



图像识别

API 参考

发布日期 2022-08-01

---

# 目录

---

<b>1 使用前必读.....</b>	<b>1</b>
1.1 概述.....	1
1.2 调用说明.....	1
1.3 终端节点.....	1
1.4 约束与限制.....	1
1.5 基本概念.....	1
<b>2 API 概览.....</b>	<b>3</b>
<b>3 如何调用 API.....</b>	<b>4</b>
3.1 申请服务.....	4
3.2 构造请求.....	4
3.3 认证鉴权.....	7
3.4 返回结果.....	9
<b>4 API.....</b>	<b>11</b>
4.1 图像标签 (V1.0) .....	11
4.2 图像标签 (V2.0) .....	14
<b>5 应用示例.....</b>	<b>20</b>
5.1 Python3 语言 API 示例.....	20
5.2 Java 语言 API 示例.....	20
5.3 PHP 语言 API 示例.....	22
<b>6 附录.....</b>	<b>24</b>
6.1 状态码.....	24
6.2 错误码.....	26

# 1 使用前必读

## 1.1 概述

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

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

您可以使用本文档提供的图像识别服务API的描述、语法、参数说明及样例等内容，进行相关操作，支持的全部操作请参见[API概览](#)。

在调用图像识别API之前，请确保已经充分了解图像识别相关概念，详细信息请参见《图像识别产品介绍》。

## 1.2 调用说明

图像识别提供了REST（Representational State Transfer）风格API，支持您通过HTTPS请求调用，调用方法请参见[如何调用API](#)。

## 1.3 终端节点

终端节点（Endpoint）即调用API的[请求地址](#)，不同服务不同区域的终端节点不同。

## 1.4 约束与限制

更详细的限制请参见具体API的说明和“[图像识别产品介绍>使用限制](#)”。

## 1.5 基本概念

- 用户  
由账号在IAM中创建的用户，是云服务的使用人员，具有身份凭证（密码和访问密钥）。

- 区域 ( Region )  
指云资源所在的物理位置，同一区域内可用区间内网互通，不同区域间内网不互通。通过在不同地区创建云资源，可以将应用程序设计的更接近特定客户的要求，或满足不同地区的法律或其他要求。
- 可用区 ( Availability Zone )  
一个可用区是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。

# 2 API 概览

通过使用图像识别服务的自研API，您可以完整的使用图像识别服务的如[表2-1](#)所示功能。

表 2-1 接口说明

API	说明
<a href="#">图像标签 (V1.0)</a>	图像标签服务准确识别自然图片中数百种场景、上千种通用物体及其属性，让智能相册管理、照片检索和分类、基于场景内容或者物体的广告推荐等功能更加直观。

# 3 如何调用 API

## 3.1 申请服务

正式使用服务之前，必须先申请开通服务，下面是相关服务申请的步骤。

### 📖 说明

- 用户第一次使用图像识别时，需要先申请开通该服务。服务只需要开通一次即可，后面使用时无需再申请。

## 3.2 构造请求

本节介绍REST API请求的组成，并以调用IAM服务的获取用户Token接口说明如何调用API，该API获取用户的Token，Token可以用于调用其他API时鉴权。

### 请求 URI

请求URI由如下部分组成。

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

尽管请求URI包含在请求消息头中，但大多数语言或框架都要求您从请求消息中单独传递它，所以在此单独强调。

表 3-1 URI 中的参数说明

参数	描述
URI-scheme	表示用于传输请求的协议，当前所有API均采用HTTPS协议。
Endpoint	指定承载REST服务端点的服务器域名或IP，不同服务不同区域的Endpoint不同，您可以从管理员处中获取。
resource-path	资源路径，也即API访问路径。 从具体API的URI模块获取，例如“获取用户Token”API的resource-path为“/v3/auth/tokens”。

参数	描述
Query string	查询参数，是可选部分，并不是每个API都有查询参数。查询参数前面需要带一个“？”，形式为“参数名=参数取值”，例如“limit=10”，表示查询不超过10条数据。

### 📖 说明

为查看方便，在每个具体API的URI部分，只给出resource-path部分，并将请求方法写在一起。这是因为URI-scheme都是HTTPS，而Endpoint在同一个区域也相同，所以简洁起见将这两部分省略。

