Image Recognition

SDK Reference

Issue	01
Date	2024-12-05

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 Introduction to Image Recognition SDK

Overview of Image Recognition

Image Recognition is a technology that uses computers to process, analyze, and understand images to identify objects in different modes, including Image Tagging.

Image Recognition provides services through open application programming interfaces (APIs). You can obtain the inference results by accessing and calling APIs in real time. It helps you collect key data automatically and build an intelligent service system, thereby improving service efficiency.

SDK Overview

Image Recognition Software Development Kit (Image SDK for short) encapsulates the RESTful APIs provided by Image Recognition to simplify application development. You can add dependencies or download SDKs to call APIs to use Image Recognition.

Mappings Between Services and APIs

Table 1-1 lists the mappings between the Image Recognition sub-services and APIs.

Table 1-1 Mappings between services and APIs

Service	API
Image Tagging	POST /v2/{project_id}/image/tagging

2 Applying for a Service

For details about how to apply for the service, see **Applying for a Service** in the *Image Recognition API Reference*.

3 Obtaining Authentication Information

Service APIs need to be authenticated. The procedure is as follows:

Step 1 Register with and log in to the HUAWEI CLOUD management console.

Step 2 Hover the cursor on the username and select **My Credentials** from the drop-down list.

Step 3 Click the **Access Keys** tab and then click **Add Access Key**.

Step 4 Enter the verification code sent to your mail or mobile phone.

Step 5 Click **OK** to download the AK/SK of the authentication account. The AK/SK data is saved as a local file. Keep the file properly.

----End

4 Preparing the Environment

Table 4-1 lists the environments you need to prepare when using Image Recognition SDKs of different programming languages.

Table 4-1 Development environment

Programming Language	Preparation	Description
Java	Installing JDK	An environment used for developing Java applications. Java JDK 1.8 and later versions are supported. You are advised to use the Java SDK by installing dependencies through Apache Maven.
Python	Installing Python	Python 3.3 and later versions are supported.
.NET	Installing .NET	.NET Standard 2.0 and later versions, and C# 4.0 and later versions are supported.
Go	Installing Go	Go 1.14 and later versions are supported.
PHP	Installing PHP	PHP 5.6 and later versions are supported.
C++	Installing C++	C++ 14 or later and CMake 3.10 or later are supported.

5 Using Java SDK

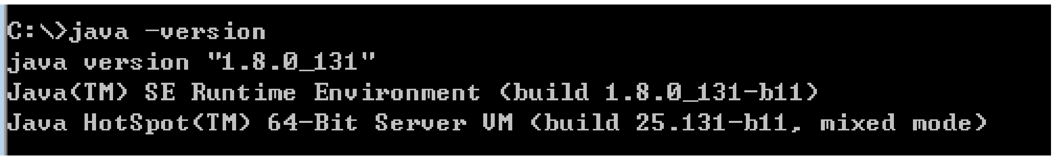
5.1 Preparing a Java Development Environment

Configuring the Environment

The Image Recognition Java SDK uses JDK 8 or later. The following uses JDK 8 (Windows x64) running on Windows 7 as an example. If you have downloaded the JDK and configured the environment, skip this section.

1. **Download the JDK file**.
2. After the JDK file is downloaded, install the JDK as prompted. For example, install the JDK to the **C:\Program Files\Java\jdk1.8.0_131** directory on the local PC.
3. Right-click **Computer**, choose **Properties** > **Advanced System Settings** > **Environment Variables**, and perform the following operations to configure Java environment variables:
   a. Create system variable **JAVA_HOME** whose value is the JDK installation path.
   b. Add **%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin** to **Path**. Separate multiple values with semicolons (;).
   c. Create system variable **CLASSPATH** whose value is **%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar**.
4. Open the command line interface (CLI) and run **java -version**. If the information shown in **Figure 5-1** is displayed, the configuration is successful.

Figure 5-1 Java version information



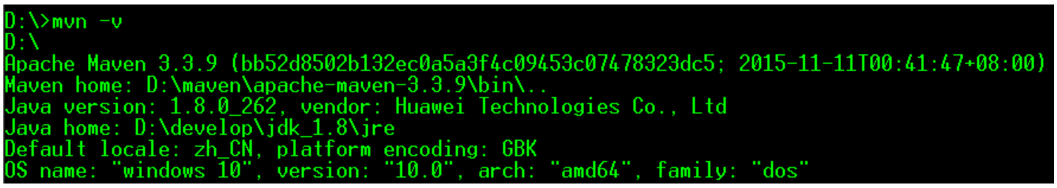
```
C:\>java -version
java version "1.8.0_131"
Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)
```

Obtaining and Installing an SDK

You are advised to use the Java SDK for Image Recognition by installing dependencies through Apache Maven.

