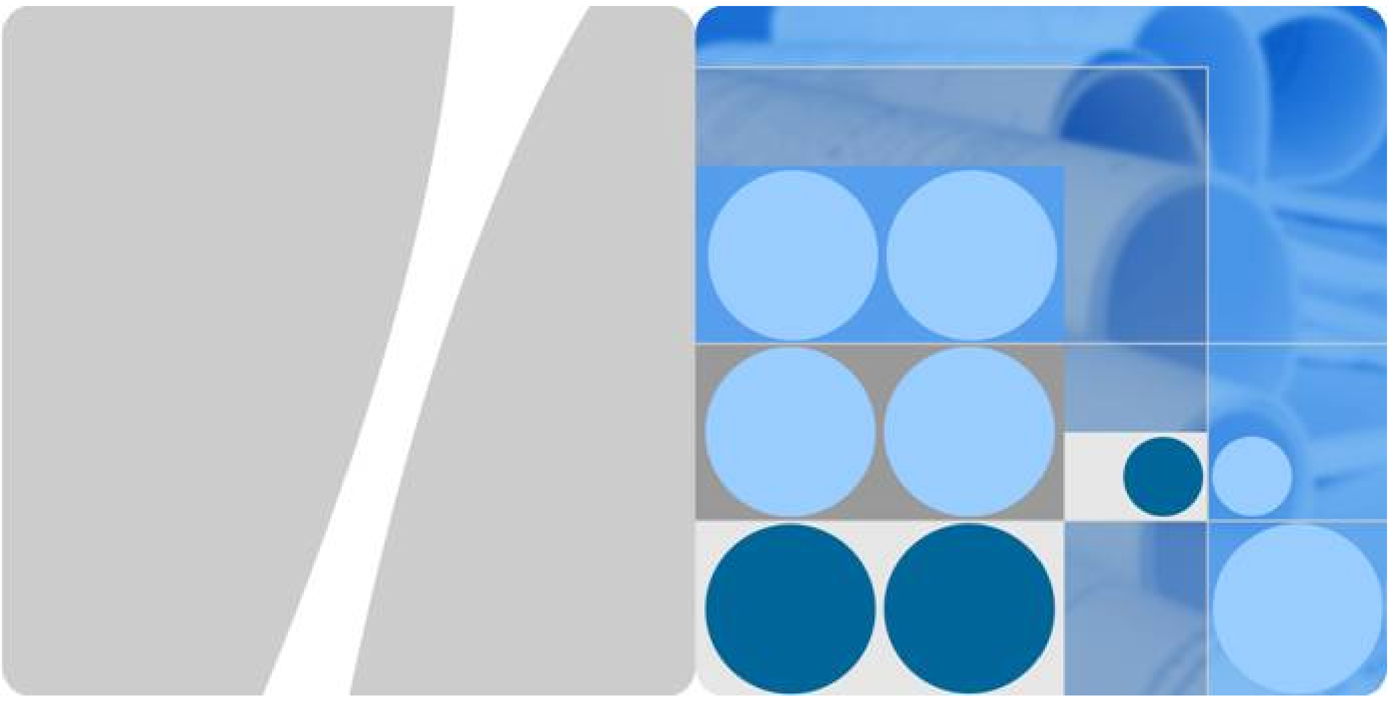

Image Recognition

API Reference

Date **2022-08-01**

Contents

1 Before You Start

1.1 Overview

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 result by accessing and calling APIs in real time. It helps you collect key data automatically and build an intelligent business system, thereby improving service efficiency.

You can perform related operations based on the API description, syntax, parameter description, and examples provided in this document. For details about all supported operations, see **API Overview**.

If you plan to access Image Recognition through an API, ensure that you are familiar with Image Recognition concepts. For details, see the *Image Recognition Service Overview*.

1.2 API Calling

Image Recognition provides Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details about API calling, see **Calling APIs**.

1.3 Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions.

1.4 Limitations and Constraints

For details, see the API description and "Constraints" in the *Image Recognition Service Overview*.

1.5 Concepts

- User

  A user is created in IAM using an account to use cloud services. Each user has its own identity credentials (password and access keys).

- Region

  A region is a physical location where cloud resources are deployed. Availability zones (AZs) in the same region can communicate with each other over an intranet but AZs in different regions cannot communicate with each other. Deploying cloud resources in different regions can better suit certain user requirements or comply with local laws or regulations.

- Availability Zone (AZ)

  AZs are physically isolated locations in a region, but are interconnected through an internal network for enhanced application availability.

2 API Overview

By using the APIs of Image Recognition, you can perform the operations described in **Table 2-1**.

Table 2-1 API description

API	Description
Image Tagging (v1.0)	This API can recognize hundreds of scenes and thousands of objects and their properties in natural images, making intelligent album management, picture retrieval and classification, and scene- or object-based advertising more intuitive.

3 Calling APIs

3.1 Applying for a Service

Before using Image Recognition, you must apply for your desired services of Image Recognition. The following is the procedure for applying for services of Image Recognition.

NOTE

- If you use Image Recognition for the first time, apply for your desired services first. You only need to apply for a service once.