## 请求方法

HTTP请求方法（也称为操作或动词），可以向服务表明您正在请求什么类型的操作。

表 3-2 HTTP 方法

方法	说明
GET	请求服务器返回指定资源。
PUT	请求服务器更新指定资源。
POST	请求服务器新增资源或执行特殊操作。
DELETE	请求服务器删除指定资源，如删除对象等。
HEAD	请求服务器资源头部。
PATCH	请求服务器更新资源的部分内容。 当资源不存在的时候，PATCH可能会去创建一个新的资源。

## 请求消息头

附加请求头字段，如指定的URI和HTTP方法所要求的字段。例如定义消息体类型的请求头“Content-Type”，请求鉴权信息等。

如下公共消息头需要添加到请求中。

表 3-3 公共请求消息头

名称	描述	是否必选	示例
X-Sdk-Date	请求的发生时间，格式(YYYYMMDD'T'HHMMSS'Z')。 取值为当前系统的GMT时间。	否 使用AK/SK认证时必选。	20150907T101459Z

名称	描述	是否必选	示例
Authorization	签名认证信息。 该值来源于请求签名结果。	否 使用AK/SK认证时必选。	SDK-HMAC-SHA256 Credential=ZIRKMTWPTQFQI1WKNKB/20150907//ec2/sdk_request, SignedHeaders=content-type;host;x-sdk-date, Signature=55741b610f3c9fa3ae40b5a8021ebf7ebc2a28a603fc62d25cb3bfe6608e1994
Host	请求的服务器信息，从服务API的URL中获取。值为hostname[:port]。 端口缺省时使用默认的端口，https的默认端口为443。	否 使用AK/SK认证时必选。	code.test.com or code.test.com:443
Content-Type	发送的实体的MIME类型。	是	application/json
Content-Length	请求body长度，单位为Byte。	POST/PUT请求必填。GET不能包含。	3495
X-Project-Id	project id，用于不同project取token。 如果是DeC的请求或者多project的请求则必须传入project id。	否	e9993fc787d94b6c886cbaa340f9c0f4
X-Auth-Token	用户Token。	否 使用Token认证时必选。	-

### 📖 说明

API同时支持使用AK/SK认证，AK/SK认证是使用SDK对请求进行签名，签名过程会自动往请求中添加Authorization（签名认证信息）和X-Sdk-Date（请求发送的时间）请求头。

其它header属性，请遵照https协议。

## 请求消息体

请求消息体通常以结构化格式（如JSON或XML）发出，与请求消息头中Content-type对应，传递除请求消息头之外的内容。若请求消息体中参数支持中文，则中文字符必须为UTF-8编码。

每个接口的请求消息体内容不同，也并不是每个接口都需要有请求消息体（或者说消息体为空），GET、DELETE操作类型的接口就不需要消息体，消息体具体内容需要根据具体接口而定。

例如，对于获取用户Token接口接口，您可以从接口的请求部分看到所需的请求参数及参数说明。将消息体加入后的请求如下所示，加粗的斜体字段需要根据实际值填写，其中 *username* 为用户名，*domainname* 为用户所属的账号名称，*\*\*\*\*\** 为用户登录密码，*xxxxx* 为 project 的 name。

#### 📖 说明

scope 参数定义了 Token 的作用域，上面示例中获取的 Token 仅能访问 project 下的资源。您还可以设置 Token 作用域为某个账号下所有资源或账号的某个 project 下的资源，详细定义请参见获取用户 Token。

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

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

到这里为止这个请求需要的内容就具备齐全了，您可以使用 curl、Postman 或直接编写与代码等方式发送请求调用 API。对于获取用户 Token 接口，返回的响应消息头中“x-subject-token”就是需要获取的用户 Token。有了 Token 之后，您就可以使用 Token 认证调用其他 API。

## 3.3 认证鉴权

调用接口有如下两种认证方式，您可以选择其中一种进行认证鉴权。

- Token 认证：通过 Token 认证通用请求。
- AK/SK 认证：通过 AK ( Access Key ID ) /SK ( Secret Access Key) 加密调用请求。