Step 1 **Download** and **install** Apache Maven on your operating system.

Step 2 After installing and configuring Maven, run the **mvn -v** command. Maven is installed successfully if the following information is displayed.



```
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"
```

Step 3 Add dependencies to an existing Maven project or use an integrated development environment (IDE) to create a Maven project.

For example, to create a Maven project using IntelliJ IDEA, perform the following steps (skip them if you already have a Maven project):

1. Start IntelliJ IDEA.
2. Choose **File** > **New** > **project...**
3. In the displayed **New Project** dialog box, click **Maven** and then **Next**.
4. Specify **GroupId** and **ArtifactId**, and click **Next**.
5. Specify **Project name** and **Project location**, and click **Finish**.

Step 4 Add dependency items to the **pom.xml** file of the Maven project.

For example, obtain the latest version of Image Recognition SDK from **SDK Center** and use it to replace the version in the code.

```
<dependency>
    <groupId>com.huaweicloud.sdk</groupId>
    <artifactId>huaweicloud-sdk-image</artifactId>
    <version>3.1.8</version>  ///Enter the latest SDK version.
</dependency>
```

The following sample code uses the fastjson dependency (ignore and delete it if it is not required):

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

----End

5.2 Demo Project of Image Tagging

This section describes how to use the Image Tagging SDK by AK/SK authentication.

- Replace the AK/SK in the sample code with the actual AK/SK. You can use the **withUrl** or **withImage** method of the **ImageTaggingReq** class to configure image information (either the **image** or **url** parameter). This section uses **url** as an example.

- **endpoint** indicates the regions and endpoints for HUAWEI CLOUD services. For details, see **Regions and Endpoints**.

1. The sample code for calling Image Tagging is as follows:

```
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) {
       // Hard-coded or plaintext AK and SK are risky. For security purposes, encrypt your AK and SK and store them in the configuration file or environment variables.
       // In this example, the AK and SK are stored in environment variables for identity authentication. Before running this example, configure environment variables HUAWEICLOUD_SDK_AK and 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("XXXX"))   //Replace XXXX with the region where you enable the service. For details, see Regions and Endpoints.
                .build();
        RunImageTaggingRequest request = new RunImageTaggingRequest();
        ImageTaggingReq body = new ImageTaggingReq();
        body.withLimit(50);
        body.withThreshold(95f);
        body.withLanguage("zh");
        body.withUrl("https://XXX.jpg");  //Replace it with an image URL that can be accessed from the Internet.
        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. Execute the sample code. If **200** is displayed on the console, the code is successfully executed. For details about related parameters, see **Image Tagging APIs**.

```
class RunImageTaggingResponse {
    result: class ImageTaggingResponseResult {
        tags: [class ImageTaggingItemBody {
            confidence: 98.00
            type: Tree
            tag: Tree
            i18nTag: class ImageTaggingItemBodyI18nTag {
                zh: Chinese characters for tree
                en: Tree
            }
            i18nType: class ImageTaggingItemBodyI18nType {
                zh: Chinese characters for tree
                en: Tree
            }
            instances: []
        }]
    }
}

