

图像识别

# SDK 参考

文档版本 01  
发布日期 2024-12-05



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。

# 华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <https://www.huawei.com>

客户服务邮箱： [support@huawei.com](mailto:support@huawei.com)

客户服务电话： 4008302118

# 安全声明

## 漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

---

# 目录

---

<b>1 图像识别 SDK 简介</b> .....	<b>1</b>
<b>2 申请服务</b> .....	<b>2</b>
<b>3 获取认证信息</b> .....	<b>3</b>
<b>4 准备环境</b> .....	<b>4</b>
<b>5 SDK 列表及指导文档（通用版）</b> .....	<b>5</b>
<b>6 使用 SDK（Java）</b> .....	<b>6</b>
6.1 Java 开发环境配置.....	6
6.2 媒资图像标签示例.....	7
6.3 名人识别示例.....	10
6.4 主体识别示例.....	11
6.5 翻拍识别示例.....	12
6.6 图像标签示例.....	14
<b>7 使用 SDK（Python）</b> .....	<b>16</b>
7.1 Python 开发环境配置.....	16
7.2 媒资图像标签示例.....	18
7.3 名人识别示例.....	19
7.4 主体识别示例.....	20
7.5 翻拍识别示例.....	21
7.6 图像标签示例.....	22
<b>8 使用 SDK（.NET）</b> .....	<b>23</b>
8.1 .NET 开发环境配置.....	23
8.2 媒资图像标签示例.....	24
8.3 名人识别示例.....	25
8.4 主体识别示例.....	27
8.5 翻拍识别示例.....	28
8.6 图像标签示例.....	30
<b>9 使用 SDK（GO）</b> .....	<b>32</b>
9.1 GO 开发环境配置.....	32
9.2 媒资图像标签示例.....	32
9.3 名人识别示例.....	34

---

9.4 主体识别示例.....	35
9.5 翻拍识别示例.....	36
9.6 图像标签示例.....	37
<b>10 使用 SDK ( PHP ) .....</b>	<b>38</b>
10.1 PHP 开发环境配置.....	38
10.2 媒资图像标签示例.....	41
10.3 翻拍识别示例.....	43
10.4 图像标签示例.....	44
<b>11 使用 SDK ( C++ ) .....</b>	<b>46</b>
11.1 C++开发环境配置.....	46
11.2 媒资图像标签示例.....	50
11.3 翻拍识别示例.....	51
11.4 图像标签示例.....	52

# 1 图像识别 SDK 简介

## 图像识别概述

图像识别（Image Recognition），是指利用计算机对图像进行分析和理解，以识别各种不同模式的目标和对象的技术，包括媒资图像标签，名人识别，主体识别，翻拍识别，图像标签等。

图像识别以开放API（Application Programming Interface，应用程序编程接口）的方式提供给用户，用户通过实时访问和调用API获取推理结果，帮助用户自动采集关键数据，打造智能化业务系统，提升业务效率。

## SDK 概述

图像识别软件开发工具包（Image Recognition Software Development Kit，简称 Image SDK）是对图像识别提供的REST API进行的封装，以简化用户的开发工作。用户通过添加依赖或下载的方式调用API即可实现使用图像识别业务能力的目的。

## 接口与 API 对应关系

图像识别接口与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
媒资图像标签	POST /v2/{project_id}/image/media-tagging
名人识别	POST /v2/{project_id}/image/celebrity-recognition
主体识别	POST /v3/{project_id}/image/main-object-detection
翻拍识别	POST /v2/{project_id}/image/recapture-detect
图像标签	POST /v2/{project_id}/image/tagging

# 2 申请服务

---

申请图像识别服务的具体操作步骤请参见《图像识别API参考》的“[如何调用API > 申请服务](#)”章节。

# 3 获取认证信息

---

使用服务API需要进行认证，具体方法如下：

- 步骤1** 注册并登录华为云管理控制台。
- 步骤2** 鼠标移动至用户名处，在下拉列表中单击“我的凭证”。
- 步骤3** 选择“访问密钥”页签，单击“新增访问密钥”按钮。
- 步骤4** 通过邮箱或者手机进行验证，输入对应的验证码。
- 步骤5** 单击“确定”，下载认证账号的AK/SK，AK/SK数据会以本地文件的形式保存，请妥善保管。

----结束



# 4 准备环境

在使用图像识别SDK时，各语言需要准备的环境 参照[表4-1](#)。

**表 4-1** 开发环境

开发语言	准备项	说明
JAVA	安装JDK	JAVA环境开发配置，支持 Java JDK 1.8 及以上版本，推荐通过Maven 安装依赖的方式使用JAVA版本SDK。
PYTHON	安装python	python版本sdk支持python3.3及以上版本。
.NET	安装.NET	.NET Standard 2.0 及其以上版本或C# 4.0 及其以上版本。
GO	安装go	支持 go 1.14 及以上版本。
PHP	安装PHP	支持PHP 5.6 及以上版本。
C++	安装C++	支持 C++ 14 及以上版本，要求安装 CMake 3.10 及以上版本。

# 5 SDK 列表及指导文档（通用版）

**API Explorer**能根据需要动态生成SDK代码功能，降低您使用SDK的难度，推荐使用。

SDK中心 <https://sdkcenter.developer.huaweicloud.com/zh-cn>提供服务SDK依赖引入的方式。

**表5-1** 提供了各个云服务支持的SDK列表，您可以在GitHub仓库查看SDK更新历史、获取安装包以及查看指导文档进行配置。

**表 5-1** SDK 列表

编程语言	github地址	指导文档	视频指导
JAVA	<a href="#">huaweicloud-sdk-java-v3</a>	<a href="#">Java SDK使用指导</a>	<a href="#">Java SDK视频指导</a>
PYTHON	<a href="#">huaweicloud-sdk-python-v3</a>	<a href="#">Python SDK使用指导</a>	<a href="#">Python SDK视频指导</a>
.NET	<a href="#">huaweicloud-sdk-net-v3</a>	<a href="#">.Net SDK使用指导</a>	-
GO	<a href="#">huaweicloud-sdk-go-v3</a>	<a href="#">Go SDK使用指导</a>	-
PHP	<a href="#">huaweicloud-sdk-php-v3</a>	<a href="#">PHP SDK使用指导</a>	-
C++	<a href="#">huaweicloud-sdk-cpp-v3</a>	<a href="#">C++ SDK使用指导</a>	-

# 6 使用 SDK (Java)

## 6.1 Java 开发环境配置

### 环境配置

使用图像识别Java版本SDK包，要求的JDK版本必须高于JDK8版本，以下步骤以win7环境配置JDK8 64位为例，若已经下载JDK并配置好环境请忽略本章节。

1. 下载[JDK文件](#)。
2. 下载完成后按照提示安装，位置自选，比如安装到本地C:\Program Files\Java\jdk1.8.0\_131。
3. 配置Java环境变量：右键“计算机>属性>高级系统设置>环境变量”，进行如下操作。
  - a. 新建系统变量JAVA\_HOME，变量值为实际JDK安装位置。
  - b. 在Path中添加%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin（注意用英文分号分隔）。
  - c. 新建系统变量CLASSPATH，变量值为%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar。
4. 打开命令行窗口，输入“java -version”，显示如[图6-1](#)表示配置成功。

图 6-1 Java 版本信息

```
C:\>java -version
java version "1.8.0_131"
Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)
```

### SDK 获取和安装

JAVA版本图像识别的SDK推荐您通过 Maven 安装依赖的方式使用华为云 Java SDK:

**步骤1** 首先您需要在您的操作系统中 [下载](#) 并 [安装](#)Maven 。

**步骤2** 安装配置完成后，输入命令“mvn -v”，显示如下图 Maven版本信息 表示成功。

```
D:\>mvn -v
D:\>
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-11T00:41:47+08:00)
Maven home: D:\maven\apache-maven-3.3.9\bin\..
Java version: 1.8.0_262, vendor: Huawei Technologies Co., Ltd
Java home: D:\develop\jdk_1.8\jre
Default locale: zh_CN, platform encoding: GBK
OS name: "windows 10", version: "10.0", arch: "amd64", family: "dos"
```

**步骤3** Maven安装完成后, 可根据开发需要, 直接引入依赖到已有的Maven工程或先用开发工具创建Maven工程。

创建项目以idea开发工具为例(已有Maven 工程可跳过此步骤):

1. 打开IntelliJ IDEA 开发工具。
2. 单击File - New - project...
3. 在New Project弹窗单击-Maven-单击Next。
4. 输入GroupId和ArtifactId, 单击Next。
5. 输入Project name 和 Project location,单击Finish。

**步骤4** 在Maven 项目的 pom.xml 文件加入相应的依赖项即可。

以引入最新版本的图像识别SDK为例, 请在[SDK中心](#)获取最新的sdk包版本, 替换代码中版本。

```
<dependency>
  <groupId>com.huaweicloud.sdk</groupId>
  <artifactId>huaweicloud-sdk-image</artifactId>
  <version>3.1.8</version> ///此处需要您输入最新的sdk包版本
</dependency>
```

示例代码因需要输出结果更直观, 所以新增fastjson的依赖 (如接入不涉及此项, 可忽略并在示例代码中删除相关内容), 依赖如下:

```
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>fastjson</artifactId>
  <version>1.2.70</version>
</dependency>
```

----结束

## 6.2 媒资图像标签示例

本章节对媒资图像标签AK/SK方式使用SDK进行示例说明。

- 媒资图像标签示例代码只需将AK/SK信息替换为实际AK/SK, 代码中可以使用 ImageMediaTaggingReq中的withUrl或withImage方法配置图像信息 ( image和url参数二选一), 示例中以url为例, 配置完成后运行即可。
- endpoint是华为云各服务应用区域和各服务的终端节点, 详情请查看[地区和终端节点](#)。

1. 媒资图像标签调用示例代码如下:

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.image.v2.region.ImageRegion;
import com.huaweicloud.sdk.image.v2.*;
```

```
import com.huaweicloud.sdk.image.v2.model.*;

public class RunImageMediaTaggingSolution {

    public static void main(String[] args) {
        // 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全
        // 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK
        String ak = System.getenv("HUAWEICLOUD_SDK_AK");
        String sk = System.getenv("HUAWEICLOUD_SDK_SK");
        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        ImageClient client = ImageClient.newBuilder()
            .withCredential(auth)
            .withRegion(ImageRegion.valueOf("cn-north-4")) //此处替换为您开通服务的区域，详情请查看地区和终端节点
            .build();
        RunImageMediaTaggingRequest request = new RunImageMediaTaggingRequest();
        ImageMediaTaggingReq body = new ImageMediaTaggingReq();
        body.withThreshold(10f);
        body.withLanguage("zh");
        body.withUrl("https://XXX.jpg"); //此处替换为公网可以访问的图片地址
        request.withBody(body);
        try {
            RunImageMediaTaggingResponse response = client.runImageMediaTagging(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

2. 执行示例代码文件，控制台输出媒资图像标签结果即表示程序执行成功。相关参数说明请参考[媒资图像标签API](#)。

```
class RunImageMediaTaggingResponse {
    result: class ImageMediaTaggingResponseResult {
        tags: [class ImageMediaTaggingItemBody {
            confidence: 83.15
            type: 社会
            tag: 火灾
            i18nTag: class ImageMediaTaggingItemBodyI18nTag {
                zh: 火灾
                en: Fire
            }
            i18nType: class ImageMediaTaggingItemBodyI18nType {
                zh: 社会
                en: Society
            }
            instances: []
        }, class ImageMediaTaggingItemBody {
            confidence: 70.21
            type: 其他
            tag: 火
            i18nTag: class ImageMediaTaggingItemBodyI18nTag {
                zh: 火
                en: Fire
            }
            i18nType: class ImageMediaTaggingItemBodyI18nType {
                zh: 其他
                en: Other
            }
        }
    }
}
```

```

    }
    instances: []
  }, class ImageMediaTaggingItemBody {
    confidence: 53.63
    type: 学科科学
    tag: 火球
    i18nTag: class ImageMediaTaggingItemBodyI18nTag {
      zh: 火球
      en: Fireball
    }
    i18nType: class ImageMediaTaggingItemBodyI18nType {
      zh: 学科科学
      en: Subject science
    }
    instances: []
  }, class ImageMediaTaggingItemBody {
    confidence: 53.45
    type: 其他
    tag: 营火
    i18nTag: class ImageMediaTaggingItemBodyI18nTag {
      zh: 营火
      en: Campfire
    }
    i18nType: class ImageMediaTaggingItemBodyI18nType {
      zh: 其他
      en: Other
    }
    instances: []
  }, class ImageMediaTaggingItemBody {
    confidence: 50.73
    type: 人工场景
    tag: 林业局
    i18nTag: class ImageMediaTaggingItemBodyI18nTag {
      zh: 林业局
      en: Forestry bureau
    }
    i18nType: class ImageMediaTaggingItemBodyI18nType {
      zh: 人工场景
      en: Manual scenario
    }
    instances: []
  }, class ImageMediaTaggingItemBody {
    confidence: 50.34
    type: 其他
    tag: 烧毁的
    i18nTag: class ImageMediaTaggingItemBodyI18nTag {
      zh: 烧毁的
      en: Burned
    }
    i18nType: class ImageMediaTaggingItemBodyI18nType {
      zh: 其他
      en: Other
    }
    instances: []
  }, class ImageMediaTaggingItemBody {
    confidence: 46.63
    type: 场景
    tag: 火焰
    i18nTag: class ImageMediaTaggingItemBodyI18nTag {
      zh: 火焰
      en: Flame
    }
    i18nType: class ImageMediaTaggingItemBodyI18nType {
      zh: 场景
      en: Scene
    }
    instances: []
  }, class ImageMediaTaggingItemBody {
    confidence: 45.27