### Token 认证

#### 📖 说明

Token 的有效期为 24 小时，需要使用一个 Token 鉴权时，可以先缓存起来，避免频繁调用。

Token在计算机系统中代表令牌（临时）的意思，拥有Token就代表拥有某种权限。Token认证就是在调用API的时候将Token加到请求消息头，从而通过身份认证，获得操作API的权限。

Token可通过调用获取用户Token接口获取，调用本服务API需要project级别的Token，即调用获取用户Token接口时，请求body中auth.scope的取值需要选择project，如下所示。

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****", //此处密码为当前账号的密码，如果您登录的是子账户，请使用子账户的密码。
        }
      }
    },
    "domain": {
      "name": "domainname"
    }
  },
  "scope": {
    "project": {
      "name": "xxxxxxx"
    }
  }
}
```

获取Token后，再调用其他接口时，您需要在请求消息头中添加“X-Auth-Token”，其值即为Token。例如Token值为“ABCDEFJ....”，则调用接口时将“X-Auth-Token: ABCDEFJ....”加到请求消息头即可，如下所示。

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

## AK/SK 认证

### 📖 说明

AK/SK签名认证、Token认证方式仅支持消息体大小12M以内的请求使用。

AK/SK认证就是使用AK/SK对请求进行签名，在请求时将签名信息添加到消息头，从而通过身份认证。

- AK(Access Key ID)：访问密钥ID。与私有访问密钥关联的唯一标识符；访问密钥ID和私有访问密钥一起使用，对请求进行加密签名。
- SK(Secret Access Key)：与访问密钥ID结合使用的密钥，对请求进行加密签名，可标识发送方，并防止请求被修改。

使用AK/SK认证时，您可以基于签名算法使用AK/SK对请求进行签名。

## 3.4 返回结果

### 状态码

请求发送以后，您会收到响应，包含状态码、响应消息头和消息体。

状态码是一组从1xx到5xx的数字代码，状态码表示了请求响应的状态，完整的状态码列表请参见[状态码](#)。

对于获取用户Token接口，如果调用后返回状态码为“201”，则表示请求成功。

### 响应消息头

对应请求消息头，响应同样也有消息头，如“Content-type”。

表 3-4 响应消息头

名称	描述
Content-Length	响应消息体的字节长度，单位为Byte。
Date	系统响应的时间。
Content-Type	响应消息体的MIME类型。

对于获取用户Token接口，返回的消息头，其中“x-subject-token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

### 响应消息体

响应消息体通常以结构化格式（如JSON或XML）返回，与响应消息头中Content-type对应，传递除响应消息头之外的内容。

对于获取用户Token接口，返回如下消息体。为篇幅起见，这里只展示部分内容。

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

```

当接口调用出错时，会返回错误码及错误信息说明，错误响应的Body体格式如下所示。

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

其中，“error\_code”表示错误码，“error\_msg”表示错误描述信息。

# 4 API

## 4.1 图像标签 (V1.0)

### 功能介绍

图像标签服务准确识别自然图片中数百种场景、上千种通用物体及其属性。让智能相册管理、照片检索和分类、基于场景内容或者物体的广告推荐等功能更加直观。使用时用户发送待处理图片，返回图片标签内容及相应置信度。

### 前提条件

- 在使用图像标签服务之前需要您完成[服务申请](#)和[认证鉴权](#)。

### URI

URI格式

POST /v1.0/image/tagging

### 请求消息

请求参数请参见[表4-1](#)。

表 4-1 请求参数说明

名称	是否必选	类型	说明
image	false	String	与url二选一。 图像数据，base64编码，要求base64编码后大小不超过10MB，最短边至少15px，最长边最大4096px，支持JPG、PNG、BMP格式。

名称	是否必选	类型	说明
url	false	String	与image二选一。 图片的URL路径，目前支持： <ul style="list-style-type: none"> <li>• 公网HTTP/HTTPS URL</li> <li>• OBS提供的URL，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权</li> </ul> <b>说明</b> <ul style="list-style-type: none"> <li>• 接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。</li> <li>• 请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。</li> <li>• Image不支持跨区域OBS，OBS的区域需要和服务保持一致。</li> </ul>
language	false	String	zh：返回标签的语言类型为中文。 en：返回标签的语言类型为英文。 默认值为zh。
limit	false	Integer	最多返回的标签数，默认值：30。
threshold	false	Float	置信度的阈值（0~100），输入非该范围内值算法即取默认值。 默认值：0。