Process finished with exit code 0
```

6 Using the Python SDK

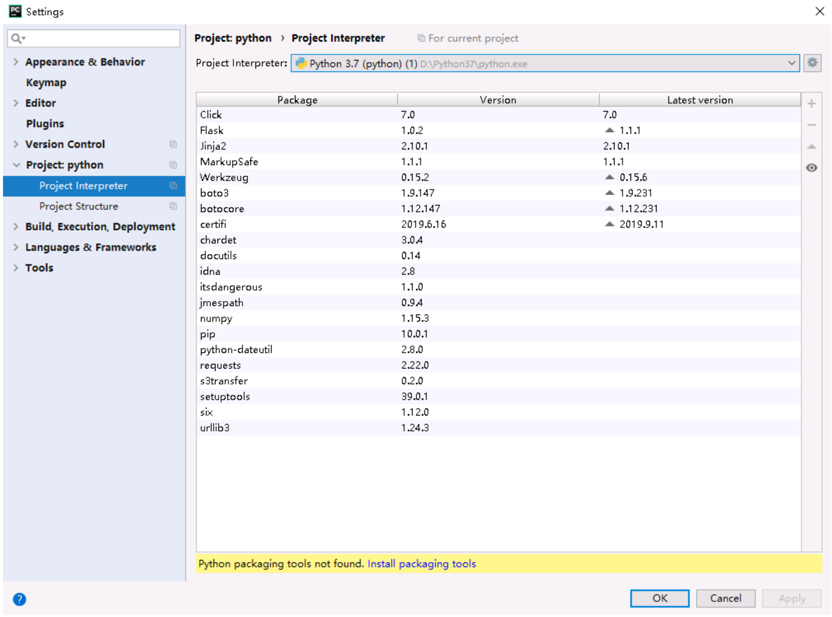
6.1 Preparing a Python Development Environment

Configuring the Environment

To use the Python SDK of Image Recognition, you need to configure the Python development environment.

1. Download Python of a proper version from **Python's official website** and install it. Python3.3 or later is recommended. This section uses Python 3.7 as an example.
2. Download the latest version of PyCharm from **PyCharm's official website**.
3. Start the PyCharm development tool and choose **File** > **Settings** > **Project Interpreter** to configure the Python environment.
4. Select the Python installation path. See **Figure 6-1**. After selecting the target Python, click **Apply** at the bottom of the page to complete the configuration.

Figure 6-1 Configuring the python environment using PyCharm



Project Interpreter: Python 3.7 (python) (1) D:\Python37\python.exe

Package	Version	Latest version
Click	7.0	7.0
Flask	1.0.2	▲ 1.1.1
Jinja2	2.10.1	2.10.1
MarkupSafe	1.1.1	1.1.1
Werkzeug	0.15.2	▲ 0.15.6
boto3	1.9.147	▲ 1.9.231
botocore	1.12.147	▲ 1.12.231
certifi	2019.6.16	▲ 2019.9.11
chardet	3.0.4	
docutils	0.14	
idna	2.8	
itsdangerous	1.1.0	
jmespath	0.9.4	
numpy	1.15.3	
pip	10.0.1	
python-dateutil	2.8.0	
requests	2.22.0	
s3transfer	0.2.0	
setuptools	39.0.1	
six	1.12.0	
urllib3	1.24.3	

Python packaging tools not found. Install packaging tools

Obtaining and Installing an SDK

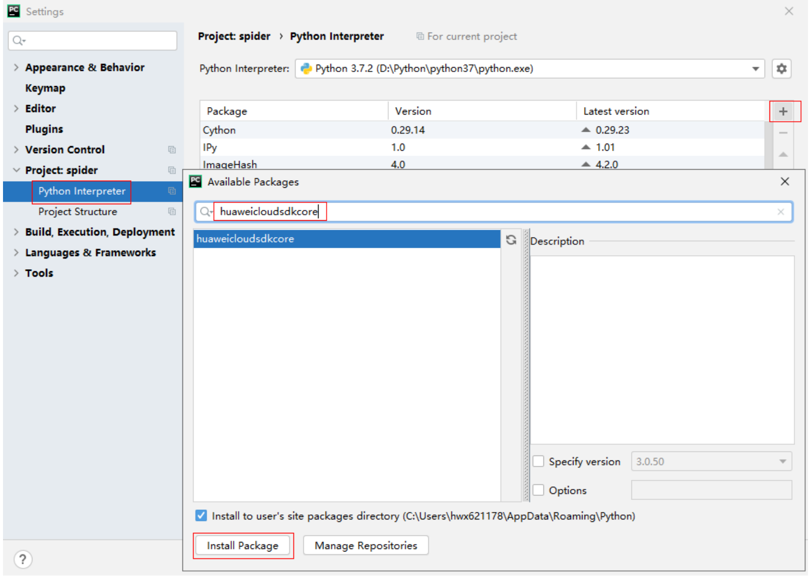
To obtain and install the Python SDK, you are advised to use pip commands or PyCharm. You need to install the **huaweicloudsdkcore** and **huaweicloudsdkimage** packages as follows:

```
Install the SDK using pip commands:
# Install the core library.
pip install huaweicloudsdkcore