```



```
public class RunCelebrityRecognitionSolution {

    public static void main(String[] args) {
        // 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量
        // 中密文存放，使用时解密，确保安全
        // 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境
        // 变量HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK
        String ak = System.getenv("HUAWEICLOUD_SDK_AK");
        String sk = System.getenv("HUAWEICLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        ImageClient client = ImageClient.newBuilder()
            .withCredential(auth)
            .withRegion(ImageRegion.valueOf("cn-north-1")) //此处替换为您开通服务的区域，详情请查
            .build();
        RunCelebrityRecognitionRequest request = new RunCelebrityRecognitionRequest();
        CelebrityRecognitionReq body = new CelebrityRecognitionReq();
        body.withThreshold(0f);
        body.withUrl("https://XXX.jpg"); //此处替换为公网可以访问的图片地址
        request.withBody(body);
        try {
            RunCelebrityRecognitionResponse response = client.runCelebrityRecognition(request);
            System.out.println(response.getHttpStatusCode());
            System.out.println(JSON.toJSONString(response));
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

2. 执行示例代码文件，控制台输出名人识别结果即表示程序执行成功。相关参数说明请参考[名人识别API](#)。

```
200
{"httpStatusCode":200,"result":[{"confidence":0.9990679,"faceDetail":
{"w":357,"h":510,"x":260,"y":201,"label":"迈克尔·杰克逊"}]}
Process finished with exit code 0
```

## 6.4 主体识别示例

本章节对主体识别AK/SK方式使用SDK进行示例说明。

- 主体识别示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用ImageMainObjectDetectionReq中的withUrl或withImage方法配置图像信息（image和url参数二选一），示例中以url为例，配置完成后运行即可。
- endpoint是华为云各服务应用区域和各服务的终端节点，详情请查看[地区和终端节点](#)。

1. 主体识别调用示例代码如下：

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
```



```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.image.v2.region.ImageRegion;
import com.huaweicloud.sdk.image.v2.*;
import com.huaweicloud.sdk.image.v2.model.*;

public class RunImageMainObjectDetectionSolution {

    public static void main(String[] args) {
        // 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量
        // 中密文存放，使用时解密，确保安全
        // 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境
        // 变量HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK
        String ak = System.getenv("HUAWEICLOUD_SDK_AK");
        String sk = System.getenv("HUAWEICLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        ImageClient client = ImageClient.newBuilder()
            .withCredential(auth)
            .withRegion(ImageRegion.valueOf("cn-north-4")) //此处替换为您开通服务的区域，详情请查
            // 看地区和终端节点
            .build();
        RunImageMainObjectDetectionRequest request = new RunImageMainObjectDetectionRequest();
        ImageMainObjectDetectionReq body = new ImageMainObjectDetectionReq();
        body.withThreshold(30f);
        body.withUrl("https://XXX.jpg"); //替换为公网可以访问的图片地址
        request.withBody(body);
        try {
            RunImageMainObjectDetectionResponse response =
            client.runImageMainObjectDetection(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

2. 执行示例代码文件，控制台输出主体识别结果即表示程序执行成功。相关参数说明请参考[主体识别API](#)。

```
RunImageMainObjectDetectionResponse {
  result: [class ImageMainObjectDetectionInstance {
    label: bounding_box
    location: {height=327.77, top_left_x=3.7, top_left_y=1.72, width=463.03}
    confidence: 95.73
  }, class ImageMainObjectDetectionInstance {
    label: main_object_box
    location: {height=328.35, top_left_x=3.22, top_left_y=1.5, width=456.63}
    confidence: 88.35
  }]
}
```

## 6.5 翻拍识别示例

本章节对翻拍识别AK/SK方式使用SDK进行示例说明。

- 翻拍识别示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用 RunRecaptureDetectReq中的withUrl或withImage方法配置图像信息（image和url参数二选一），示例中以url为例，配置完成后运行即可。
- endpoint是华为云各服务应用区域和各服务的终端节点，详情请查看[地区和终端节点](#)。

1. 翻拍识别调用示例代码如下：

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.image.v2.region.ImageRegion;
import com.huaweicloud.sdk.image.v2.*;
import com.huaweicloud.sdk.image.v2.model.*;

public class RunRecaptureDetectSolution {

    public static void main(String[] args) {
        // 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全
        // 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK
        String ak = System.getenv("HUAWEICLOUD_SDK_AK");
        String sk = System.getenv("HUAWEICLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        ImageClient client = ImageClient.newBuilder()
            .withCredential(auth)
            .withRegion(ImageRegion.valueOf("cn-north-4")) //此处替换为您开通服务的区域，详情请查看地区和终端节点.
            .build();
        RunRecaptureDetectRequest request = new RunRecaptureDetectRequest();
        RecaptureDetectReq body = new RecaptureDetectReq();
        body.withThreshold(0f);
        body.withUrl("https://bucketname.obs.cn-north-4.myhuaweicloud.com/image/test1.jpg"); //替换为公网可以访问的图片地址
        request.withBody(body);
        try {
            RunRecaptureDetectResponse response = client.runRecaptureDetect(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

2. 执行示例代码文件，控制台输出翻拍识别结果即表示程序执行成功。

```
class RunRecaptureDetectResponse {
    result: class RecaptureDetectResponseResult {
        suggestion: false
        category: recapture
        score: 0.9998
        detail: [class RecaptureDetectResponseResultDetail {
            label: recapture
```

```

        confidence: 0.9998
    }
}
}

```

## 6.6 图像标签示例

本章节对图像标签AK/SK方式使用SDK进行示例说明。

- 图像标签示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用 ImageTaggingReq 中的 withUrl 或 withImage 方法配置图像信息（image 和 url 参数二选一），示例中以 url 为例配置完成后运行即可。
- endpoint 是华为云各服务应用区域和各服务的终端节点，详情请查看 [地区和终端节点](#)。

1. 图像标签调用示例代码如下：

```

package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.image.v2.region.ImageRegion;
import com.huaweicloud.sdk.image.v2.*;
import com.huaweicloud.sdk.image.v2.model.*;

public class RunImageTaggingSolution {

    public static void main(String[] args) {
        // 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量
        // 中密文存放，使用时解密，确保安全
        // 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境
        // 变量HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK
        String ak = System.getenv("HUAWEICLOUD_SDK_AK");
        String sk = System.getenv("HUAWEICLOUD_SDK_SK");
        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);
        ImageClient client = ImageClient.newBuilder()
            .withCredential(auth)
            .withRegion(ImageRegion.valueOf("XXX")) //此处替换为您开通服务的区域，详情请查看地
            // 区和终端节点
            .build();
        RunImageTaggingRequest request = new RunImageTaggingRequest();
        ImageTaggingReq body = new ImageTaggingReq();
        body.withLimit(50);
        body.withThreshold(95f);
        body.withLanguage("zh");
        body.withUrl("https://XXX.jpg"); //此处替换为公网可以访问的图片地址
        request.withBody(body);
        try {
            RunImageTaggingResponse response = client.runImageTagging(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}

```

2. 执行示例代码，控制台返回200即表示程序执行成功。相关参数说明请参考[图像标签API](#)。

```
class RunImageTaggingResponse {
  result: class ImageTaggingResponseResult {
    tags: [class ImageTaggingItemBody {
      confidence: 98.00
      type: 树
      tag: 树
      i18nTag: class ImageTaggingItemBodyI18nTag {
        zh: 树
        en: Tree
      }
      i18nType: class ImageTaggingItemBodyI18nType {
        zh: 树
        en: Tree
      }
      instances: []
    }]
  }
}
Process finished with exit code 0
```

# 7 使用 SDK ( Python )

---

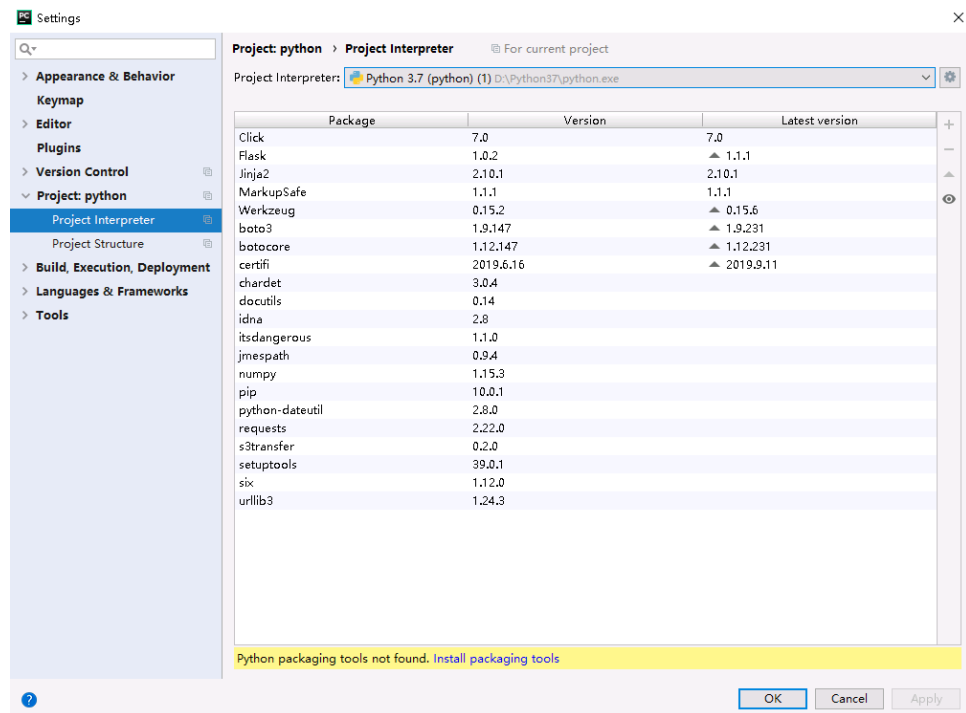
## 7.1 Python 开发环境配置

### 环境配置

使用图像识别Python版本SDK包，需要您配置Python开发环境。

1. 从[Python官网](#)下载并安装合适的Python版本。请使用Python3.3以上版本，如下以Python3.7 版本为例进行说明。
2. 从[PyCharm官网](#)下载并安装最新版本。
3. 在PyCharm开发工具中配置Python环境，在菜单依次选择“File > Settings > Project Interpreter”。
4. 在页面上方选择您的Python安装路径，如图 [PyCharm配置python环境所示](#)。选择好目标Python之后单击页面下方“Apply”完成配置。

图 7-1 PyCharm 配置 python 环境



## SDK 获取和安装

python语言SDK的获取和安装推荐使用pip命令进行安装或使用pycharm进行安装，需要安装huaweicloudsdkcore包以及huaweicloudsdkimage，参考方法如下：

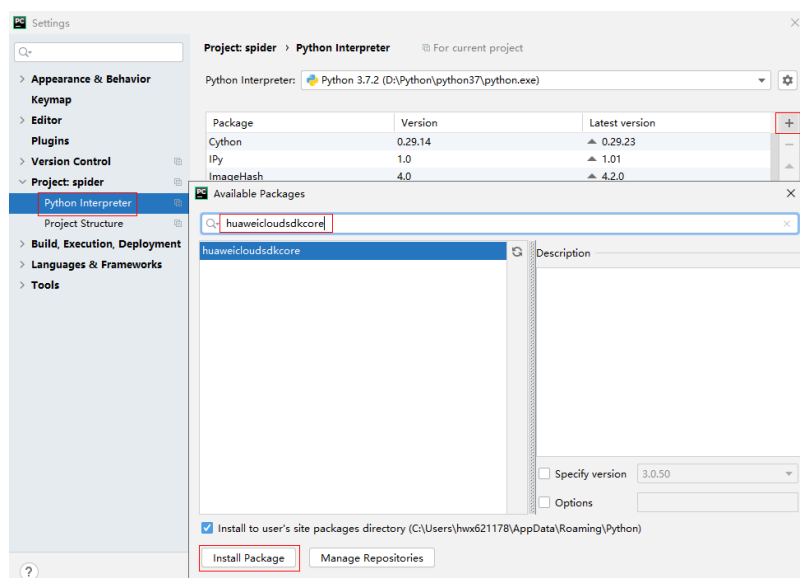
```
pip 安装：
# 安装核心库
pip install huaweicloudsdkcore
```