3.2 Making an API Request

This section describes the structure of a REST API, and uses the IAM API for obtaining a user token as an example to describe how to call an API. The obtained token is used to authenticate the calling of other APIs.

Request URI

A request URI is in the following format:

{URI-scheme} :// {Endpoint} / {resource-path} ? {query-string}

Although a request URI is included in the request header, most programming languages or frameworks require the request URI to be transmitted separately.

Table 3-1 URI parameter description

Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from the administrator.

Parameter	Description
resource-path	Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the **resource-path** of the API used to obtain a user token is **/v3/auth/tokens**.
Query string	Query parameter, which is optional. Ensure that a question mark (?) is included before a query parameter that is in the format of "Parameter name=Parameter value". For example, **?limit=10** indicates that a maximum of 10 pieces of data is to be viewed.

NOTE

To simplify the URI display in this document, each API is provided only with a **resource-path** and a request method. The **URI-scheme** of all APIs is **HTTPS**, and the endpoints of all APIs in the same region are identical.

Request Methods

The HTTP protocol defines the following request methods that can be used to send a request to the server:

Table 3-2 HTTP-defined request methods

Method	Description
GET	Requests the server to return specified resources.
PUT	Requests the server to update specified resources.
POST	Requests the server to add resources or perform special operations.
DELETE	Requests the server to delete specified resources, for example, an object.
HEAD	Same as GET except that the server must return only the response header.
PATCH	Requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

Common request header fields are as follows:

Table 3-3 Common request header fields

Parameter	Description	Mandatory	Example
X-Sdk-Date	Specifies the time when the request is sent. The time is in **YYYYMMDD'T'HHMMSS'Z'** format. The value is the current GMT time of the system.	No This field is mandatory for AK/SK-based authentication.	20150907T101459Z
Authorization	Specifies signature authentication information. The value can be obtained from the request signing result.	No This field is mandatory for AK/SK-based authentication.	SDK-HMAC-SHA256 Credential=ZIRRKMTWPTQFQI1WKNKB/20150907//ec2/sdk_request, SignedHeaders=content-type;host;x-sdk-date, Signature=55741b610f3c9fa3ae40b5a8021ebf7ebc2a28a603fc62d25cb3bfe6608e1994
Host	Specifies the server domain name and port number of the resource being requested. The value can be obtained from the URL of a service API. The value is *hostname[:port]*. If the port number is not specified, the default port is used. The default port number for **https** is **443**.	No This field is mandatory for AK/SK-based authentication.	code.test.com or code.test.com:443
Content-Type	Specifies the MIME type of the request body.	Yes	application/json
Content-Length	Specifies the length of the request body. The unit is byte.	This field is mandatory for POST and PUT requests, but must be left blank for GET requests.	3495

Parameter	Description	Mandatory	Example
X-Project-Id	Specifies the project ID. This field is used to obtain the token for each project. This field is mandatory for the request from a DeC or multi-project user.	No	e9993fc787d94b6c886cbaa340f9c0f4
X-Auth-Token	Specifies the user token.	No This field is mandatory for token-based authentication.	-

NOTE

In addition to supporting token-based authentication, APIs also support authentication using access key ID/secret access key (AK/SK). During AK/SK-based authentication, an SDK is used to sign the request, and the **Authorization** (signature information) and **X-Sdk-Date** (time when the request is sent) header fields are automatically added to the request.

For details about other fields in the header, see the HTTPS protocol documentation.

Request Body

The body of a request is often sent in a structured format (JSON or XML) as specified in the **Content-Type** header field. The request body transfers content except the request header.

The request body varies between APIs. Some APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

In the case of the API used to obtain a user token, the request parameters and parameter description can be obtained from the API request. The following provides a sample request with the body included. Set the username (**username**), account name (**domainname**), login password (************), and project name (**xxxxx**).

NOTE

The **scope** parameter specifies where a token takes effect. You can set **scope** to an account or a project under an account. In the following example, the token takes effect only for the resources in a specified project. For more information about this API, see Obtaining a User Token.

```
POST https://{{IAM endpoint}}/v3/auth/tokens
Content-Type: application/json

{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",
                    "password": "********",
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "xxxxx"
            }
        }
    }
}
```

If all data required for the API request is available, you can send the request to call the API through curl, Postman, or coding. In the response to the API used to obtain a user token, **x-subject-token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

3.3 Authentication

Requests for calling an API can be authenticated using either of the following methods:

- Token-based authentication: Requests are authenticated using a token.
- AK/SK-based authentication: Requests are authenticated by encrypting the request body using an AK/SK pair.

Token-based Authentication

NOTE

The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API.