# Install the image service library.
pip install huaweicloudsdkimage

To install the Python SDK using PyCharm, perform the following steps:
1. Start PyCharm and choose File > Settings...
2. Click Python Interpreter and then +.
3. Click +, search for huaweicloudsdkcore and huaweicloudsdkimage, and click Install Package in the
lower left corner to install the packages.
```

Figure 6-2 Installing the Python SDK for Image Recognition using PyCharm



Python Interpreter: Python 3.7.2 (D:\Python\python37\python.exe)

Package	Version	Latest version
Cython	0.29.14	▲ 0.29.23
IPy	1.0	▲ 1.01
ImageHash	4.0	▲ 4.2.0

Search: huaweicloudsdkcore

huaweicloudsdkcore

□ Specify version 3.0.50

□ Options

☑ Install to user's site packages directory (C:\Users\hwx621178\AppData\Roaming\Python)

6.2 Demo Project of Image Tagging

This section describes how to use the Image Tagging SDK by AK/SK authentication.

- Replace the AK/SK in the sample code with the actual AK/SK. You can use either the **url** or **image** parameter in the initialized **ImageTaggingReq** to configure image information. This section uses **url** as an example.
- **endpoint** indicates the regions and endpoints for Huawei Cloud services. For details, see **Regions and Endpoints**.

1. The sample code for calling Image Tagging is as follows:

```
# 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__":
    # Hard-coded or plaintext AK and SK are risky. For security purposes, encrypt your AK and SK and
store them in the configuration file or environment variables.
    # In this example, the AK and SK are stored in environment variables for identity authentication.
Before running this example, configure environment variables HUAWEICLOUD_SDK_AK and
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("xx-xxx-xx")) \   //Replace xx-xxx-xx with the region where
you enable the service. For details, see Regions and Endpoints.
        .build()

    try:
        request = RunImageTaggingRequest()
        request.body = ImageTaggingReq(
            limit=50,
            threshold=95,
            language="zh",
            url="https://XXX.jpg"  //Replace it with an image URL that can be accessed from the Internet.
        )
        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. Execute the sample code. If **200** is displayed on the console, the code is successfully executed. For details about related parameters, see **Image Tagging APIs**.

```
200
{"result": {"tags": [{"confidence": "98.00", "type": "Tree", "tag": "Tree", "i18n_tag": {"zh": "Chinese
characters for tree", "en": "Tree"}, "i18n_type": {"zh": "Chinese characters for tree", "en": "Tree"},
"instances": []}]}}
Process finished with exit code 0
```

7 Using the .NET SDK

7.1 Preparing a .NET Development Environment

Configuring the Environment

To use the .NET SDK of Image Recognition, you need to configure the .Net development environment.

1. Download the latest Visual Studio from the **Visual Studio official website**.
2. Click the .exe file to start installation.
3. During the installation, select **ASP.NET and web development** on the **Workloads** tab page.

Obtaining and Installing an SDK

You can install the .Net SDK using commands or Visual Studio.

```
Use .NET CLI to install the .Net SDK.
dotnet add package HuaweiCloud.SDK.Core
dotnet add package HuaweiCloud.SDK.Image

Use Package Manager to install the .Net SDK.
Install-Package HuaweiCloud.SDK.Core
Install-Package HuaweiCloud.SDK.Image
```

If you use Visual Studio, choose **Tools** > **NuGet Package Manager** > **Manage NuGet Packages for Solution...**, and search for and install **HuaweiCloud.SDK.Core** and **HuaweiCloud.SDK.Image**.

7.2 Demo Project of Image Tagging

This section describes how to use the Image Tagging SDK by AK/SK authentication.

- Replace the AK/SK in the sample code with the actual AK/SK. You can use either the **url** or **image** parameter in the initialized **ImageTaggingReq** to configure image information. This section uses **url** as an example.

- **endpoint** indicates the regions and endpoints for HUAWEI CLOUD services. For details, see **Regions and Endpoints**.

1. The sample code for calling Image Tagging is as follows:

```
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) {
       //Hard-coded or plaintext AK and SK are risky. For security purposes, encrypt your AK and SK and
store them in the configuration file or environment variables.
      //In this example, the AK and SK are stored in environment variables for identity authentication.
Before running this example, configure environment variables HUAWEICLOUD_SDK_AK and
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"))  //Replace cn-north-1 with the region
where you enable the service. For details, see Regions and Endpoints.
                    .WithHttpConfig(config)
                    .Build();

            var req = new RunImageTaggingRequest
            {
            };
            req.Body = new ImageTaggingReq()
            {
                Limit = 50,
                Threshold = 95,
                Language = "zh",
                Url = "https://XXX.jpg"    //Replace it with an image URL that can be accessed from the
Internet.
            };

            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. Execute the sample code. If **200** is displayed on the console, the code is successfully executed. For details about related parameters, see **Image Tagging APIs**.