```
# 安装image服务库
pip install huaweicloudsdkimage
```

使用pycharm安装，步骤如下：

1. 打开pycharm,单击File -> Settings...
2. 单击Python Interpreter -> 单击+
3. 单击+, 分别搜索huaweicloudsdkcore及huaweicloudsdkimage，搜索到包内容单击左下角Install Package完成安装

图 7-2 pycharm 安装图像识别 python 版本 sdk 包



## 7.2 媒资图像标签示例

本章节对媒资图像标签AK/SK方式使用SDK进行示例说明。

- 媒资图像标签示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用 ImageMediaTaggingReq 中的 url 或 image 方法配置图像信息（image 和 url 参数二选一），示例中以 url 为例，配置完成后运行即可。
- endpoint 是华为云各服务应用区域和各服务的终端节点，详情请查看 [地区和终端节点](#)。

1. 媒资图像标签调用示例代码如下：

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkimage.v2.region.image_region import ImageRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkimage.v2 import *

if __name__ == "__main__":
    # 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中
    # 密文存放，使用时解密，确保安全；
    # 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
    # HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK。
    ak = os.getenv("HUAWEICLOUD_SDK_AK")
    sk = os.getenv("HUAWEICLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = ImageClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ImageRegion.value_of("cn-north-4")) \ //此处替换为您开通服务的区域
        .build()

    try:
        request = RunImageMediaTaggingRequest()
        request.body = ImageMediaTaggingReq(
            threshold=10,
            language="zh",
            url="https://XXX.jpg" //此处替换为公网可以访问的图片地址
        )
```

```
response = client.run_image_media_tagging(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

2. 执行示例代码文件，控制台输出媒资图像标签结果即表示程序执行成功。相关参数说明请参考[媒资图像标签API](#)。

```
{
  "result": {
    "tags": [
      {
        "confidence": "83.15",
        "type": "社会",
        "tag": "火灾",
        "i18n_tag": {
          "zh": "火灾",
          "en": "Fire"
        },
        "i18n_type": {
          "zh": "社会",
          "en": "Society"
        },
        "instances": []
      },
      {
        "confidence": "70.21",
        "type": "其他",
        "tag": "火",
        "i18n_tag": {
          "zh": "火",
          "en": "Fire"
        },
        "i18n_type": {
          "zh": "其他",
          "en": "Other"
        },
        "instances": []
      },
      {
        "confidence": "53.63",
        "type": "学科科学",
        "tag": "火球",
        "i18n_tag": {
          "zh": "火球",
          "en": "Fireball"
        },
        "i18n_type": {
          "zh": "学科科学",
          "en": "Subject science"
        },
        "instances": []
      },
      {
        "confidence": "53.45",
        "type": "其他",
        "tag": "营火",
        "i18n_tag": {
          "zh": "营火",
          "en": "Campfire"
        },
        "i18n_type": {
          "zh": "其他",
          "en": "Other"
        },
        "instances": []
      },
      {
        "confidence": "50.73",
        "type": "人工场景",
        "tag": "林业局",
        "i18n_tag": {
          "zh": "林业局",
          "en": "Forestry bureau"
        },
        "i18n_type": {
          "zh": "人工场景",
          "en": "Manual scenario"
        },
        "instances": []
      },
      {
        "confidence": "50.34",
        "type": "其他",
        "tag": "烧毁的",
        "i18n_tag": {
          "zh": "烧毁的",
          "en": "Burned"
        },
        "i18n_type": {
          "zh": "其他",
          "en": "Other"
        },
        "instances": []
      },
      {
        "confidence": "46.63",
        "type": "场景",
        "tag": "火焰",
        "i18n_tag": {
          "zh": "火焰",
          "en": "Flame"
        },
        "i18n_type": {
          "zh": "场景",
          "en": "Scene"
        },
        "instances": []
      },
      {
        "confidence": "45.27",
        "type": "其他",
        "tag": "燃烧",
        "i18n_tag": {
          "zh": "燃烧",
          "en": "Burning"
        },
        "i18n_type": {
          "zh": "其他",
          "en": "Other"
        },
        "instances": []
      },
      {
        "confidence": "41.32",
        "type": "职业/角色",
        "tag": "消防员",
        "i18n_tag": {
          "zh": "消防员",
          "en": "Fireman"
        },
        "i18n_type": {
          "zh": "职业/角色",
          "en": "Career/Role"
        },
        "instances": []
      },
      {
        "confidence": "39.97",
        "type": "动作",
        "tag": "烧香",
        "i18n_tag": {
          "zh": "烧香",
          "en": "Incense burning"
        },
        "i18n_type": {
          "zh": "动作",
          "en": "Action"
        },
        "instances": []
      }
    ]
  }
}
```

## 7.3 名人识别示例

本章节对名人识别AK/SK方式使用SDK进行示例说明。

- 名人识别示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用初始化CelebrityRecognitionReq中的url或image配置图像信息（image和url参数二选一），示例中以url为例，配置完成后运行即可。
- endpoint是华为云各服务应用区域和各服务的终端节点，详情请查看[地区和终端节点](#)。

1. 名人识别调用示例代码如下：

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcore.http.http_config import HttpConfig
from huaweicloudsdkimage.v2 import ImageClient, RunCelebrityRecognitionRequest, CelebrityRecognitionReq
from huaweicloudsdkimage.v2.region.image_region import ImageRegion

if __name__ == "__main__":
    # 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；
    # 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK。
    ak = os.getenv("HUAWEICLOUD_SDK_AK")
    sk = os.getenv("HUAWEICLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk)

    client = ImageClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ImageRegion.value_of("cn-north-1")) \
        //此处替换为您开通服务的区域，详情请查看地区和终端节点
        .build()

    try:
        request = RunCelebrityRecognitionRequest()
```



```

request.body = CelebrityRecognitionReq(
    threshold=0,
    url="https://XXX.jpg" //此处替换为公网可以访问的图片地址
)
response = client.run_celebrity_recognition(request)
print(response.status_code)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

2. 执行示例代码文件，控制台输出名人识别结果即表示程序执行成功。相关参数说明请参考[名人识别API](#)。

```

200
{'result': [{'confidence': 0.9990678852332598,
             'face_detail': {'h': 510, 'w': 357, 'x': 260, 'y': 201},
             'label': '迈克尔·杰克逊'}]}

```

## 7.4 主体识别示例

本章节对主体识别AK/SK方式使用SDK进行示例说明。

- 主体识别示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用初始化ImageMainObjectDetectionReq中的url或image配置图像信息（image和url参数二选一），示例中以url为例，配置完成后运行即可。
- endpoint是华为云各服务应用区域和各服务的终端节点，详情请查看[地区和终端节点](#)。

1. 主体识别调用示例代码如下：

```

# coding: utf-8
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkimage.v2.region.image_region import ImageRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkimage.v2 import *

if __name__ == "__main__":
    # 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中
    # 密文存放，使用时解密，确保安全；
    # 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
    # HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK。
    ak = os.getenv("HUAWEICLOUD_SDK_AK")
    sk = os.getenv("HUAWEICLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = ImageClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ImageRegion.value_of("cn-north-4")) \ //此处替换为您开通服务的区域，详情请查看
地区和终端节点
        .build()

    try:
        request = RunImageMainObjectDetectionRequest()
        request.body = ImageMainObjectDetectionReq(
            threshold=30,
            url="https://XXX.jpg" //替换为公网可以访问的图片地址
        )
        response = client.run_image_main_object_detection(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)

```

- ```
print(e.error_code)
print(e.error_msg)
```
2. 执行示例代码文件，控制台输出主体识别结果即表示程序执行成功。相关参数说明请参考[主体识别API](#)。
 

```
{
  "result": [{
    "label": "bounding_box",
    "location": {"height": 327.77, "top_left_x": 3.7, "top_left_y": 1.72, "width": 463.03},
    "confidence": "95.73"}, {"label": "main_object_box",
    "location": {"height": 328.35, "top_left_x": 3.22, "top_left_y": 1.5, "width": 456.63},
    "confidence": "88.35"}]}
}
```

## 7.5 翻拍识别示例

本章节对翻拍识别AK/SK方式使用SDK进行示例说明。

- 翻拍识别示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用初始化RunRecaptureDetectReq中的url或image配置图像信息（image和url参数二选一），示例中以url为例，配置完成后运行即可。
- endpoint是华为云各服务应用区域和各服务的终端节点，详情请查看[地区和终端节点](#)。

1. 翻拍识别调用示例代码如下：

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkimage.v2.region.image_region import ImageRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkimage.v2 import *

if __name__ == "__main__":
    # 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中
    # 密文存放，使用时解密，确保安全；
    # 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
    # HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK。
    ak = os.getenv("HUAWEICLOUD_SDK_AK")
    sk = os.getenv("HUAWEICLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = ImageClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ImageRegion.value_of("cn-north-4")) \ //此处替换为您开通服务的区域，详情请查看
地区和终端节点
        .build()

    try:
        request = RunRecaptureDetectRequest()
        request.body = RecaptureDetectReq(
            threshold=0,
            url="https://bucketname.obs.cn-north-4.myhuaweicloud.com/image/test1.jpg"//替换为公网可
            以访问的图片地址
        )
        response = client.run_recapture_detect(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

2. 执行示例代码文件，控制台输出翻拍识别结果即表示程序执行成功。

```
{"result":{"suggestion":"false","category":"recapture","score":"0.9998","detail":[{"label":"recapture","confidence":"0.9998"}]}
```

## 7.6 图像标签示例

本章节对图像标签AK/SK方式使用SDK进行示例说明。

- 图像标签示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用初始化 ImageTaggingReq 中的 url 或 image 配置图像信息（image 和 url 参数二选一），示例中以 url 为例，配置完成后运行即可。
- endpoint 是华为云各服务应用区域和各服务的终端节点，详情请查看 [地区和终端节点](#)。

1. 图像标签调用示例代码如下：

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcore.http.http_config import HttpConfig
from huaweicloudsdkimage.v2 import ImageClient, RunImageTaggingRequest, ImageTaggingReq
from huaweicloudsdkimage.v2.region.image_region import ImageRegion

if __name__ == "__main__":
    # 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中
    # 密文存放，使用时解密，确保安全；
    # 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
    # HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK。
    ak = os.getenv("HUAWEICLOUD_SDK_AK")
    sk = os.getenv("HUAWEICLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk)

    client = ImageClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ImageRegion.value_of("cn-north-1")) \
        //此处替换为您开通服务的区域，详情请查看
        // 地区和终端节点
        .build()

    try:
        request = RunImageTaggingRequest()
        request.body = ImageTaggingReq(
            limit=50,
            threshold=95,
            language="zh",
            url="https://XXX.jpg" //此处替换为公网可以访问的图片地址
        )
        response = client.run_image_tagging(request)
        print(response.status_code)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

2. 执行示例代码，控制台返回200即表示程序执行成功。相关参数说明请参考 [图像标签API](#)。

```
200
{"result": {"tags": [{"confidence": "98.00", "type": "树", "tag": "树", "i18n_tag": {"zh": "树", "en": "Tree"}}, {"i18n_type": {"zh": "树", "en": "Tree"}, "instances": []]}}
```

Process finished with exit code 0

# 8 使用 SDK (.NET)

## 8.1 .NET 开发环境配置

### 环境配置

使用图像识别.NET版本SDK包，需要先配置.Net开发环境。

1. 从[Visual Studio官网](#)，下载最新版Visual Studio。
2. 下载完毕后，单击exe文件，开始安装。
3. 安装过程中选择工作负荷ASP.NET和Web开发，安装完成后启动即可。

图 8-1 选择安装.NET 环境依赖



### SDK 获取和安装

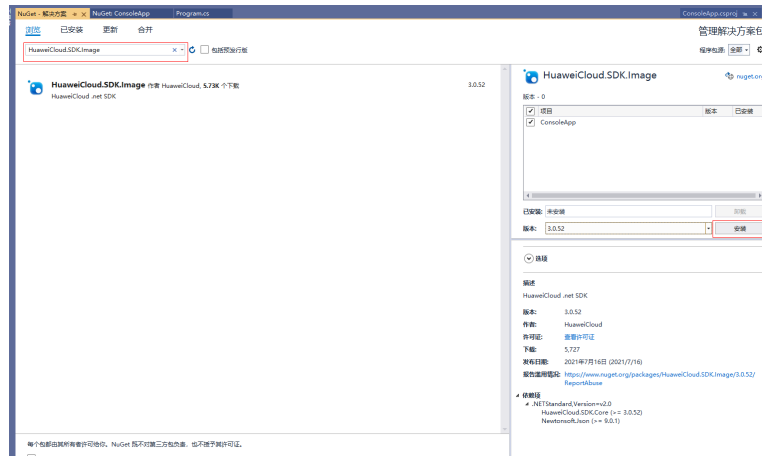
.NET语言SDK的获取和安装可以使用可以通过命令方式安装或使用Visual Studio开发工具安装第三方引用，参考方法如下：

```
使用 .NET CLI 工具
dotnet add package HuaweiCloud.SDK.Core
dotnet add package HuaweiCloud.SDK.Image
```

```
使用 Package Manager
Install-Package HuaweiCloud.SDK.Core
Install-Package HuaweiCloud.SDK.Image
```

使用Visual Studio开发工具安装第三方引用：单击工具->选择NuGet包管理器->单击管理解决方案的NuGet程序包，选择浏览中搜索并安装HuaweiCloud.SDK.Core及HuaweiCloud.SDK.Image,如下图所示：

图 8-2 安装图像识别.NET 版本 sdk 包



## 8.2 媒资图像标签示例

本章节对媒资图像标签AK/SK方式使用SDK进行示例说明。

- 媒资图像标签示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用 ImageMediaTaggingReq中的url或image方法配置图像信息（image和url参数二选一），示例中以url为例，配置完成后运行即可。
- endpoint是华为云各服务应用区域和各服务的终端节点，详情请查看 [地区和终端节点](#)。

1. 媒资图像标签调用示例代码如下：