When calling the API to obtain a user token, you must set **auth.scope** in the request body to **project**.

```
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",
                    "password": "********",  //The password is that of the current account. If you log in using a 
subaccount, set this parameter to the password of the subaccount.
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "xxxxxxxx"
            }
        }
    }
}
```

After a token is obtained, the **X-Auth-Token** header field must be added to requests to specify the token when calling other APIs. For example, if the token is **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request as follows:

```
POST https://{{IAM endpoint}}/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

AK/SK-based Authentication

NOTE

AK/SK-based authentication and token-based authentication apply only to requests whose body size is less than 12 MB.

In AK/SK-based authentication, AK/SK is used to sign requests and the signature is then added to the requests for authentication.

- AK: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.
- SK: secret access key used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

In AK/SK-based authentication, you can use an AK/SK to sign a request based on the signature algorithm.

3.4 Response

Status Code

After sending a request, you will receive a response, including a status code, response header, and response body.

A status code is a group of digits, ranging from 1*xx* to 5*xx*. It indicates the status of a request. For more information, see **Status Codes**.

For example, if status code **201** is returned for calling the API used to obtain a user token, the request is successful.

Response Header

Similar to a request, a response also has a header, for example, **Content-Type**.

Table 3-4 Response header

Parameter	Description
Content-Length	Indicates the length of the response body. The unit is byte.
Date	Indicates the time when a request response is returned.
Content-Type	Indicates the MIME type of the response body.

In the response header fields for the API used to obtain a user token, the **x-subject-token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

Response Body

The body of a response is often returned in structured format as specified in the **Content-Type** header field. The response body transfers content except the response header.

The following is part of the response body for the API used to obtain a user token.

```
{
    "token": {
        "expires_at": "2019-02-13T06:52:13.855000Z",
        "methods": [
            "password"
        ],
        "catalog": [
            {
                "endpoints": [
                    {
                        "region_id": "xxx",
......
```

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

```
{
    "error_msg": "The format of message is error",
    "error_code": "AS.0001"
}
```

In the response body, **error_code** is an error code, and **error_msg** provides information about the error.

4 Image Recognition APIs

4.1 Image Tagging (v1.0)

Function

Image Tagging can accurately identify hundreds of scenes and thousands of common objects and their attributes in natural images. It makes intelligent album management, photo retrieval and classification, and scene- or object-based advertising more intuitive. After you upload the image to be processed, Image Tagging returns tags and confidence scores to you.

Prerequisites

- Before using Image Tagging, you need to apply for the service and complete authentication. For details, see **Applying for a Service** and **Authentication**.

URI

URI format

POST /v1.0/image/tagging

Request Message

Table 4-1 describes the request parameters.

Table 4-1 Parameter description

Parameter	Mandatory	Type	Description
image	false	String	Configure either this parameter or **url**. Indicates the Base64 character string converted from the image. The size cannot exceed 10 MB. The image resolution of the narrow sides must be greater than 15 pixels, and that of the wide sides cannot exceed 4096 pixels. The supported image formats include JPG, PNG, and BMP.
url	false	String	Configure either this parameter or **image**. Indicates the URL of an image. The options are as follows: ● HTTP/HTTPS URLs on the public network ● OBS URLs. To use OBS data, authorization is required, including service authorization, temporary authorization, and anonymous public authorization. For details, see Configuring the Access Permission for OBSConfiguring the Access Permission for OBS. **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the images to be detected reside is stable and reliable. OBS is recommended for storing image data. ● The region of OBS must be consistent with that of Image Recognition.
language	false	String	**zh**: indicates that the language of the returned tag is Chinese. **en**: indicates that the language of the returned tag is English. The default value is **zh**.
limit	false	Integer	Indicates the maximum number of tags that can be returned. The default value is **30**.
threshold	false	Float	Indicates the threshold of the confidence score. The value ranges from 0 to 100. If you input a value beyond the value range, the default value is used. The default value is **0**.

Response Message

Table 4-2 describes the response parameters.

Table 4-2 Parameter description

Parameter	Type	Description
result	JSON	Indicates the content of the image tag when the API is successfully called. This parameter is not included when the API fails to be called.
tags	List	Indicates the list of tags.
confidence	Float	Indicates the confidence score. The value ranges from 0 to 100.
tag	String	Indicates the tag name.
type	String	Indicates the tag type. Possible values are as follows: ● **object**: entity tag ● **scene**: scenario tag ● **concept**: concept tag
i18n_tag	JSON	Indicates the internationalization field of the tag. (**i18n** is only an internationalization flag and has no special meaning.) ● **zh**: Chinese ● **en**: English
error_code	String	Indicates the error code of a failed API call. For details, see **Error Codes**. This parameter is not included when the API is successfully called.
error_msg	String	Indicates the error message of a failed API call. This parameter is not included when the API is successfully called.

Examples

- Example request (Method 1: Use a Base64-encoded image.)

```
POST https://{{Endpoint of Image Recognition}}/v1.0/image/tagging

Request Header:
Content-Type:application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:
{
 "image":"/9j/4AAQSkZJRgABAgEASABIAAD/
4RFZRXhpZgAATU0AKgAAAAgABwESAAMAAAABAAEAAAEaAAUAAAABAAAAYgEbAAUAAAABAAAAag
EoAAMAAAABAAIAAAExAAIAAAAcAAAAcgEyAAIAAAAUAAAAjodpAAQAAAABAAAApAAAANAACvyAAA
AnEAAK/
IAAACcQQWRvYmUgUGhvdG9zaG9wIENTMyBXaW5kb3dzADIwMTc6MTA6MjAgMTA6NDU6MzYAAAA
AA6ABAAMAAAAB//
8AAKACAAQAAAABAAALIKADAAQAAAABAAAGQAAAAAAAAAAGAQMAAwAAAAEABgAAARoABQAAA
AEAAAEeARsABQAAAAEAAAEmASgAAwAAAAEAAgAAAgEABAAAAAEAAAEuAgIABAAAAAEAABAj...",
  "url": "",
  "language": "en",
  "limit": 5,
  "threshold": 60.0
}
```

- Example request (Method 2: Use the URL redirecting to an image file.)