```
200
{
    "result": {
        "tags": [
            {
                 "confidence": "98.01",
                "type": "Tree",
                "tag": "Tree",
                 "i18n_tag": {
                    "zh": "Chinese character for tree",
                     "en": "Tree"
                 },
                 "i18n_type": {
                    "zh": "Chinese character for tree",
                     "en": "Tree"
                 },
                 "instances": [ ]
            }
        ]
    }
 }
```

8 Using the Go SDK

8.1 Preparing a Go Development Environment

Preparations

To use the Go SDK of Image Recognition, you need to configure the Go development environment. If the Go development environment has been installed, skip this section.

1. Download Go 1.14 or a later version from the **Go official website**.
2. Install the downloaded Go in a specified path, for example, **D:\develop\go**.
3. Right-click **Computer**, choose **Properties** > **Advanced System Settings** > **Environment Variables**, and perform the following operations:
    a. Create system variable **GOROOT** whose value is the Go installation path. Add environment variable **GOPATH**, which indicates the directory where the Go project is located.
    b. Append **;%GOROOT%\bin** (path of the Go compiler) to the **PATH** variable.
    c. Append **;%GOROOT%\lib** (path of the Go compiler) to the **CLASSPATH** variable.
4. Download Visual Studio Code from the **Visual Studio Code official website** and install it.
5. Install the Go and Go Critic plug-ins.

Obtaining and Installing an SDK

Run the following command:

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

8.2 Demo Project of Image Tagging

This section describes how to use the Image Tagging SDK by AK/SK authentication.

- Replace the AK/SK in the sample code with the actual AK/SK. You can use either the **url** or **image** parameter in the initialized **ImageTaggingReq** to configure image information. This section uses **url** as an example.
- **endpoint** indicates the regions and endpoints for Huawei Cloud services. For details, see **Regions and Endpoints**.

1. The sample code for calling Image Tagging is as follows:

```
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() {
   // Hard-coded or plaintext AK and SK are risky. For security purposes, encrypt your AK and SK and
store them in the configuration file or environment variables.
   // In this example, the AK and SK are stored in environment variables for identity authentication.
Before running this example, configure environment variables HUAWEICLOUD_SDK_AK and
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("xx-xxx-xx")).  //Replace xx-xxx-xx with the region where you
enable the service. For details, see Regions and Endpoints.
            WithCredential(auth).
            Build())

    request := &model.RunImageTaggingRequest{}
    limitImageTaggingReq:= int32(50)
    thresholdImageTaggingReq:= float32(95)
    languageImageTaggingReq:= "zh"
    urlImageTaggingReq:= "https://XXX.jpg"  //Replace it with an image URL that can be accessed from
the Internet.
    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. If the recognition result is displayed on the console, the execution is successful. For details about related parameters, see **Image Tagging APIs**.

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

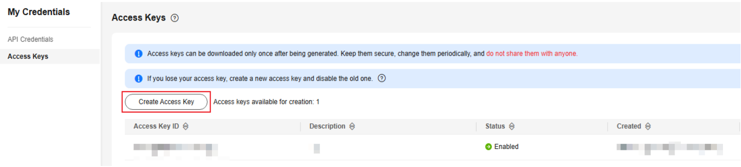
9 Using the PHP SDK

9.1 Preparing a PHP Development Environment

Preparations

- **Sign up for a HUAWEI ID and enable Huawei Cloud services**. Complete real-name authentication and ensure that your account is not in arrears or frozen.
- Obtain PHP 5.6 or later. You can run the **php --version** command to check the version information.
- Obtain an AK and an SK on the **My Credentials** > **Access Keys** page.

Figure 9-1 Obtaining an AK and an SK



Access keys can be downloaded only once after being generated. Keep them secure, change them periodically, and do not share them with anyone.

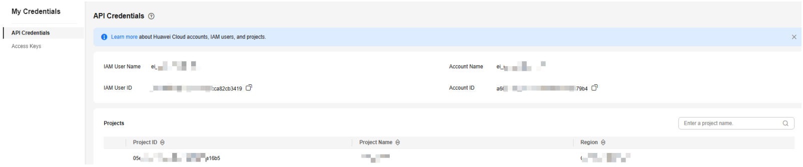
If you lose your access key, create a new access key and disable the old one.

Access keys available for creation: 1

Access Key ID	Description	Status	Created
[redacted]	[redacted]	Enabled	[redacted]

- Obtain the IAM user name, account name, and the project ID of your target region on the **My Credentials** > **API Credentials** page. The information will be used during service calling. Save it in advance.

Figure 9-2 My Credentials



Learn more about Huawei Cloud accounts, IAM users, and projects.

IAM User Name ei_[redacted]

IAM User ID [redacted]ca82cb3419

Account Name ei_[redacted]

Account ID a6[redacted]79b4

Projects

Project ID	Project Name	Region
05[redacted]a16b5	[redacted]	[redacted]

Installing the SDK

You are advised to use **Composer** to install the SDK.

