华为云“[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。

### 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。

## 开始使用

设置参数，初始化SDK客户端，然后调用SDK访问服务的API。

```
//设置认证参数
String ak = "ak";
String sk = "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.cn-north-1.myhuaweicloud.com";
String region = "cn-north-1";

AuthInfo authInfo = new AuthInfo(url, region, ak, sk);

try (FrsClient frsClient = new FrsClient(authInfo, projectId)) {
//try (FrsClient frsClient = new FrsClient(authInfo, projectId, proxyHostInfo)) { //需使用代理时
//调用前请在管理控制台开通人脸识别服务
//创建人脸集合
String faceSetName = "faceSetName";
int faceSetCapacity = 10000;
//创建人脸集合时, 自定义属性
CreateExternalFields createExternalFields = new CreateExternalFields();
createExternalFields.addField("testInteger", FieldType.INTEGER);
CreateFaceSetResult createFaceSetResult = frsClient.getFaceSetService().createFaceSet(faceSetName,
faceSetCapacity, createExternalFields);
System.out.println(createFaceSetResult.toJSONString());

//获取人脸集合
GetAllFaceSetResult getAllFaceSetsResult = frsClient.getFaceSetService().getAllFaceSet();
System.out.println(getAllFaceSetResult.toJSONString());

//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.getFaceService().addFaceByBase64(faceSetName, imageId,
imageBase64, addExternalFields);
System.out.println(addFaceResult.toJSONString());

//人脸对比
String image1Path = "data/face1.jpg";
String image2Path = "data/face2.jpg";
CompareFaceResult compareFaceResult = frsClient.getCompareService().compareFaceByFile(image1Path,
image2Path);
System.out.println(compareFaceResult.toJSONString());

//人脸检测
String obsUrl = "/bucket/face.jpg";
String attr = "0,1,2,3,4,5";
DetectFaceResult detectFaceResult = frsClient.getDetectService().detectFaceByObsUrl(obsUrl, attr);
System.out.println(detectFaceResult.toJSONString());
}
```

## 1.2 SDK 接口示例

### 1. 构造服务客户端

```
String ak = "ak";
String sk = "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.cn-north-1.myhuaweicloud.com";
String region = "cn-north-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.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.getFaceService().addFaceByBase64(faceSetName,
imageId, imageBase64, addExternalFields);
```

### 4. 调用人脸检测服务

```
String obsUrl = "/bucket/face.jpg";
String attr = "0,1,2,3,4,5";
DetectFaceResult detectFaceResult = frsClient.getDetectService().detectFaceByObsUrl(obsUrl,
attr);
```

## 1.3 SDK & API 对应关系

接口	方法	API
CompareService	compareFaceByBase64	POST /v1/{project_id}/face-compare
	compareFaceByFile	
	compareFaceByObsUrl	
DetectService	detectFaceByBase64	POST /v1/{project_id}/face-detect
	detectFaceByFile	
	detectFaceByObsUrl	

接口	方法	API
FaceService	addFaceByBase64	POST /v1/{project_id}/face-sets/ {face_set_name}/faces
	addFaceByFile	
	addFaceByObsUrl	
	getFaces	GET /v1/{project_id}/face-sets/ {face_set_name}/faces?offset=xxx&limit=xxx
	getFace	GET /v1/{project_id}/face-sets/ {face_set_name}/faces?face_id={face_id}
	deleteFaceByFaceId	DELETE /v1/{project_id}/face-sets/ {face_set_name}/faces ? external_image_id=imageID
	deleteFaceByExternalImageId	DELETE /v1/{project_id}/face-sets/ {face_set_name}/faces ?face_id=faceID
	deleteFaceByFieldId	DELETE /v1/{project_id}/face-sets/ {face_set_name}/faces ?id=home
FaceSetService	createFaceSet	POST /v1/{project_id}/face-sets
	getAllFaceSets	GET /v1/{project_id}/face-sets
	getFaceSet	GET /v1/{project_id}/face-sets/{face_set_name}
	deleteFaceSet	DELETE /v1/{project_id}/face-sets/ {face_set_name}
LiveDetectService	liveDetectByBase64	POST /v1/{project_id}/live-detect
	liveDetectByFile	
	liveDetectByObsUrl	
SearchService	searchFaceByBase64	POST /v1/{project_id}/face-sets/ {face_set_name}/search
	searchFaceByFile	
	searchFaceByObsUrl	
	searchFaceByFaceId	