```
POST https://{{Endpoint of Image Recognition}}/v1.0/image/tagging

Request Header:
Content-Type:application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:
{
  "image":"",
  "url":"https://{{OBS path for storing images}}",
  "language": "en",
  "limit": 5,
  "threshold": 60.0
}
```

- Example successful response

```
{
  "result":{
      "tags":[
        {
            "confidence": 38.51,
            "tag":"sky",
            "i18n_tag":
            {
                 "en": "sky",
                 "zh": "Tag added to the image"
            },
            "type": "object"
        },
        {
            "confidence": 25.75,
            "tag":"landscape",
            "i18n_tag":
            {
                 "en": "landscape",
                 "zh": "Tag added to the image"
             },
            "type": "scene"
        }
        ]
    }
}
```

- Example failed response

```
{
    "error_code": "AIS.0014",
    "error_msg": "The JSON format of the input data is incorrect."
}
```

Return Value

- Normal

  200

- Abnormal

Return Value	Description
400	• The request cannot be understood by the server due to malformed syntax. The client should not repeat the request without modifications. • The request parameter is incorrect.
401	The request requires user authentication.
403	No operation permission.
404	The server has not found anything matching the Request-URI.
500	The server encountered an unexpected condition which prevented it from fulfilling the request.

Error Codes

For details about error codes, see **Error Codes**.

4.2 Image Tagging (v2.0)

Function

Image Tagging can accurately identify hundreds of scenes, thousands of common objects, and dozens of tag types in natural images. It makes intelligent album management, photo retrieval and classification, and scene- or object-based advertising more intuitive. After you upload an image, Image Tagging returns the tags, tag types, and confidence scores of the image to you. Image Tagging also returns the location information of common objects in the image.

NOTE

Compared with v1.0, v2.0 refines tag types and provides more return information. For example, v2.0 can return location information of common objects. In addition, v2.0 uses an upgraded model that provides higher recognition accuracy. Therefore, v2.0 is recommended.

Prerequisites

- Before using Image Tagging, you need to apply for the service and complete authentication. For details, see **Applying for a Service** and **Authentication**.

URI

URI format

POST /v2/{project_id}/image/tagging

Table 4-3 Description

Parameter	Mandatory	Description
project_id	Yes	Indicates the project ID.

Request Message

Table 4-4 describes the request parameters.

Table 4-4 Parameter description

Parameter	Mandatory	Type	Description
image	false	String	Configure either this parameter or **url**. Indicates the Base64 character string converted from the image. The size cannot exceed 10 MB. The image resolution of the narrow sides must be greater than 15 pixels, and that of the wide sides cannot exceed 4096 pixels. The supported image formats include JPG, PNG, and BMP.
url	false	String	Configure either this parameter or **image**. Indicates the URL of an image. The options are as follows: ● HTTP/HTTPS URLs on the public network ● OBS URLs. To use OBS data, authorization is required, including service authorization, temporary authorization, and anonymous public authorization. **NOTE** ● The API response time depends on the image download time. If the image download takes a long time, the API call will fail. ● Ensure that the storage service where the images to be detected reside is stable and reliable. OBS is recommended for storing image data. ● The region of OBS must be consistent with that of Image Recognition.
language	false	String	**zh**: indicates that the language of the returned tag is Chinese. **en**: indicates that the language of the returned tag is English. The default value is **zh**.
limit	false	Integer	Indicates the maximum number of tags that can be returned. The default value is **50**.

Parameter	Mandatory	Type	Description
threshold	false	Float	Indicates the confidence threshold. The value ranges from 0 to 100. The default value is **60**.

Response Message

Table 4-5 describes the response parameters.

Table 4-5 Parameter description

Parameter	Type	Description
result	JSON	Indicates the content of the image tag when the API is successfully called. This parameter is not included when the API fails to be called.
tags	List	Indicates the list of tags.
confidence	String	Indicates the tag confidence. The float value is converted into a string and returned. The float value ranges from 0 to 100.
tag	String	Indicates the tag name in the selected language (currently, only Chinese and English are supported).
type	String	Indicates the tag type in the selected language (currently, only Chinese and English are supported).
i18n_tag	JSON	Indicates the internationalization field of the tag. (**i18n** is only an internationalization flag and has no special meaning.) ● **zh**: Chinese ● **en**: English
i18n_type	JSON	Indicates the internationalization field of the tag type. (**i18n** is only an internationalization flag and has no special meaning.) ● **zh**: Chinese ● **en**: English

Parameter	Type	Description
instances	List	If this parameter is empty, there is no object bounding box. If this parameter is not empty, it has the following fields: ● **bounding_box**: location of the object bounding box – **width**: width of the bounding box, which is in float format – **height**: height of the bounding box, which is in float format – **top_left_y**: distance from the upper left corner of the bounding box to the vertical axis, which is in float format – **top_left_x**: distance from the upper left corner of the bounding box to the horizontal axis, which is in float format ● **confidence**: tag confidence. The float value is converted into a string and returned. The float value ranges from 0 to 100.
error_code	String	Indicates the error code of a failed API call. For details, see **Error Codes**. This parameter is not included when the API is successfully called.
error_msg	String	Indicates the error message of a failed API call. This parameter is not included when the API is successfully called.

Examples

- Example request (Method 1: Use a Base64-encoded image.)