Composer is a dependency manager for PHP. It allows you to declare and install dependencies in a project.

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

After the installation is complete, you need to import the autoload file of Composer.

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

Using the SDK

1. Import dependency modules.
   ```
   <?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. Configure client connection parameters.
   - Using the default configuration
     ```
     // Use the default configuration.
     $config = HttpConfig::getDefaultConfig();
     ```
   - (Optional) Configuring a network proxy
     ```
     // Use a proxy server.
     $config->setProxyProtocol('http');
     $config->setProxyHost('proxy.huawei.com');
     $config->setProxyPort(8080);
     $config->setProxyUser('username');
     $config->setProxyPassword('password');
     ```
   - (Optional) Configuring the timeout
     ```
     // The default connection timeout interval is 60 seconds, and the default read timeout interval is 120 seconds. You can change the default values as needed.
     $config->setTimeout(120);
     $config->setConnectionTimeout(60);
     ```
   - (Optional) Configuring an SSL
     ```
     // Skip server certificate verification.
     $config->setIgnoreSslVerification(true);
     // Configure the server CA certificate so that the SDK can verify the server certificate.
     $config->setCertFile("{yourCertFile}");
     ```
3. Configure authentication information.

   Configure **ak**, **sk**, and **projectId**. AK is used together with SK to sign requests cryptographically, ensuring that the requests are secret, complete, and correct.
   ```
   $ak = getenv('HUAWEICLOUD_SDK_AK');
   $sk = getenv('HUAWEICLOUD_SDK_SK')
   $projectId = getenv("PROJECT_ID");;
   $credentials = new BasicCredentials($ak,$sk,$projectId);
   ```
   The authentication parameters are as follows:
   - **ak** and **sk**: access key and secrete access key, respectively. For how to obtain them, see **Preparations**.
   - **projectId**: Huawei Cloud project ID. For details about how to obtain the ID, see **Preparations**.
   - **endpoint** indicates the regions and endpoints for Huawei Cloud services. For details, see **Regions and Endpoints**.

⚠ CAUTION

- Hard-coded or plaintext AK and SK are risky. For security purposes, encrypt your AK and SK and store them in the configuration file or environment variables.
- In this example, the AK and SK are stored in environment variables for identity authentication. Before running this example, configure environment variables **HUAWEICLOUD_SDK_AK**, **HUAWEICLOUD_SDK_SK**, and **PROJECT_ID**.

4. Initialize the client.

   Specify the cloud service endpoint and initialize the client.

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

5. Send a request and check the response.

```
$request = new RunImageMediaTaggingRequest();
$body = new ImageMediaTaggingReq();
$body->setLimit(10);
$body->setThreshold(60);
$body->setLanguage("en");
$body->setImage ("Image 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->getHttpStatusCode(). "\n";
  echo $e->getRequestId(). "\n";
  echo $e->getErrorCode() . "\n";
  echo $e->getErrorMsg() . "\n";
}
```

6. Handle exceptions.

Table 9-1 Exceptions

Level-1 Category	Level-1 Category Description	Level-2 Category	Level-2 Category Description
ConnectionException	Connection exception	HostUnreachableException	The network is unreachable or access is rejected.
		SslHandShakeException	SSL authentication is abnormal.

Level-1 Category	Level-1 Category Description	Level-2 Category	Level-2 Category Description
RequestTimeoutException	Response timeout exception	CallTimeoutException	The server fails to respond to a single request before timeout.
		RetryOutageException	No valid response is returned after the maximum number of retries specified in the retry policy is reached.
ServiceResponseException	Server response exception	ServerResponseException	Internal server error. HTTP response code: [500,].
		ClientRequestException	Invalid request parameter. HTTP response code: [400, 500).

```
// Capture and process different types of exceptions.
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->getErrorMsg() . "\n";
}
```

NOTE

For details about how to use an asynchronous client and configure logs, see **SDK Center** and **PHP SDK Usage Guide**.

9.2 Demo Project of Image Tagging

This section describes how to use the Image Tagging SDK by AK/SK authentication.

- Replace the AK/SK in the sample code with the actual AK/SK. You can use either the **url** or **image** parameter in the initialized **ImageTaggingReq** to configure image information. This section uses **url** as an example.
- **endpoint** indicates the regions and endpoints for Huawei Cloud services. For details, see **Regions and Endpoints**.

1. The sample code for calling Image Tagging is as follows:

```
<?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;
   // Hard-coded or plaintext AK and SK are risky. For security purposes, encrypt your AK and SK and store them in the configuration file or environment variables.
   // In this example, the AK and SK are stored in environment variables for identity authentication. Before running this example, configure environment variables CLOUD_SDK_AK and CLOUD_SDK_SK.
$ak = getenv('CLOUD_SDK_AK');
$sk = getenv('CLOUD_SDK_SK');
//Replace image.xx-xxx-xx.myhuaweicloud.com with the endpoint of the region where the service is enabled.
$endpoint = "https://image.xx-xxx-xx.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 ("Input image encoded using Base64");
$request->setBody($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->getHttpStatusCode(). "\n";
  echo $e->getRequestId(). "\n";
  echo $e->getErrorCode() . "\n";
  echo $e->getErrorMsg() . "\n";
}
```

2. If the recognition result is displayed on the console, the execution is successful. For details about related parameters, see **Image Tagging APIs**.

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

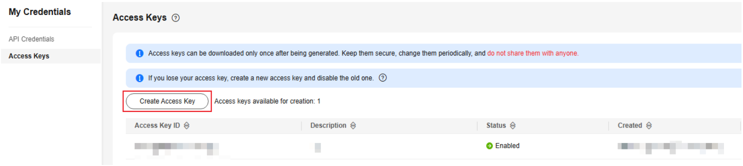
10 Using the C++ SDK

10.1 Preparing a C++ Development Environment

Preparations

- **Sign up for a HUAWEI ID and enable Huawei Cloud services**. Complete real-name authentication and ensure that your account is not in arrears or frozen.
- Obtain C++ 14 or later and CMake 3.10 or later.
- Obtain an AK and an SK on the **My Credentials** > **Access Keys** page.

  Figure 10-1 Obtaining an AK and an SK

  Access keys can be downloaded only once after being generated. Keep them secure, change them periodically, and do not share them with anyone.

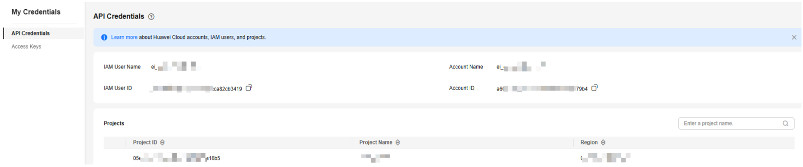
  If you lose your access key, create a new access key and disable the old one.

  Access keys available for creation: 1

Access Key ID	Description	Status	Created
[redacted]	[redacted]	Enabled	[redacted]

- Obtain the IAM user name, account name, and the project ID of your target region on the **My Credentials** > **API Credentials** page. The information will be used during service calling. Save it in advance.

  Figure 10-2 My Credentials

  Learn more about Huawei Cloud accounts, IAM users, and projects.

  IAM User Name ei_[redacted]

  IAM User ID [redacted]ca82cb3419

  Account Name ei_[redacted]

  Account ID a6[redacted]79b4

Project ID	Project Name	Region
05[redacted]a16b5	[redacted]	[redacted]

Installing the SDK

- Installing the SDK on a Linux VM
  a. Obtain dependency packages.

     The required third-party software packages are contained in the package management tools of most Linux OSs, for example, Debian and Ubuntu.

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

spdlog needs to be installed using the source code.

```
git clone https://github.com/gabime/spdlog.git
cd spdlog
mkdir build
cd build
cmake -DCMAKE_POSITION_INDEPENDENT_CODE=ON ..  // Used to generate a dynamic library.
make
sudo make install
```

b. Perform compilation and installation.

```
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
```

After the operations are complete, the installation directory of the C++ SDK is **/usr/local**.

- Installing the SDK on a Windows VM

  a. Install vcpkg and use it to install required software packages.

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

  b. Use CLion for compilation.

     i. Use CLion to open the **huaweicloud-sdk-cpp-v3** directory.

     ii. Choose **File** > **Settings**.

     iii. Choose **Build** > **Execution** > **Deployment** > **CMake**.

     iv. Add the following content to CMake options:

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

     v. Right-click **CMakeLists.txt** and choose **Load CMake Project** from the shortcut menu.

     vi. Choose **Build** to start compilation.

  c. Install the C++ SDK.

  After the compilation is complete, choose **Build** > **Install**.

  After the operations are complete, the installation directory of the C++ SDK is **C:\Program File (x86)\huaweicloud-sdk-cpp-v3**.

Using the SDK

1. Import dependency modules.

```
#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. Configure client connection parameters.