# 2 Python SDK

## 2.1 入门指导

### 环境准备

1. 获取API相关文档，人脸识别服务的API请见《[人脸识别API参考](#)》。  
访问[华为云开放API的产品界面](#)，可获取其他云服务的API参考。
2. 获取人脸识别服务的Endpoint，请参见[地区和终端节点](#)。
3. 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。
4. 获取华为云账号的AK/SK。  
如果之前没有生成过AK/SK，可登录华为云“[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Python环境，Python SDK适用于Python2.7、Python3.6。

#### 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。

### SDK 获取和安装

1. 下载人脸识别服务的[Python SDK](#)并解压。
2. 下载和安装[setuptools](#)工具。
3. 命令行切换到Python SDK解压目录。
4. 在SDK目录中，执行Python `setup.py install`命令安装Python SDK到开发环境，或者将.py文件直接引入项目。

#### 说明

人脸识别Python编程时要引用frsclient文件夹下所有的文件：`from frsclient import *`。

### 开始使用

设置参数，初始化SDK客户端，然后调用SDK访问服务的API。



```

ak = "ak"
sk = "sk"
project_id = "projectId"
end_point = "https://face.cn-north-1.myhuaweicloud.com"
proxy = {"http": "http://127.0.0.1:1234",
        "https": "http://127.0.0.1:1234"}

auth_info = AuthInfo(ak=ak, sk=sk, end_point=end_point)
frs_client = FrsClient(auth_info=auth_info, project_id=project_id)
# frs_client = FrsClient(auth_info=auth_info, project_id=project_id, proxies=proxy) # 需要使用代理
访问云服务时配置

//人脸对比
image1_path = ""
image2_path = ""
cs = frs_client.get_compare_service()
res = cs.compare_face_by_file(image1_path, image2_path)
print res.get_eval_result()

//活体检测
lds = frs_client.get_live_detect_service()
video_file = "data/video.mp4"
actions = "1"
res = lds.live_detect_by_file(video_file, actions)
print res.get_eval_result()

```

## 2.2 SDK 接口示例

### 1. 构造服务客户端

```

ak = "ak"
sk = "sk"
project_id = "projectId"
end_point = "https://face.cn-north-1.myhuaweicloud.com"
proxy = {"http": "http://127.0.0.1:1234",
        "https": "http://127.0.0.1:1234"}
auth_info = AuthInfo(ak=ak, sk=sk, end_point=end_point)
frs_client = FrsClient(auth_info=auth_info, project_id=project_id)
# frs_client = FrsClient(auth_info=auth_info, project_id=project_id, proxies=proxy) # 当您需
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

```

fss = frs_client.get_face_set_service()
external_fields = "{\"timestamp\": {\"type\": \"long\"}, \"id\": {\"type\": \"string\"}, \"number\": {\"type\": \"integer\"}}\"
ret = fss.create_face_set("faceSetName", 10000, external_fields)

```

### 3. 添加人脸

```

fs = frs_client.get_face_service()
external_fields = "{\"timestamp\": 12, \"id\": \"home\"}"
res = fs.add_face_by_obs_url("faceSetName", "/obs/image.jpg", "externalImageId",
external_fields)

```

### 4. 人脸检测