```
POST https://{{Endpoint of Image Recognition}}/v2/{project_id}/image/tagging

Request Header:
Content-Type:application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:
{
 "image":"/9j/4AAQSkZJRgABAgEASABIAAD/
4RFZRXhpZgAATU0AKgAAAAgABwESAAMAAAABAAEAAAEaAAUAAAABAAAAYgEbAAUAAAABAAAAag
EoAAMAAAABAAIAAAExAAIAAAAcAAAAcgEyAAIAAAAUAAAAjodpAAQAAAABAAAApAAAANAACvyAAA
AnEAAK/
IAAACcQQWRvYmUgUGhvdG9zaG9wIENTMyBXaW5kb3dzADIwMTc6MTA6MjAgMTA6NDU6MzYAAAA
AA6ABAAMAAAAB//
8AAKACAAQAAAABAAALIKADAAQAAAABAAAGQAAAAAAAAAGAQMAAwAAAAEABgAAARoABQAAA
AEAAAEeARsABQAAAAEAAAEmASgAAwAAAAEAAgAAAgEABAAAAAEAAAEuAgIABAAAAAEAABAj...",
  "threshold": 60
}
```

- Example request (Method 2: Use the URL redirecting to an image file.)

```
POST https://{{Endpoint of Image Recognition}}/v2/{project_id}/image/tagging

Request Header:
Content-Type:application/json
X-Auth-Token:
MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:
{
  "image":"",
  "url":"https://{{OBS path for storing images}}",
  "threshold": 60.0
}
```

- Example successful response

```
{
    "result":
    {
        "tags":
        [
            {
                "confidence": "92.38",
                "instances":
                [
                    {
                        "bounding_box":
                        {
                            "height": 133.32496056189905,
                            "top_left_x": 53.134917332575874,
                            "top_left_y": 254.21347984900842,
                            "width": 117.5866567171537
                        },
                        "confidence": "92.38"
                    },
                    {
                        "bounding_box":
                        {
                            "height": 133.32496056189905,
                            "top_left_x": 53.134917332575874,
                            "top_left_y": 254.21347984900842,
                            "width": 117.5866567171537
                        },
                        "confidence": "90.38"
                    }
                ],
                "tag": "Person",
                "i18n_tag":
                {
                    "en": "Person",
                    "zh": "Chinese character for person"
                },
                "type": "Human",
                "i18n_type":
                {
                    "en": "Human",
                    "zh": "Chinese characters for human"
                }
            },
            {
                "confidence": "94.38",
                "instances":
                [],
                "tag": "Book",
                "i18n_tag":
                {
                    "en": "Book",
                    "zh": "Chinese characters for book"
                },
                "type": "Education",
                "i18n_type":
                {
                    "en": "Education",
                    "zh": "Chinese characters for education"
                }
            },
            {
                "confidence": "92.38",
                "instances":
                [
                    {
                        "bounding_box":
                        {
                            "height": 133.32496056189905,
                            "top_left_x": 53.134917332575874,
                            "top_left_y": 254.21347984900842,
                            "width": 117.5866567171537
                        },
                        "confidence": "92.38"
                    }
                ],
                "tag": "Bed",
                "i18n_tag":
                {
                    "en": "Bed",
                    "zh": "Chinese character for bed"
                },
                "type": "Home category",
                "i18n_type":
                {
                    "en": "Home category",
                    "zh": "Chinese characters for home category"
                }
            }
        ]
    }
}
```

- Example failed response

```
{
    "error_code": "AIS.0014",
    "error_msg": "The JSON format of the input data is incorrect."
}
```

Return Value

- Normal

  200

- Abnormal

Return Value	**Description**
400	● The request cannot be understood by the server due to malformed syntax. The client should not repeat the request without modifications. ● The request parameter is incorrect.
401	The request requires user authentication.
403	No operation permission.
404	The server has not found anything matching the Request-URI.

Return Value	Description
500	The server encountered an unexpected condition which prevented it from fulfilling the request.

Error Codes

For details about error codes, see **Error Codes**.

5 Application Examples

5.1 Python 3 API Example

This section uses Image Tagging as an example to describe how to call Python 3 APIs.