## 响应消息

响应参数请参见[表4-2](#)。

表 4-2 响应参数说明

名称	类型	说明
result	JSON	调用成功时为图片标签内容。 调用失败时无此字段。
tags	List	标签列表集合。
confidence	Float	置信度，取值范围（0~100）。
tag	String	标签名称。
type	String	标签的类别。有以下三种类别： <ul style="list-style-type: none"> <li>• object：实体标签</li> <li>• scene：场景标签</li> <li>• concept：概念标签</li> </ul>

名称	类型	说明
i18n_tag	JSON	标签的国际化字段。（i18n只是个国际化标志，无特殊含义）。 <ul style="list-style-type: none"> <li>zh: 中文</li> <li>en: 英文</li> </ul>
error_code	String	调用失败时的错误码，具体请参见 <a href="#">错误码</a> 。 调用成功时无此字段。
error_msg	String	调用失败时的错误信息。 调用成功时无此字段。

## 示例

- 请求示例（方式一：使用图片的BASE64编码）

POST https://{{image的Endpoint}}/v1.0/image/tagging

Request Header:

Content-Type:application/json

X-Auth-Token:

MIINRwYJKoZlIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwggUwBgcqhkiG...

Request Body:

```
{
  "image":"/9j/4AAQSkZJRgABAgEASABIAAD/
  4RFZRXhpZgAATU0AKgAAAQABWESAAMAAAABAAEAAAEaAAUAAAABAAAAYgEBAAUAAAABAAAAG
  EoAAAMAAAABAAIAAAExAAIAAAACAAAQcEYAAIAAAAUAAAQAAAAABAAAAPAAAAAACVYAAA
  AnEAAK/
  IAAACcQQWRvYmUgUGhvdG9zaG9wIENTMXBxW5kb3dzADlwMTc6MTA6MjAgMTA6NDU6MzYAAAA
  AA6ABAAMAAAAB//
  8AAKACAAQAAAABAAALIKADAAQAAAAABAAAGQAAAAAAAAAAGAQMAAwAAAAEABgAAARoABQAAA
  AEAAAEEARsABQAAAAEAAAEEmASgAAwAAAAEAAAGAAAgEABAAAAEAAAEuAglABAAAAEAAABAj...",
  "url": "",
  "language": "en",
  "limit": 5,
  "threshold": 60.0
}
```

- 请求示例（方式二：使用图片URL）

POST https://{{image的Endpoint}}/v1.0/image/tagging

Request Header:

Content-Type:application/json

X-Auth-Token:

MIINRwYJKoZlIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwggUwBgcqhkiG...

Request Body:

```
{
  "image": "",
  "url": "https://{{图片的OBS存储路径}}",
  "language": "en",
  "limit": 5,
  "threshold": 60.0
}
```

- 成功响应示例

```
{
  "result": {
    "tags": [
      {
        "confidence": 38.51,

```

```

    "tag": "sky",
    "i18n_tag":
    {
      "en": "sky",
      "zh": "天空"
    },
    "type": "object"
  },
  {
    "confidence": 25.75,
    "tag": "landscape",
    "i18n_tag":
    {
      "en": "landscape",
      "zh": "风景"
    },
    "type": "scene"
  }
]
}

```

- 失败响应示例

```

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

```

## 返回值

- 正常  
200
- 失败

返回值	说明
400	<ul style="list-style-type: none"> <li>语义有误，当前请求无法被服务器解析。除非进行修改，否则客户端不应重复提交这个请求。</li> <li>请求参数有误。</li> </ul>
401	当前请求需要用户验证。
403	没有操作权限。
404	请求失败，在服务器上未找到请求所希望得到的资源。
500	服务器遇到了一个未曾预料的状态，导致无法完成对请求的处理。