```
using System;
using System.Collections.Generic;
using HuaweiCloud.SDK.Core;
using HuaweiCloud.SDK.Core.Auth;
using HuaweiCloud.SDK.Image;
using HuaweiCloud.SDK.Image.V2;
using HuaweiCloud.SDK.Image.V2.Model;

namespace RunImageMediaTaggingSolution
{
    class Program
    {
        static void Main(string[] args)
        {
            //认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量
            //中密文存放，使用时解密，确保安全
            //本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
            //HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK
            string accessKey= Environment.GetEnvironmentVariable("HUAWEICLOUD_SDK_AK",
            EnvironmentVariableTarget.Machine);
            string secretKey= Environment.GetEnvironmentVariable("HUAWEICLOUD_SDK_SK",
            EnvironmentVariableTarget.Machine);

            var config = HttpConfig.GetDefaultConfig();
            config.IgnoreSslVerification = true;
            var auth = new BasicCredentials(ak, sk);

            var client = ImageClient.NewBuilder()
```

```

        .WithCredential(auth)
        .WithRegion(ImageRegion.ValueOf("cn-north-4")) //此处替换为您开通服务的区域，详情
        请查看地区和终端节点。
        .WithHttpConfig(config)
        .Build();

        var req = new RunImageMediaTaggingRequest
        {
        };
        req.Body = new ImageMediaTaggingReq()
        {
            Threshold = 10,
            Language = "zh",
            Url = "https://XXX.jpg" //此处替换为公网可以访问的图片地址
        };

        try
        {
            var resp = client.RunImageMediaTagging(req);
            var respStatusCode = resp.HttpStatusCode;
            Console.WriteLine(respStatusCode);
            Console.WriteLine(JsonConvert.DeserializeObject(resp.HttpBody));
        }
        catch (RequestTimeoutException requestTimeoutException)
        {
            Console.WriteLine(requestTimeoutException.ErrorMessage);
        }
        catch (ServiceResponseException clientRequestException)
        {
            Console.WriteLine(clientRequestException.HttpStatusCode);
            Console.WriteLine(clientRequestException.ErrorCode);
            Console.WriteLine(clientRequestException.ErrorMessage);
        }
        catch (ConnectionException connectionException)
        {
            Console.WriteLine(connectionException.ErrorMessage);
        }
    }
}

```

2. 执行示例代码文件，控制台输出媒资图像标签结果即表示程序执行成功。相关参数说明请参考[媒资图像标签API](#)。

```

200
{"result": {"tags": [{"confidence": "83.15", "type": "社会", "tag": "火灾", "i18n_tag": {"zh": "火灾", "en": "Fire"}, "i18n_type": {"zh": "社会", "en": "Society"}, "instances": []}, {"confidence": "70.21", "type": "其他", "tag": "火", "i18n_tag": {"zh": "火", "en": "Fire"}, "i18n_type": {"zh": "其他", "en": "Other"}, "instances": []}, {"confidence": "53.63", "type": "学科科学", "tag": "火球", "i18n_tag": {"zh": "火球", "en": "Fireball"}, "i18n_type": {"zh": "学科科学", "en": "Subject science"}, "instances": []}, {"confidence": "53.45", "type": "其他", "tag": "营火", "i18n_tag": {"zh": "营火", "en": "Campfire"}, "i18n_type": {"zh": "其他", "en": "Other"}, "instances": []}, {"confidence": "50.73", "type": "人工场景", "tag": "林业局", "i18n_tag": {"zh": "林业局", "en": "Forestry bureau"}, "i18n_type": {"zh": "人工场景", "en": "Manual scenario"}, "instances": []}, {"confidence": "50.34", "type": "其他", "tag": "烧毁的", "i18n_tag": {"zh": "烧毁的", "en": "Burned"}, "i18n_type": {"zh": "其他", "en": "Other"}, "instances": []}, {"confidence": "46.63", "type": "场景", "tag": "火焰", "i18n_tag": {"zh": "火焰", "en": "Flame"}, "i18n_type": {"zh": "场景", "en": "Scene"}, "instances": []}, {"confidence": "45.27", "type": "其他", "tag": "燃烧", "i18n_tag": {"zh": "燃烧", "en": "Burning"}, "i18n_type": {"zh": "其他", "en": "Other"}, "instances": []}, {"confidence": "41.32", "type": "职业/角色", "tag": "消防员", "i18n_tag": {"zh": "消防员", "en": "Fireman"}, "i18n_type": {"zh": "职业/角色", "en": "Career/Role"}, "instances": []}, {"confidence": "39.97", "type": "动作", "tag": "烧香", "i18n_tag": {"zh": "烧香", "en": "Incense burning"}, "i18n_type": {"zh": "动作", "en": "Action"}, "instances": []}]}

```

## 8.3 名人识别示例

本章节对名人识别AK/SK方式使用SDK进行示例说明。

- 名人识别示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用 CelebrityRecognitionReq中的url或image方法配置图像信息（image和url参数二选一），示例中以url为例，配置完成后运行即可。
- endpoint是华为云各服务应用区域和各服务的终端节点，详情请查看 [地区和终端节点](#)。

1. 名人识别调用示例代码如下：

```
using System;
using System.Collections.Generic;
using HuaweiCloud.SDK.Core;
using HuaweiCloud.SDK.Core.Auth;
using HuaweiCloud.SDK.Image;
using HuaweiCloud.SDK.Image.V2;
using HuaweiCloud.SDK.Image.V2.Model;
using Newtonsoft.Json;

namespace RunCelebrityRecognitionSolution
{
    class Program
    {
        static void Main(string[] args){
            //认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全
            //本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK
            string accessKey= Environment.GetEnvironmentVariable("HUAWEICLOUD_SDK_AK", EnvironmentVariableTarget.Machine);
            string secretKey= Environment.GetEnvironmentVariable("HUAWEICLOUD_SDK_SK", EnvironmentVariableTarget.Machine);

            var config = HttpConfig.GetDefaultConfig();
            config.IgnoreSslVerification = true;
            var auth = new BasicCredentials(ak, sk);

            var client = ImageClient.NewBuilder()
                .WithCredential(auth)
                .WithRegion(ImageRegion.ValueOf("cn-north-1")) //此处替换为您开通服务的区域，详情请查看地区和终端节点
                .WithHttpConfig(config)
                .Build();

            var req = new RunCelebrityRecognitionRequest
            {
            };
            req.Body = new CelebrityRecognitionReq()
            {
                Threshold = 0.48,
                Url = "https://XXX.jpg" //此处替换为公网可以访问的图片地址
            };

            try
            {
                var resp = client.RunCelebrityRecognition(req);
                var respStatusCode = resp.HttpStatusCode;
                Console.WriteLine(respStatusCode);
                Console.WriteLine(JsonConvert.DeserializeObject(resp.HttpBody));
            }
            catch (RequestTimeoutException requestTimeoutException)
            {
                Console.WriteLine(requestTimeoutException.ErrorMessage);
            }
            catch (ServiceResponseException clientRequestException)
            {
                Console.WriteLine(clientRequestException.HttpStatusCode);
                Console.WriteLine(clientRequestException.ErrorCode);
                Console.WriteLine(clientRequestException.ErrorMsg);
            }
            catch (ConnectionException connectionException)
            {
            }
        }
    }
}
```

```

        {
            Console.WriteLine(connectionException.ErrorMessage);
        }
    }
}

```

2. 执行示例代码文件，控制台输出名人识别结果即表示程序执行成功。相关参数说明请参考[名人识别API](#)。

```

200
{
  'result': [{
    'confidence': 0.9990678852332598,
    'face_detail': {
      'h': 510,
      'w': 357,
      'x': 260,
      'y': 201
    },
    'label': '迈克尔·杰克逊'
  }]
}

```

## 8.4 主体识别示例

本章节对主体识别AK/SK方式使用SDK进行示例说明。

- 主体识别示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用 ImageMainObjectDetectionReq 中的 url 或 image 方法配置图像信息（image 和 url 参数二选一），示例中以 url 为例，配置完成后运行即可。
- endpoint 是华为云各服务应用区域和各服务的终端节点，详情请查看 [地区和终端节点](#)。

1. 主体识别调用示例代码如下：

```

using System;
using System.Collections.Generic;
using HuaweiCloud.SDK.Core;
using HuaweiCloud.SDK.Core.Auth;
using HuaweiCloud.SDK.Image;
using HuaweiCloud.SDK.Image.V2;
using HuaweiCloud.SDK.Image.V2.Model;

namespace RunImageMainObjectDetectionSolution
{
    class Program
    {
        static void Main(string[] args) {
            //认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量
            //中密文存放，使用时解密，确保安全
            //本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
            //HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK
            string accessKey= Environment.GetEnvironmentVariable("HUAWEICLOUD_SDK_AK",
            EnvironmentVariableTarget.Machine);
            string secretKey= Environment.GetEnvironmentVariable("HUAWEICLOUD_SDK_SK",
            EnvironmentVariableTarget.Machine);

            var config = HttpConfig.GetDefaultConfig();
            config.IgnoreSslVerification = true;
            var auth = new BasicCredentials(ak, sk);

            var client = ImageClient.NewBuilder()
                .WithCredential(auth)
                .WithRegion(ImageRegion.ValueOf("cn-north-4")) //此处替换为您开通服务的区域，详情
                //请查看地区和终端节点
                .WithHttpConfig(config)
                .Build();

```





1. 翻拍识别调用示例代码如下：

```
using System;
using System.Collections.Generic;
using HuaweiCloud.SDK.Core;
using HuaweiCloud.SDK.Core.Auth;
using HuaweiCloud.SDK.Image;
using HuaweiCloud.SDK.Image.V2;
using HuaweiCloud.SDK.Image.V2.Model;

namespace RunRecaptureDetectSolution
{
    class Program
    {
        static void Main(string[] args)
        {
            //认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量
            //中密文存放，使用时解密，确保安全
            //本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
            //HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK
            string accessKey= Environment.GetEnvironmentVariable("HUAWEICLOUD_SDK_AK",
            EnvironmentVariableTarget.Machine);
            string secretKey= Environment.GetEnvironmentVariable("HUAWEICLOUD_SDK_SK",
            EnvironmentVariableTarget.Machine);

            var config = HttpConfig.GetDefaultConfig();
            config.IgnoreSslVerification = true;
            var auth = new BasicCredentials(ak, sk);

            var client = ImageClient.NewBuilder()
                .WithCredential(auth)
                .WithRegion(ImageRegion.ValueOf("cn-north-4")) //此处替换为您开通服务的区域，详情
                请查看地区和终端节点
                .WithHttpConfig(config)
                .Build();

            var req = new RunRecaptureDetectRequest
            {
            };
            req.Body = new RecaptureDetectReq()
            {
                Threshold = 0,
                Url = "https://bucketname.obs.cn-north-4.myhuaweicloud.com/image/test1.jpg"//替换为公
                网可以访问的图片地址
            };

            try
            {
                var resp = client.RunRecaptureDetect(req);
                var respStatusCode = resp.HttpStatusCode;
                Console.WriteLine(respStatusCode);
                Console.WriteLine(resp.HttpBody);
            }
            catch (RequestTimeoutException requestTimeoutException)
            {
                Console.WriteLine(requestTimeoutException.ErrorMessage);
            }
            catch (ServiceResponseException clientRequestException)
            {
                Console.WriteLine(clientRequestException.HttpStatusCode);
                Console.WriteLine(clientRequestException.RequestId);
                Console.WriteLine(clientRequestException.ErrorCode);
                Console.WriteLine(clientRequestException.ErrorMessage);
            }
            catch (ConnectionException connectionException)
            {
                Console.WriteLine(connectionException.ErrorMessage);
            }
        }
    }
}
```

```
}
}
```

2. 执行示例代码文件，控制台输出翻拍识别结果即表示程序执行成功。

```
200
{"result":{"suggestion":"false","category":"recapture","score":"0.9998","detail":
[{"label":"recapture","confidence":"0.9998"}]}}
```

## 8.6 图像标签示例

本章节对图像标签AK/SK方式使用SDK进行示例说明。

- 图像标签示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用初始化 ImageTaggingReq 中的 url 或 image 配置图像信息（image 和 url 参数二选一），示例中以 url 为例，配置完成后运行即可。
- endpoint 是华为云各服务应用区域和各服务的终端节点，详情请查看 [地区和终端节点](#)。

1. 图像标签调用示例代码如下：

```
using System;
using System.Collections.Generic;
using HuaweiCloud.SDK.Core;
using HuaweiCloud.SDK.Core.Auth;
using HuaweiCloud.SDK.Image;
using HuaweiCloud.SDK.Image.V2;
using HuaweiCloud.SDK.Image.V2.Model;
using Newtonsoft.Json;

namespace RunImageTaggingSolution
{
    class Program
    {
        static void Main(string[] args) {
            //认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量
            //中密文存放，使用时解密，确保安全
            //本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
            //HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK
            string accessKey= Environment.GetEnvironmentVariable("HUAWEICLOUD_SDK_AK",
            EnvironmentVariableTarget.Machine);
            string secretKey= Environment.GetEnvironmentVariable("HUAWEICLOUD_SDK_SK",
            EnvironmentVariableTarget.Machine);

            var config = HttpConfig.GetDefaultConfig();
            config.IgnoreSslVerification = true;
            var auth = new BasicCredentials(ak, sk);

            var client = ImageClient.NewBuilder()
                .WithCredential(auth)
                .WithRegion(ImageRegion.ValueOf("cn-north-1")) //此处替换为您开通服务的区域，详情
                //请查看地区和终端节点
                .WithHttpConfig(config)
                .Build();

            var req = new RunImageTaggingRequest
            {
            };
            req.Body = new ImageTaggingReq()
            {
                Limit = 50,
                Threshold = 95,
                Language = "zh",
                Url = "https://XXX.jpg" //此处替换为公网可以访问的图片地址
            };

            try
            {
```

```

        var resp = client.RunImageTagging(req);
        var respStatusCode = resp.HttpStatusCode;
        Console.WriteLine(respStatusCode);
        Console.WriteLine(JsonConvert.DeserializeObject(resp.HttpBody));
    }
    catch (RequestTimeoutException requestTimeoutException)
    {
        Console.WriteLine(requestTimeoutException.ErrorMessage);
    }
    catch (ServiceResponseException clientRequestException)
    {
        Console.WriteLine(clientRequestException.HttpStatusCode);
        Console.WriteLine(clientRequestException.ErrorCode);
        Console.WriteLine(clientRequestException.ErrorMsg);
    }
    catch (ConnectionException connectionException)
    {
        Console.WriteLine(connectionException.ErrorMessage);
    }
}
}
}

```

2. 执行示例代码，控制台返回200即表示程序执行成功。相关参数说明请参考[图像标签API](#)。