```

ds = frs.get_detect_service()
res = ds.detect_face_by_file("imagePath", "0,1,2,3")

```

## 2.3 SDK & API 对应关系

接口	方法	API
CompareService	compare_face_by_base64	POST /v1/{project_id}/face-compare

接口	方法	API
	compare_face_by_file	
	compare_face_by_obs_url	
DetectService	detect_face_by_base64	POST /v1/{project_id}/face-detect
	detect_face_by_file	
	detect_face_by_obs_url	
FaceService	add_face_by_base64	POST /v1/{project_id}/face-sets/{face_set_name}/faces
	add_face_by_file	
	add_face_by_obs_url	
	get_faces	GET /v1/{project_id}/face-sets/{face_set_name}/faces?offset=xxx&limit=xxx
	get_face	GET /v1/{project_id}/face-sets/{face_set_name}/faces?face_id={face_id}
	delete_face_by_face_id	DELETE /v1/{project_id}/face-sets/{face_set_name}/faces ? external_image_id=imageID
	delete_face_by_external_image_id	DELETE /v1/{project_id}/face-sets/{face_set_name}/faces ?face_id=faceID
	delete_face_by_field_id	DELETE /v1/{project_id}/face-sets/{face_set_name}/faces ?id=home
FaceSetService	create_face_set	POST /v1/{project_id}/face-sets
	get_all_face_sets	GET /v1/{project_id}/face-sets
	get_face_set	GET /v1/{project_id}/face-sets/{face_set_name}
	delete_face_set	DELETE /v1/{project_id}/face-sets/{face_set_name}
LiveDetectService	live_detect_by_base64	POST /v1/{project_id}/live-detect
	live_detect_by_file	
	live_detect_by_obs_url	
SearchService	search_face_by_base64	POST /v1/{project_id}/face-sets/{face_set_name}/search

接口	方法	API
	search_face_by_file	
	search_face_by_obs_url	
	search_face_by_face_id	

# 3 Go SDK

## 3.1 入门指导

### 环境准备

1. 获取API相关文档，人脸识别服务的API请见《[人脸识别API参考](#)》。  
访问[华为云开放API的产品界面](#)，可获取其他云服务的API参考。
2. 获取人脸识别服务的Endpoint，请参见[地区和终端节点](#)。
3. 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。
4. 获取华为云账号的AK/SK。  
如果之前没有生成过AK/SK，可登录华为云“[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Go，并设置好Go语言环境变量。

#### 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。

### SDK 获取和安装

1. 下载人脸识别服务的[Go SDK](#)并解压。
2. 使用Eclipse新建Go工程，将Go SDK文件夹下的全部内容拷贝至新建工程中的文件夹下。
3. 右键单击建好的Go工程，运行弹出菜单中的“Build Project”命令编译工程并等待编译完成。

### 开始使用

设置参数，初始化SDK客户端，然后调用SDK访问服务的API。

```
authInfo := param.AuthInfo{EndPoint: "https://face.cn-north-1.myhuaweicloud.com", Ak: "ak", Sk: "sk"}
frsClient := service.NewFrsClient(&authInfo, "projectId")
```

```
//proxyInfo := param.ProxyHostInfo{Proxy: "http://127.0.0.1:8080"}
//frsClient := service.NewFrsClientWithProxy(&authInfo, "projectId", &proxyInfo)//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

//人脸对比
compareResult, err := frsClient.GetCompareService().CompareFaceByFile("image1Path", "image2Path")
if nil != err {
    fmt.Println(err.Error())
} else {
    fmt.Println(compareResult)
}

//人脸检测
detectResult, err2 := frsClient.GetDetectService().DetectFaceByObsUrlWithAttr("obsUrl", "0,1,2")
if nil != err2 {
    fmt.Println(err2.Error())
} else {
    fmt.Println(detectResult)
}
```

## 3.2 SDK 接口示例

### 1. 构造服务客户端

```
authInfo := param.AuthInfo{EndPoint: "https://face.cn-north-1.myhuaweicloud.com", Ak: "ak",
Sk: "sk"}
frsClient := service.NewFrsClient(&authInfo, "projectId")
//proxyInfo := param.ProxyHostInfo{Proxy: "http://127.0.0.1:8080"}
//frsClient := service.NewFrsClientWithProxy(&authInfo, "projectId", &proxyInfo)//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

```
externalFields := param.NewCreateExternalFields()
externalFields.AddField("testInteger", param.INTEGER)
result, err := frsClient.GetFaceSetService().CreateFaceSetWithExtFields("FaceSetName",
10000, externalFields)
if nil != err {
    fmt.Println(err.Error())
} else {
    fmt.Println(result)
}
```