```
# encoding:utf-8

import requests
import base64

url = "https://{endpoint}/v1.0/image/tagging"
token = "Actual token value obtained by the user"
headers = {'Content-Type': 'application/json', 'X-Auth-Token': token}

imagepath = r'data/image-tagging.jpg'
with open(imagepath, "rb") as bin_data:
    image_data = bin_data.read()
image_base64 = base64.b64encode(image_data).decode("utf-8")  # Use Base64 encoding of images.
data= {"image": image_base64]}  # Set either the URL or the image.

response = requests.post(url, headers=headers, json=data, verify=False)
print(response.text)
```

Table 5-1 Parameter description

Parameter	Description
url	API request URL, for example, https://{endpoint}/v1.0/image/tagging.
token	A token is a user's access credential, which includes user identities and permissions. When you call an API to access a cloud service, a token is required for identity authentication. For details about how to obtain the token, see .
imagePath	Image path. An image file path or image URL is supported. The URL can be an HTTP/HTTPS or OBS URL.

5.2 Java API Example

This section uses Image Tagging as an example to describe how to call Java APIs.

```
package com.huawei.ais.demo;
import com.huawei.ais.sdk.util.HttpClientUtils;

import java.io.File;
import java.io.IOException;
import java.net.URISyntaxException;

import org.apache.http.Header;
import org.apache.http.HttpResponse;
import org.apache.http.entity.StringEntity;
import org.apache.commons.codec.binary.Base64;
import org.apache.commons.io.FileUtils;
import org.apache.commons.io.IOUtils;
import com.alibaba.fastjson.JSONObject;

import org.apache.http.entity.ContentType;
import org.apache.http.message.BasicHeader;

/**
 * This demo is used only for tests. You are advised to use the SDK.
 * Before using this demo, configure the dependent JAR package. Obtain this package by downloading the
SDK.
 */

public class ImageTaggingDemo {
    public static void main(String[] args) throws URISyntaxException, UnsupportedOperationException,
IOException{
        TokenDemo();
    }

    public static void TokenDemo() throws URISyntaxException, UnsupportedOperationException,
IOException {

        String url = "https://{endpoint}/v1.0/image/tagging";
        String token = "Actual token value obtained by the user";
        String imgPath = "data/image-tagging.jpg"; //File path or URL of the image to be recognized.

        JSONObject params = new JSONObject();
        try {
            if (imgPath.indexOf("http://") != -1 || imgPath.indexOf("https://") != -1) {
                params.put("url", imgPath);
            } else {
                byte[] fileData = FileUtils.readFileToByteArray(new File(imgPath));
                String fileBase64Str = Base64.encodeBase64String(fileData);
                params.put("image", fileBase64Str);
            }

            Header[] headers = new Header[]{new BasicHeader("X-Auth-Token", token), new
BasicHeader("Content-Type", ContentType.APPLICATION_JSON.toString())};
            StringEntity stringEntity = new StringEntity(params.toJSONString(), "utf-8");
            HttpResponse response = HttpClientUtils.post(url, headers, stringEntity);
            String content = IOUtils.toString(response.getEntity().getContent(), "utf-8");
            System.out.println(content);
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Table 5-2 Parameter description

Parameter	Description
url	API request URL, for example, https://{endpoint}/v1.0/image/tagging.
token	A token is a user's access credential, which includes user identities and permissions. When you call an API to access a cloud service, a token is required for identity authentication. For details about how to obtain the token, see .
imgPath	Image path. An image file path or image URL is supported. The URL can be an HTTP/HTTPS or OBS URL.

5.3 PHP API Example

This section uses Image Tagging as an example to describe how to call PHP APIs.

```
<?php

function TokenRequest() {
    $url = "https://{endpoint}/v1.0/image/tagging";
     $token = "Actual token value obtained by the user";
    $imagePath = __DIR__.'data/image-tagging.jpg';

    $data = array();
    if (stripos($imagePath, 'http://') !== false || stripos($imagePath, 'https://') !== false) {
        $data['url'] = $imagePath;
    } else {
        if($fp = fopen($imagePath,"rb", 0))
        {
            $gambar = fread($fp,filesize($imagePath));
            fclose($fp);

            $fileBase64 = chunk_split(base64_encode($gambar));
        } else {
            echo "Failed to read the image.";
            return;
        }
        $data['image'] = $fileBase64;
    }

    $curl = curl_init();
    $headers = array(
        "Content-Type:application/json",
        "X-Auth-Token:" . $token
    );

    /* Setting the request body */
    curl_setopt($curl, CURLOPT_URL, $url);
    curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, json_encode($data));
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, TRUE);
    curl_setopt($curl, CURLOPT_NOBODY, FALSE);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($curl, CURLOPT_TIMEOUT, 30);

    $response = curl_exec($curl);
    $status = curl_getinfo($curl, CURLINFO_HTTP_CODE);
    curl_close($curl);
    echo $response;
}

TokenRequest();
```

Table 5-3 Parameter description

Parameter	Description
url	API request URL, for example, https://{endpoint}/v1.0/image/tagging.
token	A token is a user's access credential, which includes user identities and permissions. When you call an API to access a cloud service, a token is required for identity authentication. For details about how to obtain the token, see .
imagePath	Image path. An image file path or image URL is supported. The URL can be an HTTP/HTTPS or OBS URL.