```

200
{
  "result": {
    "tags": [
      {
        "confidence": "98.01",
        "type": "树",
        "tag": "树",
        "i18n_tag": {
          "zh": "树",
          "en": "Tree"
        },
        "i18n_type": {
          "zh": "树",
          "en": "Tree"
        },
        "instances": [ ]
      }
    ]
  }
}

```

# 9 使用 SDK (GO)

## 9.1 GO 开发环境配置

### 准备工作

使用图像识别GO版本SDK包，需要您配置Go语言开发环境。如已经安装配置go开发环境，可跳过该步骤。

1. 在[golang 官网](#)下载并安装对应的go版本，sdk支持 go 1.14 及以上版本。
2. 下载完成后按照提示安装，位置自选，比如安装到本地D:\develop\go。
3. 配置GO环境变量：右键“计算机>属性>高级系统设置>环境变量”，进行如下操作。
  - a. 新建系统变量GOROOT，变量值为实际go安装位置。新增环境变量GOPATH，变量为go语言工程所在目录。
  - b. 修改PATH变量，末尾增加GO的编译器路径“;%GOROOT%\bin”，中间分隔符是【;】。
  - c. 修改CLASSPATH变量，末尾增加GO的编译器路径“;%GOROOT%\lib”，中间分隔符是【;】。
4. 在[vs code官网](#)，下载安装操作系统的vs code。
5. 安装go语言的插件Go和 Go Critic。

### SDK 获取和安装

GO语言SDK的获取和安装使用命令进行安装，参考方法如下：

```
go get -u github.com/huaweicloud/huaweicloud-sdk-go-v3
```

## 9.2 媒资图像标签示例

本章节对媒资图像标签AK/SK方式使用SDK进行示例说明。

- 媒资图像标签示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用ImageMediaTaggingReq中的Url或Image方法配置图像信息（image和url参数二选一），示例中以url为例，配置完成后运行即可。

- endpoint是华为云各服务应用区域和各服务的终端节点，详情请查看[地区和终端节点](#)。

1. 媒资图像标签调用示例代码如下：

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    image "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/image/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/image/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/image/v2/region"
)

func main() {
    // 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中
    // 密文存放，使用时解密，确保安全；
    // 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
    // HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK。
    ak := os.Getenv("HUAWEICLOUD_SDK_AK")
    sk := os.Getenv("HUAWEICLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := image.NewImageClient(
        image.ImageClientBuilder().
        WithRegion(region.ValueOf("cn-north-4")). //此处替换为您开通服务的区域，详情请查看地区和终端节点
        WithCredential(auth).
        Build())

    request := &model.RunImageMediaTaggingRequest{
        thresholdImageMediaTaggingReq:= float32(10)
        languageImageMediaTaggingReq:= "zh"
        urlImageMediaTaggingReq:= "https://XXX.jpg" //此处替换为公网可以访问的图片地址
        request.Body = &model.ImageMediaTaggingReq{
            Threshold: &thresholdImageMediaTaggingReq,
            Language: &languageImageMediaTaggingReq,
            Url: &urlImageMediaTaggingReq,
        }
    }
    response, err := client.RunImageMediaTagging(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

2. 执行示例代码文件，控制台输出媒资图像标签结果即表示程序执行成功。相关参数说明请参考[媒资图像标签API](#)。

```
RunImageMediaTaggingResponse {"result": {"tags": [{"confidence": "83.15", "type": "社会", "tag": "火灾", "i18n_tag": {"zh": "火灾", "en": "Fire"}, "i18n_type": {"zh": "社会", "en": "Society"}, "instances": []}, {"confidence": "70.21", "type": "其他", "tag": "火", "i18n_tag": {"zh": "火", "en": "Fire"}, "i18n_type": {"zh": "其他", "en": "Other"}, "instances": []}, {"confidence": "53.63", "type": "学科科学", "tag": "火球", "i18n_tag": {"zh": "火球", "en": "Fireball"}, "i18n_type": {"zh": "学科科学", "en": "Subject science"}, "instances": []}, {"confidence": "53.45", "type": "其他", "tag": "营火", "i18n_tag": {"zh": "营火", "en": "Campfire"}, "i18n_type": {"zh": "其他", "en": "Other"}, "instances": []}, {"confidence": "50.73", "type": "人工场景", "tag": "林业局", "i18n_tag": {"zh": "林业局", "en": "Forestry bureau"}, "i18n_type": {"zh": "人工场景", "en": "Manual scenario"}, "instances": []}, {"confidence": "50.34", "type": "其他", "tag": "烧毁的", "i18n_tag": {"zh": "烧毁的", "en": "Burned"}, "i18n_type": {"zh": "其他", "en": "Other"}, "instances": []}, {"confidence": "46.63", "type": "场景", "tag": "火焰", "i18n_tag": {"zh": "火焰", "en": "Flame"}, "i18n_type": {"zh": "场景", "en": "Scene"}, "instances": []}, {"confidence": "45.27", "type": "其他", "tag": "燃烧", "i18n_tag": {"zh": "燃烧", "en": "Burning"}, "i18n_type": {"zh": "其他", "en": "Other"}, "instances": []}, {"confidence": "41.32", "type": "职业/角色", "tag": "消防员", "i18n_tag": {"zh": "消防员", "en": "Fireman"}, "i18n_type": {"zh": "职业/角色", "en": "Career/Role"}, "instances":
```

```
[], {"confidence": "39.97", "type": "动作", "tag": "烧香", "i18n_tag": {"zh": "烧香", "en": "Incense burning"}, "i18n_type": {"zh": "动作", "en": "Action"}, "instances": []}]}
```

## 9.3 名人识别示例

本章节对名人识别AK/SK方式使用SDK进行示例说明。

- 名人识别示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用初始化CelebrityRecognitionReq中的Image或Url配置图像信息（image和url参数二选一），示例中以url为例，配置完成后运行即可。
- endpoint是华为云各服务应用区域和各服务的终端节点，详情请查看[地区和终端节点](#)。

### 1. 名人识别调用示例代码如下：

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    image "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/image/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/image/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/image/v2/region"
)

func main() {
    // 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中
    // 密文存放，使用时解密，确保安全；
    // 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
    // HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK。
    ak := os.Getenv("HUAWEICLOUD_SDK_AK")
    sk := os.Getenv("HUAWEICLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := image.NewImageClient(
        image.ImageClientBuilder().
        WithRegion(region.ValueOf("cn-north-1")). //此处替换为您开通服务的区域，详情请查看地区和终端节点
        WithCredential(auth).
        Build())

    request := &model.RunCelebrityRecognitionRequest{
        thresholdCelebrityRecognitionReq:= float32(0.48)
        urlCelebrityRecognitionReq:= "https://XXX.jpg" //此处替换为公网可以访问的图片地址
        request.Body = &model.CelebrityRecognitionReq{
            Threshold: &thresholdCelebrityRecognitionReq,
            Url: &urlCelebrityRecognitionReq,
        }
    }
    response, err := client.RunCelebrityRecognition(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

### 2. 识别结果输出到控制台表示执行成功。相关参数说明请参考[名人识别API](#)。

```
RunCelebrityRecognitionResponse {"result":[{"confidence":0.9990679,"face_detail":
{"h":510,"w":357,"x":260,"y":201},"label":"迈克尔·杰克逊"}]}
Process exiting with code: 0
```

## 9.4 主体识别示例

本章节对主体识别AK/SK方式使用SDK进行示例说明。

- 主体识别示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用初始化 ImageMainObjectDetectionReq 中的 Image 或 Url 配置图像信息（image 和 url 参数二选一），示例中以 url 为例，配置完成后运行即可。
- endpoint 是华为云各服务应用区域和各服务的终端节点，详情请查看[地区和终端节点](#)。

1. 主体识别调用示例代码如下：

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    image "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/image/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/image/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/image/v2/region"
)

func main() {
    // 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中
    // 密文存放，使用时解密，确保安全；
    // 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
    // HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK。
    ak := os.Getenv("HUAWEICLOUD_SDK_AK")
    sk := os.Getenv("HUAWEICLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := image.NewImageClient(
        image.ImageClientBuilder().
        WithRegion(region.ValueOf("cn-north-4")). //此处替换为您开通服务的区域，详情请查看地区和终端节点
        WithCredential(auth).
        Build())

    request := &model.RunImageMainObjectDetectionRequest{
        thresholdImageMainObjectDetectionReq:= float32(30)
        urlImageMainObjectDetectionReq:= "https://XXX.jpg" //此处替换为公网可以访问的图片地址
        request.Body = &model.ImageMainObjectDetectionReq{
            Threshold: &thresholdImageMainObjectDetectionReq,
            Url: &urlImageMainObjectDetectionReq,
        }
    }
    response, err := client.RunImageMainObjectDetection(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

2. 识别结果输出到控制台表示执行成功。相关参数说明请参考[主体识别API](#)。

```
RunImageMainObjectDetectionResponse {"result":[{"label":"bounding_box","location":{"height":327.77,"top_left_x":3.7,"top_left_y":1.72,"width":463.03},"confidence":"95.73"}, {"label":"main_object_box","location":{"height":328.35,"top_left_x":3.22,"top_left_y":1.5,"width":456.63},"confidence":"88.35"}]}
```



## 9.5 翻拍识别示例

本章节对翻拍识别AK/SK方式使用SDK进行示例说明。

- 翻拍识别示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用初始化RunRecaptureDetectReq中的Image或Url配置图像信息（image和url参数二选一），示例中以url为例，配置完成后运行即可。
- endpoint是华为云各服务应用区域和各服务的终端节点，详情请查看[地区和终端节点](#)。

1. 翻拍识别调用示例代码如下：

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    image "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/image/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/image/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/image/v2/region"
)

func main() {
    // 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中
    // 密文存放，使用时解密，确保安全；
    // 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
    // HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK。
    ak := os.Getenv("HUAWEICLOUD_SDK_AK")
    sk := os.Getenv("HUAWEICLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := image.NewImageClient(
        image.ImageClientBuilder().
        WithRegion(region.ValueOf("cn-north-4")). //此处替换为您开通服务的区域，详情请查看地区和终端节点
        WithCredential(auth).
        Build())

    request := &model.RunRecaptureDetectRequest{
        thresholdRecaptureDetectReq:= float32(0)
        urlRecaptureDetectReq:= "https://bucketname.obs.cn-north-4.myhuaweicloud.com/image/test1.jpg"//替换为公网可以访问的图片地址
        request.Body = &model.RecaptureDetectReq{
            Threshold: &thresholdRecaptureDetectReq,
            Url: &urlRecaptureDetectReq,
        }
    }
    response, err := client.RunRecaptureDetect(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

2. 识别结果输出到控制台表示执行成功。