### 3. 添加人脸

```
externalFields := param.NewAddExternalFields()
externalFields.AddField("testInteger", 123)
result, err := frsClient.GetFaceService().AddFaceByFileFull("faceSetName",
"externalImageId", "imagePath", externalFields)
if nil != err {
    fmt.Println(err.Error())
} else {
    fmt.Println(result)
}
```

### 4. 人脸检测

```
detectResult, err := frsClient.GetDetectService().DetectFaceByObsUrlWithAttr("/obs/
image.jpg", "0,1,2")
if nil != err {
    fmt.Println(err.Error())
} else {
    fmt.Println(detectResult)
}
```

### 3.3 SDK & API 对应关系

接口	方法	API
CompareService	CompareFaceByBase64	POST /v1/{project_id}/face-compare
	CompareFaceByFile	
	CompareFaceByUrl	
DetectService	DetectFaceByBase64	POST /v1/{project_id}/face-detect
	DetectFaceByFile	
	DetectFaceByUrl	
FaceService	AddFaceByBase64	POST /v1/{project_id}/face-sets/{face_set_name}/faces
	AddFaceByFile	
	AddFaceByUrl	
	GetFaces	GET /v1/{project_id}/face-sets/{face_set_name}/faces?offset=xxx&limit=xxx
	GetFace	GET /v1/{project_id}/face-sets/{face_set_name}/faces?face_id={face_id}
	DeleteFaceByFaceId	DELETE /v1/{project_id}/face-sets/{face_set_name}/faces ? external_image_id=imageID
	DeleteFaceByExternalImageId	DELETE /v1/{project_id}/face-sets/{face_set_name}/faces ?face_id=faceID
	DeleteFaceByFieldId	DELETE /v1/{project_id}/face-sets/{face_set_name}/faces ?id=home
FaceSetService	CreateFaceSet	POST /v1/{project_id}/face-sets
	GetAllFaceSets	GET /v1/{project_id}/face-sets
	GetFaceSet	GET /v1/{project_id}/face-sets/{face_set_name}
	DeleteFaceSet	DELETE /v1/{project_id}/face-sets/{face_set_name}
LiveDetectService	LiveDetectByBase64	POST /v1/{project_id}/live-detect
	LiveDetectByFile	

接口	方法	API
	LiveDetectByObsUrl	
SearchService	SearchFaceByBase64	POST /v1/{project_id}/face-sets/{face_set_name}/search
	SearchFaceByFile	
	SearchFaceByObsUrl	
	SearchFaceByFaceId	

# 4 CSharp SDK

## 4.1 入门指导

### 环境准备

1. 获取API相关文档，人脸识别服务的API请见《[人脸识别API参考](#)》。  
访问[华为云开放API的产品界面](#)，可获取其他云服务的API参考。
2. 获取人脸识别服务的Endpoint，请参见[地区和终端节点](#)。
3. 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。
4. 获取华为云账号的AK/SK。  
如果之前没有生成过AK/SK，可登录华为云“[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.net开发环境。

#### 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。

### SDK 获取和安装

下载[CSharp SDK的源码](#)。

### 开始使用

设置参数，初始化SDK客户端，然后调用SDK访问服务的API。

```
AuthInfo authInfo = new AuthInfo("https://face.cn-north-1.myhuaweicloud.com", "ak", "sk");
FrsClient frsClient = new FrsClient(authInfo, "project id");
//ProxyHostInfo proxy := new ProxyHostInfo("host", 8080);
//FrsClient frsClient = new FrsClient (authInfo, "projectId", proxy); //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

//人脸对比
CompareFaceResult compareFaceResult = frsClient.GetCompareService().CompareFaceByUrl("/obs/1.jpg", "/obs/1.jpg");
```



```

Console.WriteLine(compareFaceResult.GetJsonString());
compareFaceResult = frsClient.GetCompareService().CompareFaceByFile("/resource/1.jpeg", "/resource/1.jpeg");
Console.WriteLine(compareFaceResult.GetJsonString());
//人脸检测
DetectFaceResult detectFaceResult = frsClient.GetDetectService().DetectFaceByUrl("/obs/1.jpg", "0,1,2");
Console.WriteLine(detectFaceResult.GetJsonString());
DetectFaceResult detectFaceResult2 = frsClient.GetDetectService().DetectFaceByFile("/resource/1.jpeg", "1,2");
Console.WriteLine(detectFaceResult2.GetJsonString());
    
```