6 Appendix

6.1 Status Codes

Table 6-1 describes status codes.

Table 6-1 Status codes

Status Code	Message	Description
100	Continue	The client continues sending the request. The server has received the initial part of the request and the client should continue sending the remaining part.
101	Switching Protocols	The requester has asked the server to switch protocols and the server has agreed to do so. The target protocol must be more advanced than the source protocol. For example, the current HTTPS protocol is switched to a later version.
200	OK	The server has successfully processed the request.
201	Created	The request for creating a resource has been fulfilled.
202	Accepted	The request has been accepted for processing, but the processing has not been completed.
203	Non-Authoritative Information	The server successfully processed the request, but is returning information that may be from another source.

Status Code	Message	Description
204	No Content	The server has successfully processed the request, but has not returned any content. The status code is returned in response to an HTTP OPTIONS request.
205	Reset Content	The server has fulfilled the request, but the requester is required to reset the content.
206	Partial Content	The server has successfully processed a part of the GET request.
300	Multiple Choices	There are multiple options for the location of the requested resource. The response contains a list of resource characteristics and addresses from which the user or user agent (such as a browser) can choose the most appropriate one.
301	Moved Permanently	The requested resource has been assigned a new permanent URI, and the new URI is contained in the response.
302	Found	The requested resource resides temporarily under a different URI.
303	See Other	Retrieve a location. The response to the request can be found under a different URI and should be retrieved using a GET or POST method.
304	Not Modified	The requested resource has not been modified. When the server returns this status code, it does not return any resources.
305	Use Proxy	The requested resource must be accessed through a proxy.
306	Unused	The HTTP status code is no longer used.
400	Bad Request	The request is invalid. The client should not repeat the request without modifications.
401	Unauthorized	The status code is returned after the client provides the authentication information, indicating that the authentication information is incorrect or invalid.
402	Payment Required	This status code is reserved for future use.

Status Code	Message	Description
403	Forbidden	The server understood the request, but is refusing to fulfill it. The client should not repeat the request without modifications.
404	Not Found	The requested resource cannot be found. The client should not repeat the request without modifications.
405	Method Not Allowed	The method specified in the request is not supported for the requested resource. The client should not repeat the request without modifications.
406	Not Acceptable	The server cannot fulfill the request according to the content characteristics of the request.
407	Proxy Authentication Required	This status code is similar to 401, but indicates that the client must first authenticate itself with the proxy.
408	Request Timeout	The server times out when waiting for the request. The client may repeat the request without modifications at any later time.
409	Conflict	The request could not be completed due to a conflict with the current state of the resource. This status code indicates that the resource that the client attempts to create already exits, or the request fails to be processed because of the update of the conflict request.
410	Gone	The requested resource is no longer available. The status code indicates that the requested resource has been deleted permanently.
411	Length Required	The server refuses to process the request without a defined Content-Length.
412	Precondition Failed	The server does not meet one of the preconditions that the requester puts on the request.

Status Code	Message	Description
413	Request Entity Too Large	The request is larger than that a server is able to process. The server may close the connection to prevent the client from continuing the request. If the server cannot process the request temporarily, the response will contain a Retry-After header field.
414	Request URI Too Long	The URI provided was too long for the server to process.
415	Unsupported Media Type	The server is unable to process the media format in the request.
416	Requested Range Not Satisfiable	The requested range is invalid.
417	Expectation Failed	The server fails to meet the requirements of the Expect request-header field.
422	Unprocessable Entity	The request is well-formed but is unable to be processed due to semantic errors.
429	Too Many Requests	The client sends excessive requests to the server within a given time (exceeding the limit on the access frequency of the client), or the server receives excessive requests within a given time (beyond its processing capability). In this case, the client should repeat requests after the time specified in the Retry-After header of the response expires.
500	Internal Server Error	The server is able to receive but unable to understand the request.
501	Not Implemented	The server does not support the function required to fulfill the request.
502	Bad Gateway	The server was acting as a gateway or proxy and received an invalid response from the upstream server.
503	Service Unavailable	The requested service is invalid. The client should not repeat the request without modifications.
504	Gateway Timeout	The request cannot be fulfilled within a given time. This status code is returned to the client only when the **Timeout** parameter is specified in the request.

Status Code	Message	Description
505	HTTP Version Not Supported	The server does not support the HTTPS protocol version used in the request.

6.2 Error Codes

Function

A customized message is returned when errors occur in an extended API. This section describes error codes and their meanings.

Format of an Error Response Body

If an error occurs during API calling, the system returns an error code and message to you. The following shows the format of an error response body:

```
STATUS CODE 400
{
    "error_code": "AIS.0014",
    "error_msg": "The JSON format of the input data is incorrect."
}
```

Error Code Description

If an error occurs during API calling, no result is returned. You can locate the cause of an error using the error code of each API.

The returned message body contains a specific error code and error message.