## 错误码

错误码请参见 [错误码](#)。

## 4.2 图像标签 (V2.0)

### 功能介绍

图像标签服务准确识别自然图片中数百种场景、上千种通用物体，涉及数十种标签类别属性。让智能相册管理、照片检索和分类、基于场景内容或者物体的广告推荐等功

能更加直观。使用时用户发送待处理图片，返回图片标签内容、对应标签的类别属性以及相应置信度，对于常见物体也支持返回在图像中所在的位置信息。

### 📖 说明

与V1.0版本相比，V2.0版本细化了标签类别属性，对于常见物体支持了返回物体位置信息。返回信息更加完善，模型升级，准确度更高，推荐使用V2.0版本。

## 前提条件

- 在使用图像标签服务之前需要您完成[服务申请](#)和[认证鉴权](#)。

## URI

URI格式

POST /v2/{project\_id}/image/tagging

表 4-3 参数说明

名称	是否必选	说明
project_id	是	项目ID。

## 请求消息

请求参数请参见[表4-4](#)。

表 4-4 请求参数说明

名称	是否必选	类型	说明
image	false	String	与url二选一。 图像数据，base64编码，要求base64编码后大小不超过10MB，最短边至少15px，最长边最大4096px，支持JPG、PNG、BMP格式。
url	false	String	与image二选一。 图片的URL路径，目前支持： <ul style="list-style-type: none"> <li>• 公网HTTP/HTTPS URL</li> <li>• OBS提供的URL，使用OBS数据需要进行授权。包括对服务授权、临时授权、匿名公开授权</li> </ul> <b>说明</b> <ul style="list-style-type: none"> <li>• 接口响应时间依赖于图片的下载时间，如果图片下载时间过长，会返回接口调用失败。</li> <li>• 请保证被检测图片所在的存储服务稳定可靠，推荐使用OBS服务存储图片数据。</li> <li>• Image不支持跨区域OBS，OBS的区域需要和服务保持一致。</li> </ul>

名称	是否必选	类型	说明
language	false	String	zh: 返回标签的语言类型为中文。 en: 返回标签的语言类型为英文。 默认值为zh。
limit	false	Integer	最多返回的标签数, 默认值: 50。
threshold	false	Float	置信度的阈值 (0~100)。默认值: 60。

## 响应消息

响应参数请参见[表4-5](#)。

**表 4-5** 响应参数说明

名称	类型	说明
result	JSON	调用成功时为图片标签内容。 调用失败时无此字段。
tags	List	标签列表集合。
confidence	String	标签置信度, 将Float型置信度转为String类型返回, Float取值范围 (0~100)。
tag	String	根据所选的语言种类 (目前仅支持中文和英文), 返回对应语言的标签名称。
type	String	根据所选的语言种类 (目前仅支持中文和英文), 返回对应语言的标签类别。
i18n_tag	JSON	标签的国际化字段。(i18n只是个国际化标志, 无特殊含义)。 <ul style="list-style-type: none"> <li>zh: 中文</li> <li>en: 英文</li> </ul>
i18n_type	JSON	标签类别的国际化字段。(i18n只是个国际化标志, 无特殊含义)。 <ul style="list-style-type: none"> <li>zh: 中文</li> <li>en: 英文</li> </ul>

名称	类型	说明
instances	List	<p>为空时表示没有目标检测框。 非空时包含以下字段：</p> <ul style="list-style-type: none"> <li>• bounding_box：目标检测框位置。 <ul style="list-style-type: none"> <li>- width：检测框区域宽度，Float数据</li> <li>- height：检测框区域高度，Float数据</li> <li>- top_left_y：检测框左上角到垂直轴距离，Float数据</li> <li>- top_left_x：检测框左上角到水平轴距离，Float数据</li> </ul> </li> <li>• confidence：检测标签置信度，将Float型置信度转为String类型返回，Float取值范围（0~100）。</li> </ul>
error_code	String	<p>调用失败时的错误码，具体请参见<a href="#">错误码</a>。 调用成功时无此字段。</p>
error_msg	String	<p>调用失败时的错误信息。 调用成功时无此字段。</p>

## 示例

- 请求示例（方式一：使用图片的BASE64编码）

POST https://{image的Endpoint}/v2/{project\_id}/image/tagging

Request Header:

Content-Type:application/json

X-Auth-Token:

MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:

```
{
  "image": "/9j/4AAQSkZJRgABAgEASABIAAD/
4RFZRXhpZgAATU0AKgAAAABwESAAMAAAABAAEAAAEaAAUAAAABAAAAYgEBAAUAAAABAAAAG
EoAAMAAAABAAIAAAExAAIAAAACAAAACgEYAAIAAAAUAAAQjodpAAQAAAAABAAAAPAAAANAACVYAAA
AnEAAK/
IAAACcQQWRvYmUgUGhvdG9zaG9wIENTMXBxW5kb3dzADlwMTc6MTA6MjAgMTA6NDU6MzYAAAA
AA6ABAAMAAAAB//
8AAKACAAQAAAAABAAALIKADAAQAAAAABAAAGQAAAAAAAAAAGAQMAAwAAAAEABgAAARoABQAAA
AEAAAEeARsABQAAAAEAAAEeASgAAwAAAAEAAgAAAgEABAAAAEAAAEuAgIABAAAAEAAABAj...",
  "threshold": 60
}
```

- 请求示例（方式二：使用图片URL）

POST https://{image的Endpoint}/v2/{project\_id}/image/tagging

Request Header:

Content-Type:application/json

X-Auth-Token:

MIINRwYJKoZIhvcNAQcCoIINODCCDTQCAQExDTALBglghkgBZQMEAgEwgguVBgkqhkiG...

Request Body:

```
{
  "image": "",
  "url": "https://{图片的OBS存储路径}",
  "threshold": 60.0
}
```

- 成功响应示例

```

{
  "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": "人",
        "i18n_tag":
        {
          "en": "Person",
          "zh": "人"
        },
        "type": "人类",
        "i18n_type":
        {
          "en": "Human",
          "zh": "人类"
        }
      },
      {
        "confidence": "94.38",
        "instances":
        [],
        "tag": "书本",
        "i18n_tag":
        {
          "en": "Book",
          "zh": "书本"
        },
        "type": "教育",
        "i18n_type":
        {
          "en": "Education",
          "zh": "教育"
        }
      },
      {
        "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": "床",  
  "i18n_tag":  
  {  
    "en": "Bed",  
    "zh": "床"  
  },  
  "type": "家居",  
  "i18n_type":  
  {  
    "en": "Home category",  
    "zh": "家居"  
  }  
}  
]  
}
```

- 失败响应示例

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

## 返回值

- 正常  
200
- 失败

返回值	说明
400	<ul style="list-style-type: none"><li>● 语义有误，当前请求无法被服务器解析。除非进行修改，否则客户端不应重复提交这个请求。</li><li>● 请求参数有误。</li></ul>
401	当前请求需要用户验证。
403	没有操作权限。
404	请求失败，在服务器上未找到请求所希望得到的资源。
500	服务器遇到了一个未曾预料的状态，导致无法完成对请求的处理。

## 错误码

错误码请参见 [错误码](#)。

# 5 应用示例

## 5.1 Python3 语言 API 示例

本示例以图像标签为例介绍如何使用Python3调用API。

```
# encoding:utf-8

import requests
import base64

url = "https://{endpoint}/v1.0/image/tagging"
token = "用户获取得到的实际token值"
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") # 使用图片的base64编码
data= {"image": image_base64} # url与image参数二选一

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

表 5-1 参数说明

参数	参数说明
url	API请求URL，例如本示例中https://{endpoint}/v1.0/image/tagging。
token	Token是用户的访问令牌，承载了用户的身份、权限等信息，用户调用API接口时，需要使用Token进行鉴权。 获取Token方法请参见。
imagePath	图片路径。支持图片文件路径或图片url路径。其中，图片的url路径支持公网http/https url或OBS提供的url。

## 5.2 Java 语言 API 示例

本示例以图像标签为例介绍如何使用JAVA调用API。

```

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;