## 4.2 SDK 接口示例

### 1. 构造服务客户端

```

AuthInfo authInfo = new AuthInfo("https://face.cn-north-1.myhuaweicloud.com", "ak", "sk");
FrsClient frsClient = new FrsClient(authInfo, "project id");
//ProxyHostInfo proxy := new ProxyHostInfo("host", 8080);
//FrsClient frsClient = new FrsClient(authInfo, "projectId", proxy);//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("testInt", FieldType.INTEGER);
createExternalFields.AddField("testStr", FieldType.STRING);
CreateFaceSetResult createFaceSetResult = frsClient.GetFaceSetService().CreateFaceSet("face_set_name_test", 10000, createExternalFields);
Console.WriteLine(createFaceSetResult.GetJsonString());
    
```

### 3. 添加人脸

```

AddExternalFields addExternalFields = new AddExternalFields();
addExternalFields.AddField("testInt", 1);
addExternalFields.AddField("testStr", "str");
AddFaceResult addFaceResult = frsClient.GetFaceService().AddFaceByFile("face_set_name_test", "/resource/3.png", addExternalFields);
Console.WriteLine(addFaceResult.GetJsonString());
    
```

### 4. 人脸检测

```

DetectFaceResult detectFaceResult = frsClient.GetDetectService().DetectFaceByUrl("/obs/1.jpg", "0,1,2");
Console.WriteLine(detectFaceResult.GetJsonString());
DetectFaceResult detectFaceResult2 = frsClient.GetDetectService().DetectFaceByFile("/resource/1.jpeg", "1,2");
Console.WriteLine(detectFaceResult2.GetJsonString());
    
```

## 4.3 SDK & API 对应关系

接口	方法	API
CompareService	CompareFaceByBase64	POST /v1/{project_id}/face-compare
	CompareFaceByFile	
	CompareFaceByUrl	
DetectService	DetectFaceByBase64	POST /v1/{project_id}/face-detect
	DetectFaceByFile	
	DetectFaceByUrl	

接口	方法	API
FaceService	AddFaceByBase64	POST /v1/{project_id}/face-sets/ {face_set_name}/faces
	AddFaceByFile	
	AddFaceByUrl	
	GetFaces	GET /v1/{project_id}/face-sets/ {face_set_name}/faces? offset=xxx&limit=xxx
	GetFace	GET /v1/{project_id}/face-sets/ {face_set_name}/faces?face_id={face_id}
	DeleteFaceByFaceId	DELETE /v1/{project_id}/face-sets/ {face_set_name}/faces ? external_image_id=imageID
	DeleteFaceByExternalImageId	DELETE /v1/{project_id}/face-sets/ {face_set_name}/faces ?face_id=faceID
DeleteFaceByFieldId	DELETE /v1/{project_id}/face-sets/ {face_set_name}/faces ?id=home	
FaceSetService	CreateFaceSet	POST /v1/{project_id}/face-sets
	GetAllFaceSets	GET /v1/{project_id}/face-sets
	GetFaceSet	GET /v1/{project_id}/face-sets/ {face_set_name}
	DeleteFaceSet	DELETE /v1/{project_id}/face-sets/ {face_set_name}
LiveDetectService	LiveDetectByBase64	POST /v1/{project_id}/live-detect
	LiveDetectByFile	
	LiveDetectByUrl	
SearchService	SearchFaceByBase64	POST /v1/{project_id}/face-sets/ {face_set_name}/search
	SearchFaceByFile	
	SearchFaceByObsUrl	
	SearchFaceByFaceId	

# A 修订记录

发布日期	修改说明
2019-01-30	第三次正式发布，Python SDK样例与SDK代码风格同步。 修改如下章节： <ul style="list-style-type: none"><li>● <a href="#">入门指导</a></li><li>● <a href="#">SDK接口示例</a></li><li>● <a href="#">SDK &amp; API对应关系</a></li></ul>
2018-12-28	第二次正式发布，新增 <a href="#">CSharp SDK</a> 。
2018-12-12	第一次正式发布。