```
{ "result": { "suggestion": "false", "category": "recapture", "score": "0.9998", "detail": [ { "label": "recapture", "confidence": "0.9998" } ] } }
```

## 9.6 图像标签示例

本章节对图像标签AK/SK方式使用SDK进行示例说明。

- 图像标签示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用初始化 ImageTaggingReq 中的 Image 或 Url 配置图像信息（image 和 url 参数二选一），示例中以 url 为例，配置完成后运行即可。
- endpoint 是华为云各服务应用区域和各服务的终端节点，详情请查看 [地区和终端节点](#)。

1. 图像标签调用示例代码如下：

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    image "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/image/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/image/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/image/v2/region"
)

func main() {
    // 认证的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中
    // 密文存放，使用时解密，确保安全；
    // 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
    // HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK。
    ak := os.Getenv("HUAWEICLOUD_SDK_AK")
    sk := os.Getenv("HUAWEICLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := image.NewImageClient(
        image.ImageClientBuilder().
            WithRegion(region.ValueOf("cn-north-1")). //此处替换为您开通服务的区域，详情请查看地区和终端节点
            WithCredential(auth).
            Build())

    request := &model.RunImageTaggingRequest{
        limitImageTaggingReq:= int32(50)
        thresholdImageTaggingReq:= float32(95)
        languageImageTaggingReq:= "zh"
        urlImageTaggingReq:= "https://XXX.jpg" //此处替换为公网可以访问的图片地址
        request.Body = &model.ImageTaggingReq{
            Limit: &limitImageTaggingReq,
            Threshold: &thresholdImageTaggingReq,
            Language: &languageImageTaggingReq,
            Url: &urlImageTaggingReq,
        }
    }
    response, err := client.RunImageTagging(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

2. 识别结果输出到控制台表示执行成功。相关参数说明请参考 [图像标签API](#)。  
 RunImageTaggingResponse {"result": {"tags": [{"confidence": "98.01", "type": "树", "tag": "树", "i18n\_tag": {"zh": "树", "en": "Tree"}}, {"i18n\_type": {"zh": "树", "en": "Tree"}, "instances": []]}}  
 Process exiting with code: 0

# 10 使用 SDK ( PHP )

## 10.1 PHP 开发环境配置

### 准备工作

- [注册华为账号并开通华为云](#)，并完成实名认证，账号不能处于欠费或冻结状态。
- 已具备开发环境，PHP 5.6 及以上版本，可执行 `php --version` 检查当前的版本信息。
- 登录“[我的凭证](#)> 访问密钥”页面，获取 Access Key ( AK ) 和 Secret Access Key ( SK )。

图 10-1 获取 AK、SK



- 登录“[我的凭证](#)”页面，获取“IAM用户名”、“账号名”以及待使用区域的“项目ID”。调用服务时会用到这些信息，请提前保存。  
本样例以“华北-北京四”区域为例，获取对应的项目ID ( project\_id )。

图 10-2 我的凭证



### 安装 SDK

推荐使用 [Composer](#) 安装 SDK 。

Composer 是 php 的依赖管理工具，允许您在项目中声明依赖关系并安装这些依赖：

```
// 安装 Composer
curl -sS https://getcomposer.org/installer | php
// 安装 PHP SDK
composer require huaweicloud/huaweicloud-sdk-php
```

安装完毕后，你需要引入 Composer 的自动加载文件：

```
require 'path/to/vendor/autoload.php';
```

## 开始使用 SDK

### 1. 导入依赖模块

```
<?php
namespace HuaweiCloud\SDK\Image\V2\Model;
require_once "vendor/autoload.php";
use HuaweiCloud\SDK\Core\Auth\BasicCredentials;
use HuaweiCloud\SDK\Core\Http\HttpConfig;
use HuaweiCloud\SDK\Core\Exceptions\ConnectionException;
use HuaweiCloud\SDK\Core\Exceptions\RequestTimeoutException;
use HuaweiCloud\SDK\Core\Exceptions\ServiceResponseException;
use HuaweiCloud\SDK\Image\V2\ImageClient;
```

### 2. 配置客户端连接参数

#### - 默认配置

```
// 使用默认配置
$config = HttpConfig::getDefaultConfig();
```

#### - 网络代理（可选）

```
// 使用代理服务器
$config->setProxyProtocol('http');
$config->setProxyHost('proxy.huawei.com');
$config->setProxyPort(8080);
$config->setProxyUser('username');
$config->setProxyPassword('password');
```

#### - 超时配置（可选）

```
// 默认连接超时时间为60秒，读取超时时间为120秒。可根据需要修改默认值。
$config->setTimeout(120);
$config->setConnectionTimeout(60);
```

#### - SSL配置（可选）

```
// 配置跳过服务端证书验证
$config->setIgnoreSslVerification(true);
// 配置服务器端CA证书，用于SDK验证服务端证书合法性
$config->setCertFile("{yourCertFile}");
```

### 3. 配置认证信息

配置AK、SK、projectId信息。华为云通过AK识别用户的身份，通过SK对请求数据进行签名验证，用于确保请求的机密性、完整性和请求者身份的正确性。

```
$ak = getenv('HUAWEICLOUD_SDK_AK');
$sk = getenv('HUAWEICLOUD_SDK_SK');
$projectId = getenv("PROJECT_ID");
$credentials = new BasicCredentials($ak,$sk,$projectId);
```

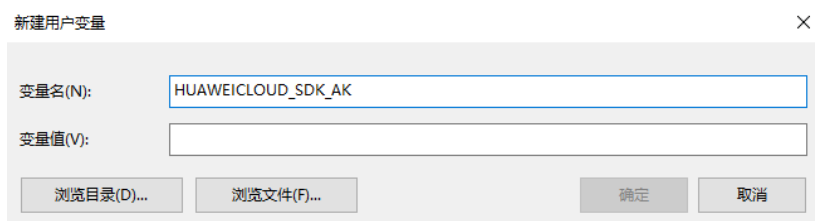
认证参数说明：

- ak、sk：访问密钥信息，获取方法请参见[准备工作](#)。
- projectId：华为云项目ID，获取方法请参见[准备工作](#)。
- endpoint：华为云各服务应用区域和各服务的终端节点，详情请查看[地区和终端节点](#)。

**注意**

- 认证用的 ak 和 sk 硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。
- 本示例以 ak 和 sk 保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量 HUAWEICLOUD\_SDK\_AK，HUAWEICLOUD\_SDK\_SK 和 PROJECT\_ID。

图 10-3 Windows 环境新建环境变量



4. 初始化客户端

指定云服务 endpoint，初始化客户端

```
$endpoint = "https://image.cn-north-4.myhuaweicloud.com";
$client = ImageClient::newBuilder(new ImageClient)
    ->withHttpConfig($config)
    ->withEndpoint($endpoint)
    ->withCredentials($credentials)
    ->build();
```

5. 发送并查看响应

```
$request = new RunImageMediaTaggingRequest();
$body = new ImageMediaTaggingReq();
$body->setLimit(10);
$body->setThreshold(60);
$body->setLanguage("en");
$body->setImage("图片Base64");
$request->setBody($body);
try {
    $response = $client->RunImageMediaTagging($request);
    echo "\n";
    echo $response;
} catch (ConnectionException $e) {
    $msg = $e->getMessage();
    echo "\n. $msg .\n";
} catch (RequestTimeoutException $e) {
    $msg = $e->getMessage();
    echo "\n. $msg .\n";
} catch (ServiceResponseException $e) {
    echo "\n";
    echo $e->getStatusCode(). "\n";
    echo $e->getRequestId(). "\n";
    echo $e->getErrorCode(). "\n";
    echo $e->getErrMsg(). "\n";
}
```

6. 异常处理

表 10-1 异常处理

| 一级分类                     | 一级分类说明  | 二级分类                     | 二级分类说明                     |
|--------------------------|---------|--------------------------|----------------------------|
| ConnectionException      | 连接类异常   | HostUnreachableException | 网络不可达、被拒绝。                 |
|                          |         | SslHandShakeException    | SSL认证异常。                   |
| RequestTimeoutException  | 响应超时异常  | CallTimeoutException     | 单次请求，服务器处理超时未返回。           |
|                          |         | RetryOutageException     | 在重试策略消耗完成后，仍无有效的响应。        |
| ServiceResponseException | 服务器响应异常 | ServerResponseException  | 服务端内部错误，Http响应码：[500,]。    |
|                          |         | ClientRequestException   | 请求参数不合法，Http响应码：[400, 500) |

```
// 捕获和处理不同类型的异常
try {
    $response = $client->RunImageMediaTagging($request);
    echo "\n";
    echo $response;
} catch (ConnectionException $e) {
    $msg = $e->getMessage();
    echo "\n". $msg ."\n";
} catch (RequestTimeoutException $e) {
    $msg = $e->getMessage();
    echo "\n". $msg ."\n";
} catch (ServiceResponseException $e) {
    echo "\n";
    echo $e->getHttpStatusCode(). "\n";
    echo $e->getRequestId(). "\n";
    echo $e->getErrorCode() . "\n";
    echo $e->getErrMsg() . "\n";
}
}
```

### 📖 说明

使用异步客户端，配置日志等操作请参见[SDK中心](#)、[PHP SDK使用指导](#)、[PHP SDK使用视频](#)。

## 10.2 媒资图像标签示例

本章节对媒资图像标签AK/SK方式使用SDK进行示例说明。

- 媒资图像标签示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用ImageMediaTaggingReq中的Url或Image方法配置图像信息（image和url参数二选一），示例中以url为例，配置完成后运行即可。
- endpoint是华为云各服务应用区域和各服务的终端节点，详情请查看[地区和终端节点](#)。

1. 媒资图像标签调用示例代码如下：

```
<?php
namespace HuaweiCloud\SDK\Image\V2\Model;
require_once "vendor/autoload.php";
use HuaweiCloud\SDK\Core\Auth\BasicCredentials;
use HuaweiCloud\SDK\Core\Http\HttpConfig;
use HuaweiCloud\SDK\Core\Exceptions\ConnectionException;
use HuaweiCloud\SDK\Core\Exceptions\RequestTimeoutException;
use HuaweiCloud\SDK\Core\Exceptions\ServiceResponseException;
use HuaweiCloud\SDK\Image\V2\ImageClient;
// 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中
// 密文存放，使用时解密，确保安全；
// 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
// CLOUD_SDK_AK和CLOUD_SDK_SK。
$ak = getenv('CLOUD_SDK_AK');
$sk = getenv('CLOUD_SDK_SK');
//image.cn-north-4.myhuaweicloud.com替换为您开通服务区域的终端节点，详情请查看地区和终端节点。
$endpoint = "https://image.cn-north-4.myhuaweicloud.com";
$projectId = "";
$credentials = new BasicCredentials($ak,$sk,$projectId);
$config = HttpConfig::getDefaultConfig();
$config->setIgnoreSslVerification(true);
$client = ImageClient::newBuilder(new ImageClient)
->withHttpConfig($config)
->withEndpoint($endpoint)
->withCredentials($credentials)
->build();
$request = new RunImageMediaTaggingRequest();
$body = new ImageMediaTaggingReq();
$body->setLimit(10);
$body->setThreshold(60);
$body->setLanguage("zh");
$body->setImage("输入图片,使用base64编码");
$request->setBody($body);
try {
    $response = $client->RunImageMediaTagging($request);
    echo "\n";
    echo $response;
} catch (ConnectionException $e) {
    $msg = $e->getMessage();
    echo "\n". $msg ."\n";
} catch (RequestTimeoutException $e) {
    $msg = $e->getMessage();
    echo "\n". $msg ."\n";
} catch (ServiceResponseException $e) {
    echo "\n";
    echo $e->getHttpStatusCode(). "\n";
    echo $e->getRequestId(). "\n";
    echo $e->getErrorCode(). "\n";
    echo $e->getErrMsg(). "\n";
}
}
```

2. 执行示例代码文件，控制台输出媒资图像标签结果即表示程序执行成功。相关参数说明请参考[媒资图像标签API](#)。

```
{
  "result": {
    "tags": [
      {
        "confidence": "83.15",
        "type": "社会",
        "tag": "火灾",
        "i18n_tag": {
          "zh": "火灾",
          "en": "Fire"
        },
        "i18n_type": {
          "zh": "社会",
          "en": "Society"
        },
        "instances": []
      },
      {
        "confidence": "70.21",
        "type": "其他",
        "tag": "火",
        "i18n_tag": {
          "zh": "火",
          "en": "Fire"
        },
        "i18n_type": {
          "zh": "其他",
          "en": "Other"
        },
        "instances": []
      },
      {
        "confidence": "53.63",
        "type": "学科科学",
        "tag": "火球",
        "i18n_tag": {
          "zh": "火球",
          "en": "Fireball"
        },
        "i18n_type": {
          "zh": "学科科学",
          "en": "Subject science"
        },
        "instances": []
      },
      {
        "confidence": "53.45",
        "type": "其他",
        "tag": "营火",
        "i18n_tag": {
          "zh": "营火",
          "en": "Campfire"
        },
        "i18n_type": {
          "zh": "其他",
          "en": "Other"
        },
        "instances": []
      },
      {
        "confidence": "50.73",
        "type": "人工场景",
        "tag": "林业局",
        "i18n_tag": {
          "zh": "林业局",
          "en": "Forestry bureau"
        },
        "i18n_type": {
          "zh": "人工场景",
          "en": "Manual scenario"
        },
        "instances": []
      },
      {
        "confidence": "50.34",
        "type": "其他",
        "tag": "烧毁的",
        "i18n_tag": {
          "zh": "烧毁的",
          "en": "Burned"
        },
        "i18n_type": {
          "zh": "其他",
          "en": "Other"
        },
        "instances": []
      },
      {
        "confidence": "46.63",
        "type": "场景",
        "tag": "火焰",
        "i18n_tag": {
          "zh": "火焰",
          "en": "Flame"
        },
        "i18n_type": {
          "zh": "场景",
          "en": "Scene"
        },
        "instances": []
      },
      {
        "confidence": "45.27",
        "type": "其他",
        "tag": "燃烧",
        "i18n_tag": {
          "zh": "燃烧",
          "en": "Burning"
        },
        "i18n_type": {
          "zh": "其他",
          "en": "Other"
        },
        "instances": []
      },
      {
        "confidence": "41.32",
        "type": "职业/角色",
        "tag": "消防员",
        "i18n_tag": {
          "zh": "消防员",
          "en": "Fireman"
        },
        "i18n_type": {
          "zh": "职业/角色",
          "en": "Career/Role"
        },
        "instances": []
      }
    ]
  }
}
```