   – Using the default configuration

   ```
   // Use the default configuration.
   HttpConfig httpConfig = HttpConfig();
   ```

- (Optional) Configuring a network proxy

  ```
  // Configure network proxy as needed.
  httpConfig.setProxyProtocol("http");
  httpConfig.setProxyHost("proxy.huawei.com");
  httpConfig.setProxyPort("8080");
  httpConfig.setProxyUser("username");
  httpConfig.setProxyPassword("password");
  ```

- (Optional) Configuring the timeout

  ```
  // The default connection timeout interval is 60 seconds, and the default read timeout interval is 120 seconds. You can change the default values as needed.
  httpConfig.setConnectTimeout(60);
  httpConfig.setReadTimeout(120);
  ```

- (Optional) Configuring an SSL

  ```
  // Skip server certificate verification.
  httpConfig.setIgnoreSslVerification(true);
  ```

3. Configure authentication information.

   Configure **ak**, **sk**, and **projectId**. AK is used together with SK to sign requests cryptographically, ensuring that the requests are secret, complete, and correct.

   ```
   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);
   ```

   The authentication parameters are as follows:

   - **ak** and **sk**: access key and secrete access key, respectively. For how to obtain them, see **Preparations**.
   - **projectId**: Huawei Cloud project ID. For details about how to obtain the ID, see **Preparations**.

   ⚠ CAUTION

   - Hard-coded or plaintext AK and SK are risky. For security purposes, encrypt your AK and SK and store them in the configuration file or environment variables.
   - In this example, the AK and SK are stored in environment variables for identity authentication. Before running this example, configure environment variables **HUAWEICLOUD_SDK_AK**, **HUAWEICLOUD_SDK_SK**, and **PROJECT_ID**.

4. Initialize the client.

   Specify the cloud service endpoint and initialize the client.

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

   endpoint indicates the regions and endpoints for Huawei Cloud services. For details, see **Regions and Endpoints**.

5. Send a request and check the response.

   ```
   // The following is an example of calling the RunImageMediaTagging API:
   RunImageMediaTaggingRequest request;
   ImageMediaTaggingReq body;
   body.setLimit(10);
   ```

```
body.setThreshold(60);
body.setLanguage("zh");
    body.setImage("Input image encoded using Base64");
request.setBody(body);
```

6. Handle exceptions.

Table 10-1 Exceptions

Level-1 Category	Level-1 Category Description	Level-2 Category	Level-2 Category Description
ConnectionException	Connection exception	HostUnreachableException	The network is unreachable or access is rejected.
		SslHandShakeException	SSL authentication is abnormal.
RequestTimeoutException	Response timeout exception	CallTimeoutException	The server fails to respond to a single request before timeout.
		RetryOutageException	No valid response is returned after the maximum number of retries specified in the retry policy is reached.
ServiceResponseException	Server response exception	ServerResponseException	Internal server error. HTTP response code: [500,].
		ClientRequestException	Invalid request parameter. HTTP response code: [400, 500).

```
// Capture and process different types of exceptions.
 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;
```

NOTE

For details about how to use the asynchronous client and configure logs, see **SDK Center** and **C++ SDK Usage Guide**.

10.2 Demo Project of Image Tagging

This section describes how to use the Image Tagging SDK by AK/SK authentication.

- Replace the AK/SK in the sample code with the actual AK/SK. You can use either the **url** or **image** parameter in the initialized **ImageTaggingReq** to configure image information. This section uses **url** as an example.
- **endpoint** indicates the regions and endpoints for Huawei Cloud services. For details, see **Regions and Endpoints**.

1. The sample code for calling Image Tagging is as follows:

```
#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)
{
  // Hard-coded or plaintext AK and SK are risky. For security purposes, encrypt your AK and SK and
store them in the configuration file or environment variables.
  // In this example, the AK and SK are stored in environment variables for identity authentication.
Before running this example, configure environment variables CLOUD_SDK_AK and CLOUD_SDK_SK.
    string ak = getenv("CLOUD_SDK_AK");
    string sk = getenv("CLOUD_SDK_SK");
    string projectId = "";
//Replace image.xx-xxx-xx.myhuaweicloud.com with the endpoint of the region where the service is
enabled.
    string endpoint = "https://image.xx-xxx-xx.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("Input image encoded using Base64");
    request.setBody(body);
    std::cout << "-----begin execute request-------" << std::endl;
    try {
        auto response = client->runImageTagging(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;
    return 0;
}
```

2. If the recognition result is displayed on the console, the execution is successful. For details about related parameters, see **Image Tagging APIs**.

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