Image Recognition

SDK Reference

Issue 01

Date 2022-12-05

HUAWEI

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

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

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

NOTE

Recapture detection does not support SDKs. For details about the APIs for using recapture detection, see **Recapture Detection APIs**.

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.

5 Using Java SDK

5.1 Preparing a Java Development Environment

Image Java SDK uses Java SE Development Kit 8 (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.

Step 1 **Download the JDK file**.

Step 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.

Step 3 Right-click **Computer**, choose **Properties** > **Advanced System Settings** > **Environment Variables**, and perform the following operations to configure Java environment variables:

1. Create system variable **JAVA_HOME** whose value is the JDK installation path.
2. Add **%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin** to **Path**. Separate multiple values with semicolons (;).
3. Create system variable **CLASSPATH** whose value is **%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar**.

Step 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)
```

----End

5.2 Obtaining and Installing the Java 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.

To create a Maven project using IntelliJ IDEA (an example IDE), perform the following steps (skip 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.

You can obtain the latest SDK version from the **SDK Center**.

```
<dependency>
   <groupId>com.huaweicloud.sdk</groupId>
   <artifactId>huaweicloud-sdk-image</artifactId>
   <version>3.1.8</version>  ///Replace it with 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

5.3 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) {
    //Enter your AK/SK.
        String ak = "<YOUR AK>";
        String sk = "<YOUR SK>";
        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);
        ImageClient client = ImageClient.newBuilder()
                .withCredential(auth)
                .withRegion(ImageRegion.valueOf("cn-north-4"))
                .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**.

```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
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
```

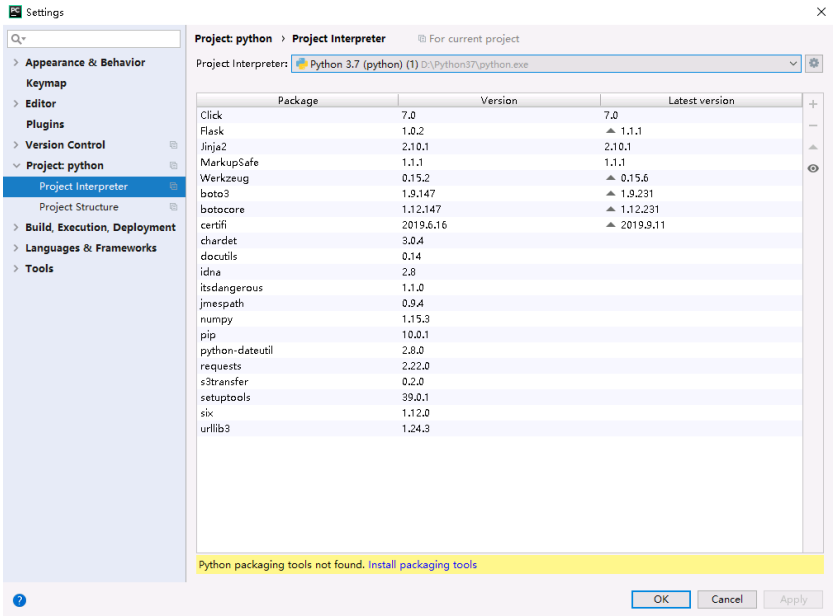

6 Using the Python SDK

6.1 Preparing a Python Development Environment

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



6.2 Obtaining and Installing the Python SDK

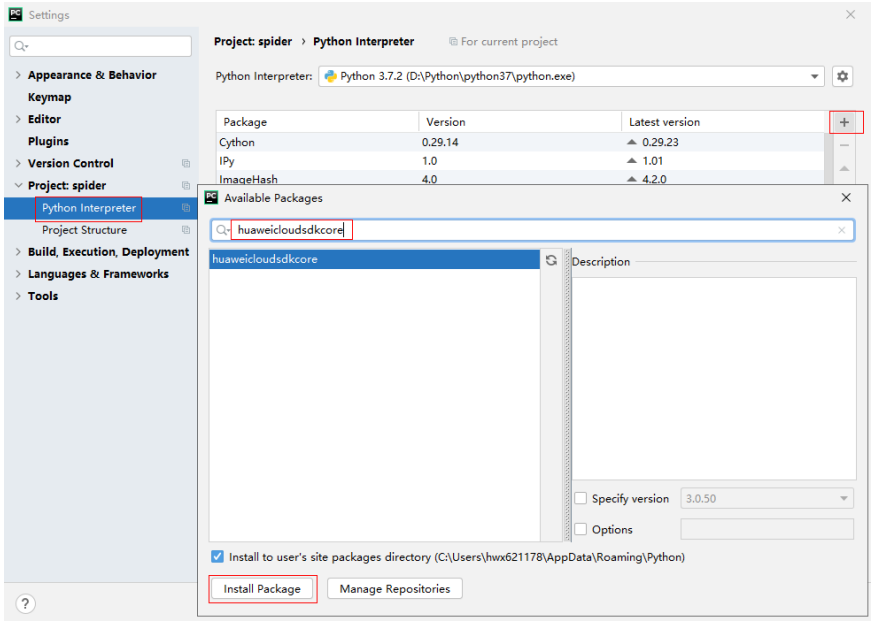
To obtain and install the Python SDK, you are advised to use pip commands or PyCharm. You need to install the **huaweicloudsdkcore** and **huaweicloudsdkimage** packages as follows:

Install the Python SDK using pip commands:
Install the **huaweicloudsdkcore** package.
pip install huaweicloudsdkcore

Install the **huaweicloudsdkimage** package.
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



6.3 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__":
    //Enter your AK/SK.
    ak = "<YOUR AK>"
    sk = "<YOUR SK>"

    credentials = BasicCredentials(ak, sk)

    client = ImageClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ImageRegion.value_of("cn-north-1")) \
        .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

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.

7.2 Obtaining and Installing the .Net 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**.

7.3 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) {
       //Enter your AK/SK.
            const string ak = "<YOUR AK>";
            const string sk = "<YOUR SK>";

            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"))
                    .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

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.

8.2 Obtaining and Installing the Go SDK

Run the following command:

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

8.3 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() {
 //Enter your AK/SK.
    ak := "<YOUR AK>"
    sk := "<YOUR SK>"

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

    client := image.NewImageClient(
        image.ImageClientBuilder().
            WithRegion(region.ValueOf("cn-north-1")).
            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
```