```
{"confidence": "39.97", "type": "动作", "tag": "烧香", "i18n_tag": {"zh": "烧香", "en": "Incense burning"}, "i18n_type": {"zh": "动作", "en": "Action"}, "instances": []}]}
```

## 10.3 翻拍识别示例

本章节对翻拍识别AK/SK方式使用SDK进行示例说明。

- 翻拍识别示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用初始化RunRecaptureDetectReq中的Image或Url配置图像信息（image和url参数二选一），示例中以url为例，配置完成后运行即可。
- endpoint是华为云各服务应用区域和各服务的终端节点，详情请查看[地区和终端节点](#)。

### 1. 翻拍识别调用示例代码如下：

```
<?php
namespace HuaweiCloud\SDK\Image\V2\Model;
require_once "vendor/autoload.php";
use HuaweiCloud\SDK\Core\Auth\BasicCredentials;
use HuaweiCloud\SDK\Core\Http\HttpConfig;
use HuaweiCloud\SDK\Core\Exceptions\ConnectionException;
use HuaweiCloud\SDK\Core\Exceptions\RequestTimeoutException;
use HuaweiCloud\SDK\Core\Exceptions\ServiceResponseException;
use HuaweiCloud\SDK\Image\V2\ImageClient;
// 认证的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中
// 密文存放，使用时解密，确保安全；
// 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
// CLOUD_SDK_AK和CLOUD_SDK_SK。
$ak = getenv('CLOUD_SDK_AK');
$sk = getenv('CLOUD_SDK_SK');
//image.cn-north-4.myhuaweicloud.com替换为您开通服务区域的终端节点
$endpoint = "https://image.cn-north-4.myhuaweicloud.com";
$projectId = "";
$credentials = new BasicCredentials($ak,$sk,$projectId);
$config = HttpConfig::getDefaultConfig();
$config->setIgnoreSslVerification(true);
$client = ImageClient::newBuilder(new ImageClient)
    ->withHttpConfig($config)
    ->withEndpoint($endpoint)
    ->withCredentials($credentials)
    ->build();
$request = new RunRecaptureDetectRequest();
$body = new RecaptureDetectReq();
$body->setThreshold(0.9);
$body->setImage("输入图片,使用base64编码");
$request->setBody($body);
try {
    $response = $client->RunRecaptureDetect($request);
    echo "\n";
    echo $response;
} catch (ConnectionException $e) {
    $msg = $e->getMessage();
    echo "\n". $msg ."\n";
} catch (RequestTimeoutException $e) {
    $msg = $e->getMessage();
    echo "\n". $msg ."\n";
} catch (ServiceResponseException $e) {
    echo "\n";
    echo $e->getStatusCode(). "\n";
    echo $e->getRequestId(). "\n";
    echo $e->getErrorCode(). "\n";
    echo $e->getErrMsg(). "\n";
}
}
```

### 2. 识别结果输出到控制台表示执行成功。

```
{"result":{"suggestion":"false","category":"recapture","score":"0.9998","detail":[{"label":"recapture","confidence":"0.9998"}]}}
```



## 10.4 图像标签示例

本章节对图像标签AK/SK方式使用SDK进行示例说明。

- 图像标签示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用初始化ImageTaggingReq中的Image或Url配置图像信息（image和url参数二选一），示例中以url为例，配置完成后运行即可。
- endpoint是华为云各服务应用区域和各服务的终端节点，详情请查看[地区和终端节点](#)。

1. 图像标签调用示例代码如下：

```
<?php
namespace HuaweiCloud\SDK\Image\V2\Model;
require_once "vendor/autoload.php";
use HuaweiCloud\SDK\Core\Auth\BasicCredentials;
use HuaweiCloud\SDK\Core\Http\HttpConfig;
use HuaweiCloud\SDK\Core\Exceptions\ConnectionException;
use HuaweiCloud\SDK\Core\Exceptions\RequestTimeoutException;
use HuaweiCloud\SDK\Core\Exceptions\ServiceResponseException;
use HuaweiCloud\SDK\Image\V2\ImageClient;
// 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中
// 密文存放，使用时解密，确保安全；
// 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
// CLOUD_SDK_AK和CLOUD_SDK_SK。
$ak = getenv('CLOUD_SDK_AK');
$sk = getenv('CLOUD_SDK_SK');
//image.cn-north-4.myhuaweicloud.com替换为您开通服务区域的终端节点
$endpoint = "https://image.cn-north-4.myhuaweicloud.com";
$projectId = "";
$credentials = new BasicCredentials($ak,$sk,$projectId);
$config = HttpConfig::getDefaultConfig();
$config->setIgnoreSslVerification(true);
$client = ImageClient::newBuilder(new ImageClient)
->withHttpConfig($config)
->withEndpoint($endpoint)
->withCredentials($credentials)
->build();
$request = new RunImageTaggingRequest();
$body = new ImageTaggingReq();
$body->setLimit(10);
$body->setThreshold(60);
$body->setLanguage("zh");
$body->setImage("输入图片,使用base64编码");
$request->setBody($body);
try {
    $response = $client->RunImageTagging($request);
    echo "\n";
    echo $response;
} catch (ConnectionException $e) {
    $msg = $e->getMessage();
    echo "\n". $msg ."\n";
} catch (RequestTimeoutException $e) {
    $msg = $e->getMessage();
    echo "\n". $msg ."\n";
} catch (ServiceResponseException $e) {
    echo "\n";
    echo $e->getStatusCode(). "\n";
    echo $e->getRequestId(). "\n";
    echo $e->getErrorCode(). "\n";
    echo $e->getErrMsg(). "\n";
}
}
```

2. 识别结果输出到控制台表示执行成功。相关参数说明请参考[图像标签API](#)。

```
RunImageTaggingResponse {"result": {"tags": [{"confidence": "98.01", "type": "树", "tag": "树",  
"i18n_tag": {"zh": "树", "en": "Tree"}, "i18n_type": {"zh": "树", "en": "Tree"}, "instances": []]}}  
Process exiting with code: 0
```

# 11 使用 SDK (C++)

## 11.1 C++开发环境配置

### 准备工作

- [注册华为账号并开通华为云](#)，并完成实名认证，账号不能处于欠费或冻结状态。
- 已具备开发环境，支持 C++ 14 及以上版本，要求安装 CMake 3.10 及以上版本。
- 登录“[我的凭证](#) > 访问密钥”页面，获取Access Key (AK) 和Secret Access Key (SK)。

图 11-1 获取 AK、SK



- 登录“[我的凭证](#)”页面，获取“IAM用户名”、“账号名”以及待使用区域的“项目ID”。调用服务时会用到这些信息，请提前保存。  
本样例以“华北-北京四”区域为例，获取对应的项目ID (project\_id)。

图 11-2 我的凭证



### 安装 SDK

- 在Linux系统上安装SDK
  - a. 获取依赖包  
所需的这些第三方软件包在大部分系统的包管理工具中都有提供，例如基于 Debian/Ubuntu 的系统。

```
sudo apt-get install libcurl4-openssl-dev libboost-all-dev libssl-dev libcpprest-dev
```

spdlog 需要从源码进行安装。

```
git clone https://github.com/gabime/spdlog.git
cd spdlog
mkdir build
cd build
cmake -DCMAKE_POSITION_INDEPENDENT_CODE=ON .. // 用以生成动态库
make
sudo make install
```

b. 编译安装

```
git clone https://github.com/huaweicloud/huaweicloud-sdk-cpp-v3.git
cd huaweicloud-sdk-cpp-v3
mkdir build
cd build
cmake ..
make
sudo make install
```

完成上述操作后，C++ SDK 安装目录为 /usr/local。

- 在Windows系统上安装SDK

a. 安装 vcpkg 并使用 vcpkg 安装所需软件包

```
vcpkg install curl cpprestsdk boost openssl spdlog
```

b. 使用CLion进行编译

i. 使用CLion打开huaweicloud-sdk-cpp-v3 目录。

ii. 选择“File > Settings”。

iii. 选择“Build, Execution, Deployment >> CMake”。

iv. 在CMake options中加入：

```
-DCMAKE_TOOLCHAIN_FILE={your vcpkg install dir}/scripts/buildsystems/vcpkg.cmake
```

v. 右键 CMakeLists.txt 选择 Load CMake Project。

vi. 选择Build开始编译。

c. 安装C++ SDK

编译完成后选择“Build > Install”。

完成上述操作后，C++ SDK 安装目录为 C:\Program File (x86)\huaweicloud-sdk-cpp-v3。

## 开始使用 SDK

1. 导入依赖模块

```
#include <cstdlib>
#include <iostream>
#include <string>
#include <memory>
#include <huaweicloud/core/exception/Exceptions.h>
#include <huaweicloud/core/Client.h>
#include <huaweicloud/image/v2/ImageClient.h>
using namespace HuaweiCloud::Sdk::Image::V2;
using namespace HuaweiCloud::Sdk::Image::V2::Model;
using namespace HuaweiCloud::Sdk::Core;
using namespace HuaweiCloud::Sdk::Core::Exception;
using namespace std;
```

2. 配置客户端连接参数

- 默认配置

```
// 使用默认配置
HttpConfig httpConfig = HttpConfig();
```

- 网络代理（可选）

```
// 根据需要配置网络代理
httpConfig.setProxyProtocol("http");
httpConfig.setProxyHost("proxy.huawei.com");
httpConfig.setProxyPort("8080");
httpConfig.setProxyUser("username");
httpConfig.setProxyPassword("password");
```

- 超时配置（可选）

```
// 默认连接超时为60秒，默认读取超时为120秒。可根据需求修改该默认值
httpConfig.setConnectTimeout(60);
httpConfig.setReadTimeout(120);
```

- SSL配置（可选）

```
// 配置跳过服务端证书验证
httpConfig.setIgnoreSslVerification(true);
```

### 3. 配置认证信息

配置AK、SK、projectId信息。华为云通过AK识别用户的身份，通过SK对请求数据进行签名验证，用于确保请求的机密性、完整性和请求者身份的正确性。

```
string ak = getenv("HUAWEICLOUD_SDK_AK");
string sk = getenv("HUAWEICLOUD_SDK_SK");
string projectId = getenv("PROJECT_ID");
auto auth = std::make_unique<BasicCredentials>();
auth->withAk(ak)
    .withSk(sk)
    .withProjectId(projectId);
```

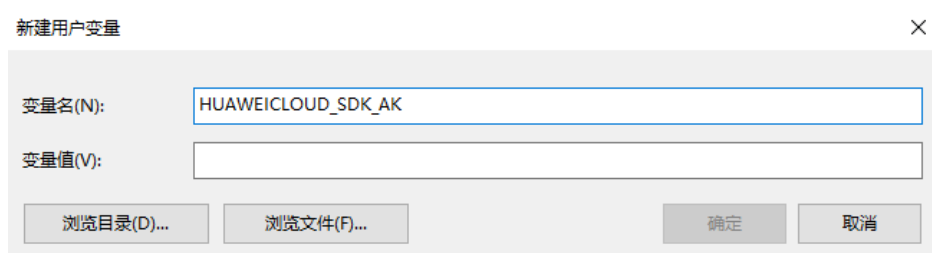
认证参数说明：

- ak、sk：访问密钥信息，获取方法请参见[准备工作](#)。
- projectId：华为云项目ID，获取方法请参见[准备工作](#)。

#### ⚠ 注意

- 认证用的 ak 和 sk 硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。
- 本示例以 ak 和 sk 保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量HUAWEICLOUD\_SDK\_AK，HUAWEICLOUD\_SDK\_SK和PROJECT\_ID。

图 11-3 Windows 环境新建环境变量



### 4. 初始化客户端

指定云服务endpoint，初始化客户端

```
string endpoint = "https://image.cn-north-4.myhuaweicloud.com";
auto client = ImageClient::newBuilder()
    .withCredentials(std::unique_ptr<Credentials>(auth.release()))
    .withHttpConfig(httpConfig)
    .withEndPoint(endpoint)
    .build();
```

endpoint：华为云各服务应用区域和各服务的终端节点，详情请查看[地区和终端节点](#)。

5. 发送请求并查看响应  
// 以调用接口RunImageMediaTagging为例  
RunImageMediaTaggingRequest request;  
ImageMediaTaggingReq body;  
body.setLimit(10);  
body.setThreshold(60);  
body.setLanguage("zh");  
body.setImage("输入图片,使用base64编码");  
request.setBody(body);
6. 异常处理

表 11-1 异常处理

| 一级分类                         | 一级分类说明      | 二级分类                         | 二级分类说明                            |
|------------------------------|-------------|------------------------------|-----------------------------------|
| ConnectionExcepti<br>on      | 连接类异常       | HostUnreachableE<br>xception | 网络不可达、被拒<br>绝。                    |
|                              |             | SslHandShakeExce<br>ption    | SSL认证异常。                          |
| RequestTimeoutEx<br>ception  | 响应超时异常      | CallTimeoutExcept<br>ion     | 单次请求，服务器<br>处理超时未返回。              |
|                              |             | RetryOutageExcep<br>tion     | 在重试策略消耗完<br>成后，仍无有效的<br>响应。       |
| ServiceResponseEx<br>ception | 服务器响应异<br>常 | ServerResponseEx<br>ception  | 服务端内部错误，<br>Http响应码：<br>[500,]。   |
|                              |             | ClientRequestExce<br>ption   | 请求参数不合法，<br>Http响应码：[400,<br>500) |