/**
 * 此demo仅供测试使用，强烈建议使用SDK
 * 使用前需配置依赖jar包。jar包可通过下载SDK获取
 */

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 = "用户获取到的实际token值";
        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();
        }
    }
}

```

表 5-2 参数说明

参数	参数说明
url	API请求URL，例如本示例中https://{endpoint}/v1.0/image/tagging。

参数	参数说明
token	Token是用户的访问令牌，承载了用户的身份、权限等信息，用户调用API接口时，需要使用Token进行鉴权。 获取Token方法请参见。
imgPath	图片路径。支持图片文件路径或图片url路径。其中，图片的url路径支持公网http/https url或OBS提供的url。

## 5.3 PHP 语言 API 示例

本示例以图像标签为例介绍如何使用PHP调用API。

```
<?php
function TokenRequest() {
    $url = "https://{endpoint}/v1.0/image/tagging";
    $token = "用户获取得到的实际token值";
    $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 "图片读取错误";
            return;
        }
        $data['image'] = $fileBase64;
    }

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

    /* 设置请求体 */
    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();
```

表 5-3 参数说明

参数	参数说明
url	API请求URL，例如本示例中https://{endpoint}/v1.0/image/tagging。
token	Token是用户的访问令牌，承载了用户的身份、权限等信息，用户调用API接口时，需要使用Token进行鉴权。 获取Token方法请参见。
imagePath	图片路径。支持图片文件路径或图片url路径。其中，图片的url路径支持公网http/https url或OBS提供的url。

# 6 附录

## 6.1 状态码

状态码如表6-1所示。

表 6-1 状态码

状态码	编码	错误码说明
100	Continue	继续请求。 这个临时响应用来通知客户端，它的部分请求已经被服务器接收，且仍未被拒绝。
101	Switching Protocols	切换协议。只能切换到更高级的协议。例如，切换到HTTPS的新版本协议。
200	OK	服务器已成功处理了请求。
201	Created	创建类的请求完全成功。
202	Accepted	已经接受请求，但未处理完成。
203	Non-Authoritative Information	非授权信息，请求成功。
204	No Content	请求完全成功，同时HTTP响应不包含响应体。 在响应OPTIONS方法的HTTP请求时返回此状态码。
205	Reset Content	重置内容，服务器处理成功。
206	Partial Content	服务器成功处理了部分GET请求。
300	Multiple Choices	多种选择。请求的资源可包括多个位置，相应可返回一个资源特征与地址的列表用于用户终端（例如：浏览器）选择。
301	Moved Permanently	永久移动，请求的资源已被永久的移动到新的URI，返回信息会包括新的URI。

状态码	编码	错误码说明
302	Found	资源被临时移动。
303	See Other	查看其它地址。 使用GET和POST请求查看。
304	Not Modified	所请求的资源未修改，服务器返回此状态码时，不会返回任何资源。
305	Use Proxy	所请求的资源必须通过代理访问。
306	Unused	已经被废弃的HTTP状态码。
400	Bad Request	非法请求。 建议直接修改该请求，不要重试该请求
401	Unauthorized	在客户端提供认证信息后，返回该状态码，表明服务端指出客户端所提供的认证信息不正确或非法。
402	Payment Required	保留请求。
403	Forbidden	请求被拒绝访问。 返回该状态码，表明请求能够到达服务端，且服务端能够理解用户请求，但是拒绝做更多的事情，因为该请求被设置为拒绝访问，建议直接修改该请求，不要重试该请求。
404	Not Found	所请求的资源不存在。 建议直接修改该请求，不要重试该请求。
405	Method Not Allowed	请求中带有该资源不支持的方法。 建议直接修改该请求，不要重试该请求。
406	Not Acceptable	服务器无法根据客户端请求的内容特性完成请求。
407	Proxy Authentication Required	请求要求代理的身份认证，与401类似，但请求者应当使用代理进行授权。
408	Request Timeout	服务器等候请求时发生超时。 客户端可以随时再次提交该请求而无需进行任何更改。
409	Conflict	服务器在完成请求时发生冲突。 返回该状态码，表明客户端尝试创建的资源已经存在，或者由于冲突请求的更新操作不能被完成。
410	Gone	客户端请求的资源已经不存在。 返回该状态码，表明请求的资源已被永久删除。