Status Code	Error Code	Description	Description	Handling Measure
400	AIS.0012	The request parameter is not supported.	The request parameter is not supported.	Check whether the fields in the request are valid. For details, see the request description of the corresponding API.
400	AIS.0014	The JSON format of the input data is incorrect.	The JSON format of the input data is incorrect.	Check the JSON format of the input data.
400	AIS.0501	The input parameter is invalid.	The input parameter is invalid.	Check the parameter.

Status Code	Error Code	Description	Description	Handling Measure
400	AIS.0502	The image format is not supported.	The image format is not supported.	Check whether the image format is supported.
400	AIS.0503	The image is damaged.	The image is damaged.	Check whether the image is damaged. If yes, re-upload an image file that meets the requirements.
400	AIS.0504	The image size does not meet requirements.	The image size does not meet requirements.	Check whether the image size meets requirements.
400	AIS.0505	Failed to run the algorithm.	Failed to run the algorithm.	Check the configuration. Contact technical support.
400	AIS.0506	An internal error occurred.	An internal error occurred.	Contact technical support.
400	APIG. 0101	The API does not exist or has not been published.	The API does not exist or has not been published.	Check whether the entered API information is correct and whether the service is available in the corresponding region. Check whether the URL of the API is spelled correctly and whether the HTTP request method (such as POST and GET) is correct. Check whether the domain name and URI configured for service call are correct.

Status Code	Error Code	Description	Description	Handling Measure
400	APIG.0301	1. Incorrect IAM authentication information: Failed to decrypt the token. Check whether the entered token is complete. 2. The validity period of a token is 24 hours. If the token expires, obtain a new token and pass it.	1. Incorrect IAM authentication information: Failed to decrypt the token. Check whether the entered token is complete. 2. The validity period of a token is 24 hours. If the token expires, obtain a new token and pass it.	**decrypt token fail**: The token request authentication information of **x-auth-token** in the HTTP request header is incorrect. Check the sent request and token. **token expires**: The token expires. Obtain a new token and pass it. **verify aksk signature fail**: Check whether the AK and SK are correct.
400	APIG.0201	The request body is oversized.	The request body is oversized.	Check whether the request body is oversized.
400	APIG.0308	The request is sent too fast and exceeds the default rate limit of the service.	The request is sent too fast and exceeds the default rate limit of the service.	The request is sent too fast and reaches the rate limit of the API. Reduce the request speed.
400	ModelArts.0203	Invalid token.	Invalid token.	Check whether the token is correct.
400	ModelArts.4101	The token is empty.	The token is empty.	The HTTP request header does not contain the token request authentication information of **x-auth-token**. Check the request.

Status Code	Error Code	Description	Description	Handling Measure
400	ModelArts.4102	Failed to parse the token.	Failed to parse the token.	The token request authentication information of **x-auth-token** in the HTTP request header is incorrect. Check the sent request and token.
400	ModelArts.4103	The token is invalid.	The token is invalid.	The token request authentication information of **x-auth-token** in the HTTP request header is incorrect. Check the sent request and token.
400	ModelArts.4104	The length of the request body is invalid.	The length of the request body is invalid.	Check the request body length.
400	ModelArts.4105	The JSON format of the request body is incorrect.	The JSON format of the request body is incorrect.	Check the JSON format of the request body.
400	ModelArts.4106	The account is restricted.	The account is restricted.	Check the user resource status. For details about the account restriction reason, see the description of the account center.
400	ModelArts.4107	An exception occurred when obtaining the temporary AK/SK.	An exception occurred when obtaining the temporary AK/SK.	Contact technical support.
400	ModelArts.4201	The request URL does not contain the service ID.	The request URL does not contain the service ID.	Check the service ID in the request URL.
400	ModelArts.4202	The request URL format is invalid.	The request URL format is invalid.	Check the request URL format.

Status Code	Error Code	Description	Description	Handling Measure
400	ModelArts.4203	No access permission.	No access permission.	Check the access permission.
400	ModelArts.4204	The API is not subscribed to.	The API is not subscribed to.	Subscribe to this API. If the service has been subscribed to, check whether the region where the service is subscribed to is the same as the region where the service is called. If the region is correct, check whether the URL of the API is spelled correctly and whether the HTTP request method (such as POST and GET) is correct.
400	ModelArts.4601	The external URL is invalid.	The external URL is invalid.	Check whether the entered URL is correct.
400	ModelArts.4603	The file failed to be downloaded from the external URL.	The file failed to be downloaded from the external URL.	Check whether the entered URL is correct.
400	ModelArts.4702	The OBS agency failed to be queried.	The OBS agency failed to be queried.	Check whether the OBS agency has been enabled for the service.
400	ModelArts.4703	The OBS URL is invalid.	The OBS URL is invalid.	Check the entered OBS URL.
400	ModelArts.4704	Failed to obtain the OBS file.	Failed to obtain the OBS file.	Failed to download the OBS file. Check whether the file exists.
400	ModelArts.4705	The OBS file is oversized.	The OBS file is oversized.	Use a file that meets the service size limit as the input.
400	ModelArts.4706	The OBS file does not exist.	The OBS file does not exist.	Failed to download the OBS file. Check whether the file exists.
400	Other	If other error codes are displayed, contact technical support.		