```
// 捕获和处理不同类型的异常
try {
    auto reponse = client->runImageMediaTagging(request);
    std::cout << response->getHttpBody() << std::endl;
} catch (HostUnreachableException& e) {
    std::cout << "host unreachable:" << e.what() << std::endl;
} catch (SslHandShakeException& e) {
    std::cout << "ssl handshake error:" << e.what() << std::endl;
} catch (RetryOutageException& e) {
    std::cout << "retryoutage error:" << e.what() << std::endl;
} catch (CallTimeoutException& e) {
    std::cout << "call timeout:" << e.what() << std::endl;
} catch (ServiceResponseException& e) {
    std::cout << "http status code:" << e.getStatusCode() << std::endl;
    std::cout << "error code:" << e.getErrorCode() << std::endl;
    std::cout << "error msg:" << e.getErrorMsg() << std::endl;
    std::cout << "RequestId:" << e.getRequestId() << std::endl;
} catch (exception& e) {
    std::cout << "unknown exception:" << e.what() << std::endl;
}
std::cout << "-----request finished-----" << std::endl;
```

## 📖 说明

使用异步客户端，配置日志等操作请参见[SDK中心](#)、[C++ SDK使用指导](#)。

## 11.2 媒资图像标签示例

本章节对媒资图像标签AK/SK方式使用SDK进行示例说明。

- 媒资图像标签示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用ImageMediaTaggingReq中的Url或Image方法配置图像信息（image和url参数二选一），示例中以url为例，配置完成后运行即可。
- endpoint是华为云各服务应用区域和各服务的终端节点，详情请查看[地区和终端节点](#)。

1. 媒资图像标签调用示例代码如下：

```
#include <cstdlib>
#include <iostream>
#include <string>
#include <memory>
#include <huaweicloud/core/exception/Exceptions.h>
#include <huaweicloud/core/Client.h>
#include <huaweicloud/image/v2/ImageClient.h>
using namespace HuaweiCloud::Sdk::Image::V2;
using namespace HuaweiCloud::Sdk::Image::V2::Model;
using namespace HuaweiCloud::Sdk::Core;
using namespace HuaweiCloud::Sdk::Core::Exception;
using namespace std;
int main(void)
{
    // 认证的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中
    // 密文存放，使用时解密，确保安全；
    // 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
    // CLOUD_SDK_AK和CLOUD_SDK_SK。
    string ak = getenv("CLOUD_SDK_AK");
    string sk = getenv("CLOUD_SDK_SK");
    string projectId = "";
    //image.cn-north-4.myhuaweicloud.com替换为您开通服务区域的终端节点
    string endpoint = "https://image.cn-north-4.myhuaweicloud.com";
    auto auth = std::make_unique<BasicCredentials>();
    auth->withAk(ak)
        .withSk(sk)
        .withProjectId(projectId);
    HttpConfig httpConfig = HttpConfig();
    auto client = ImageClient::newBuilder()
        .withCredentials(std::unique_ptr<Credentials>(auth.release()))
        .withHttpConfig(httpConfig)
        .withEndPoint(endpoint)
        .build();
    RunImageMediaTaggingRequest request;
    ImageMediaTaggingReq body;
    body.setLimit(10);
    body.setThreshold(60);
    body.setLanguage("zh");
    body.setImage("输入图片,使用base64编码");
    request.setBody(body);
    std::cout << "-----begin execute request-----" << std::endl;
    try {
        auto reponse = client->runImageMediaTagging(request);
        std::cout << reponse->getHttpBody() << std::endl;
    } catch (HostUnreachableException& e) {
        std::cout << "host unreachable." << e.what() << std::endl;
    } catch (SslHandShakeException& e) {
        std::cout << "ssl handshake error:" << e.what() << std::endl;
    } catch (RetryOutageException& e) {
        std::cout << "retryoutage error:" << e.what() << std::endl;
    }
```

```

} catch (CallTimeoutException& e) {
    std::cout << "call timeout:" << e.what() << std::endl;
} catch (ServiceResponseException& e) {
    std::cout << "http status code:" << e.getStatusCode() << std::endl;
    std::cout << "error code:" << e.getErrorCode() << std::endl;
    std::cout << "error msg:" << e.getErrorMsg() << std::endl;
    std::cout << "RequestId:" << e.getRequestId() << std::endl;
} catch (exception& e) {
    std::cout << "unknown exception:" << e.what() << std::endl;
}
std::cout << "-----request finished-----" << std::endl;
return 0;
}

```

2. 执行示例代码文件，控制台输出媒资图像标签结果即表示程序执行成功。相关参数说明请参考[媒资图像标签API](#)。

```

RunImageMediaTaggingResponse {"result": {"tags": [{"confidence": "83.15", "type": "社会", "tag": "火灾", "i18n_tag": {"zh": "火灾", "en": "Fire"}, "i18n_type": {"zh": "社会", "en": "Society"}, "instances": []}, {"confidence": "70.21", "type": "其他", "tag": "火", "i18n_tag": {"zh": "火", "en": "Fire"}, "i18n_type": {"zh": "其他", "en": "Other"}, "instances": []}, {"confidence": "53.63", "type": "学科科学", "tag": "火球", "i18n_tag": {"zh": "火球", "en": "Fireball"}, "i18n_type": {"zh": "学科科学", "en": "Subject science"}, "instances": []}, {"confidence": "53.45", "type": "其他", "tag": "营火", "i18n_tag": {"zh": "营火", "en": "Campfire"}, "i18n_type": {"zh": "其他", "en": "Other"}, "instances": []}, {"confidence": "50.73", "type": "人工场景", "tag": "林业局", "i18n_tag": {"zh": "林业局", "en": "Forestry bureau"}, "i18n_type": {"zh": "人工场景", "en": "Manual scenario"}, "instances": []}, {"confidence": "50.34", "type": "其他", "tag": "烧毁的", "i18n_tag": {"zh": "烧毁的", "en": "Burned"}, "i18n_type": {"zh": "其他", "en": "Other"}, "instances": []}, {"confidence": "46.63", "type": "场景", "tag": "火焰", "i18n_tag": {"zh": "火焰", "en": "Flame"}, "i18n_type": {"zh": "场景", "en": "Scene"}, "instances": []}, {"confidence": "45.27", "type": "其他", "tag": "燃烧", "i18n_tag": {"zh": "燃烧", "en": "Burning"}, "i18n_type": {"zh": "其他", "en": "Other"}, "instances": []}, {"confidence": "41.32", "type": "职业/角色", "tag": "消防员", "i18n_tag": {"zh": "消防员", "en": "Fireman"}, "i18n_type": {"zh": "职业/角色", "en": "Career/Role"}, "instances": []}, {"confidence": "39.97", "type": "动作", "tag": "烧香", "i18n_tag": {"zh": "烧香", "en": "Incense burning"}, "i18n_type": {"zh": "动作", "en": "Action"}, "instances": []}]}

```

## 11.3 翻拍识别示例

本章节对翻拍识别AK/SK方式使用SDK进行示例说明。

- 翻拍识别示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用初始化RunRecaptureDetectReq中的Image或Url配置图像信息（image和url参数二选一），示例中以url为例，配置完成后运行即可。
- endpoint是华为云各服务应用区域和各服务的终端节点，详情请查看[地区和终端节点](#)。

1. 翻拍识别调用示例代码如下：

```

#include <cstdlib>
#include <iostream>
#include <string>
#include <memory>
#include <huaweicloud/core/exception/Exceptions.h>
#include <huaweicloud/core/Client.h>
#include <huaweicloud/image/v2/ImageClient.h>
using namespace HuaweiCloud::Sdk::Image::V2;
using namespace HuaweiCloud::Sdk::Image::V2::Model;
using namespace HuaweiCloud::Sdk::Core;
using namespace HuaweiCloud::Sdk::Core::Exception;
using namespace std;
int main(void)
{
    // 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；
    // 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量CLOUD_SDK_AK和CLOUD_SDK_SK。
    string ak = getenv("CLOUD_SDK_AK");
    string sk = getenv("CLOUD_SDK_SK");
    string projectId = "";
}

```



```
//image.cn-north-4.myhuaweicloud.com替换为您开通服务区域的终端节点
string endpoint = "https://image.cn-north-4.myhuaweicloud.com";
auto auth = std::make_unique<BasicCredentials>();
auth->withAk(ak)
    .withSk(sk)
    .withProjectId(projectId);
HttpConfig httpConfig = HttpConfig();
auto client = ImageClient::newBuilder()
    .withCredentials(std::unique_ptr<Credentials>(auth.release()))
    .withHttpConfig(httpConfig)
    .withEndPoint(endpoint)
    .build();
RunRecaptureDetectRequest request;
RecaptureDetectReq body;
body.setThreshold(0.95);
body.setImage("输入图片,使用base64编码");
request.setBody(body);
std::cout << "-----begin execute request-----" << std::endl;
try {
    auto reponse = client->runRecaptureDetect(request);
    std::cout << reponse->getHttpBody() << std::endl;
} catch (HostUnreachableException& e) {
    std::cout << "host unreachable:" << e.what() << std::endl;
} catch (SslHandShakeException& e) {
    std::cout << "ssl handshake error:" << e.what() << std::endl;
} catch (RetryOutageException& e) {
    std::cout << "retryoutage error:" << e.what() << std::endl;
} catch (CallTimeoutException& e) {
    std::cout << "call timeout:" << e.what() << std::endl;
} catch (ServiceResponseException& e) {
    std::cout << "http status code:" << e.getStatusCode() << std::endl;
    std::cout << "error code:" << e.getErrorCode() << std::endl;
    std::cout << "error msg:" << e.getErrorMsg() << std::endl;
    std::cout << "RequestId:" << e.getRequestId() << std::endl;
} catch (exception& e) {
    std::cout << "unknown exception:" << e.what() << std::endl;
}
std::cout << "-----request finished-----" << std::endl;
return 0;
}
```

2. 识别结果输出到控制台表示执行成功。

```
{"result":{"suggestion":"false","category":"recapture","score":"0.9998","detail":
[{"label":"recapture","confidence":"0.9998"}]}
```

## 11.4 图像标签示例

本章节对图像标签AK/SK方式使用SDK进行示例说明。

- 图像标签示例代码只需将AK/SK信息替换为实际AK/SK，代码中可以使用初始化 ImageTaggingReq 中的 Image 或 Url 配置图像信息（image 和 url 参数二选一），示例中以 url 为例，配置完成后运行即可。
- endpoint 是华为云各服务应用区域和各服务的终端节点，详情请查看 [地区和终端节点](#)。

1. 图像标签调用示例代码如下：

```
#include <cstdlib>
#include <iostream>
#include <string>
#include <memory>
#include <huaweicloud/core/exception/Exceptions.h>
#include <huaweicloud/core/Client.h>
#include <huaweicloud/image/v2/ImageClient.h>
using namespace HuaweiCloud::Sdk::Image::V2;
using namespace HuaweiCloud::Sdk::Image::V2::Model;
using namespace HuaweiCloud::Sdk::Core;
```

```
using namespace HuaweiCloud::Sdk::Core::Exception;
using namespace std;
int main(void)
{
    // 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中
    // 密文存放，使用时解密，确保安全；
    // 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量
    // CLOUD_SDK_AK和CLOUD_SDK_SK。
    string ak = getenv("CLOUD_SDK_AK");
    string sk = getenv("CLOUD_SDK_SK");
    string projectId = "";
    //image.cn-north-4.myhuaweicloud.com替换为您开通服务区域的终端节点
    string endpoint = "https://image.cn-north-4.myhuaweicloud.com";
    auto auth = std::make_unique<BasicCredentials>();
    auth->withAk(ak)
        .withSk(sk)
        .withProjectId(projectId);
    HttpConfig httpConfig = HttpConfig();
    auto client = ImageClient::newBuilder()
        .withCredentials(std::unique_ptr<Credentials>(auth.release()))
        .withHttpConfig(httpConfig)
        .withEndPoint(endpoint)
        .build();
    RunImageTaggingRequest request;
    ImageTaggingReq body;
    body.setLimit(10);
    body.setThreshold(60);
    body.setLanguage("zh");
    body.setImage("输入图片,使用base64编码");
    request.setBody(body);
    std::cout << "-----begin execute request-----" << std::endl;
    try {
        auto reponse = client->runImageTagging(request);
        std::cout << reponse->getHttpBody() << std::endl;
    } catch (HostUnreachableException& e) {
        std::cout << "host unreachable:" << e.what() << std::endl;
    } catch (SslHandShakeException& e) {
        std::cout << "ssl handshake error:" << e.what() << std::endl;
    } catch (RetryOutageException& e) {
        std::cout << "retryoutage error:" << e.what() << std::endl;
    } catch (CallTimeoutException& e) {
        std::cout << "call timeout:" << e.what() << std::endl;
    } catch (ServiceResponseException& e) {
        std::cout << "http status code:" << e.getStatusCode() << std::endl;
        std::cout << "error code:" << e.getErrorCode() << std::endl;
        std::cout << "error msg:" << e.getErrorMsg() << std::endl;
        std::cout << "RequestId:" << e.getRequestId() << std::endl;
    } catch (exception& e) {
        std::cout << "unknown exception:" << e.what() << std::endl;
    }
    std::cout << "-----request finished-----" << std::endl;
    return 0;
}
```

2. 识别结果输出到控制台表示执行成功。相关参数说明请参考[图像标签API](#)。  
RunImageTaggingResponse {"result": {"tags": [{"confidence": "98.01", "type": "树", "tag": "树", "i18n\_tag": {"zh": "树", "en": "Tree"}, "i18n\_type": {"zh": "树", "en": "Tree"}, "instances": []}]}}  
Process exiting with code: 0