411	Length Required	服务器无法处理客户端发送的不带Content-Length的请求信息。

状态码	编码	错误码说明
412	Precondition Failed	未满足前提条件，服务器未满足请求者在请求中设置的其中一个前提条件。
413	Request Entity Too Large	由于请求的实体过大，服务器无法处理，因此拒绝请求。为防止客户端的连续请求，服务器可能会关闭连接。如果只是服务器暂时无法处理，则会包含一个 Retry-After 的响应信息。
414	Request URI Too Long	请求的URI过长（URI通常为网址），服务器无法处理。
415	Unsupported Media Type	服务器无法处理请求附带的媒体格式。
416	Requested Range Not Satisfiable	客户端请求的范围无效。
417	Expectation Failed	服务器无法满足Expect的请求头信息。
422	Unprocessable Entity	请求格式正确，但是由于含有语义错误，无法响应。
429	Too Many Requests	表明请求超出了客户端访问频率的限制或者服务端接收到多于它能处理的请求。建议客户端读取相应的 Retry-After 首部，然后等待该首部指出的时间后再重试。
500	Internal Server Error	表明服务端能被请求访问到，但是不能理解用户的请求。
501	Not Implemented	服务器不支持请求的功能，无法完成请求。
502	Bad Gateway	充当网关或代理的服务器，从远端服务器接收到了一个无效的请求。
503	Service Unavailable	被请求的服务无效。 建议直接修改该请求，不要重试该请求。
504	Gateway Timeout	请求在给定的时间内无法完成。客户端仅在为请求指定超时（Timeout）参数时会得到该响应。
505	HTTP Version Not Supported	服务器不支持请求的HTTPS协议的版本，无法完成处理。

## 6.2 错误码

### 功能说明

所有扩展接口发生错误时，会有自定义错误信息返回，本章节为您介绍每个错误码的含义。

## 返回体格式说明

当接口调用出错时，会返回错误码及错误信息说明，返回体格式如下所示。

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

## 错误码说明

调用接口出错后，将不会返回结果数据。调用方可根据每个接口对应的错误码来定位错误原因。

返回的消息体中是具体的错误代码及错误信息。

状态码	错误码	描述信息	说明	处理措施
400	AIS.0012	The request parameter is not supported.	请求参数不支持。	请检查请求中的字段不合法，请参考对应API的请求说明。
400	AIS.0014	The JSON format of the input data is incorrect.	输入数据JSON格式错误。	请检查输入数据的JSON格式。
400	AIS.0501	The input parameter is invalid.	输入参数不合法。	请重新检查输入的参数。
400	AIS.0502	The image format is not supported.	图像格式不支持。	检查图片格式是否是服务支持的格式。
400	AIS.0503	The image is damaged.	图像受损。	检查图像是否受损，然后重新上传符合要求的图片文件。
400	AIS.0504	The image size does not meet requirements.	图像大小不符合要求。	检查图像大小是否符合要求。
400	AIS.0505	Failed to run the algorithm.	算法运行失败。	请重新检查，如需帮助请联系支持人员。
400	AIS.0506	An internal error occurred.	出现内部错误。	请联系支持人员。

状态码	错误码	描述信息	说明	处理措施
400	APIG.0101	The API does not exist or has not been published.	您所访问的API不存在或尚未在环境中发布。	请检查输入的api信息是否正确，是否该服务存在在对应的区域中。检查API的URL是否拼写正确，HTTP请求方法（POST，GET等）是否正确，请检查服务调用配置的域名和uri信息是否正确。
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.IAM身份验证信息不正确：解密令牌失败，检查输入的token是否完整。 2.token有效期为24小时，token超时，重新传入获取token值。	decrypt token fail: 发送的HTTP请求header中x-auth-token的token请求认证信息错误，请检查发送的请求和token。 token expires: token超时，请重新获取新token传入。 verify aksk signature fail: 请检查AK、SK是否正确。
400	APIG.0201	The request body is oversized.	请求体过大。	请检查请求的body体是否过大。
400	APIG.0308	The request is sent too fast and exceeds the default rate limit of the service.	请求发送太快，超过了服务的默认配置限流。	请求发送太快，达到了API的限速阈值，请降低请求速度。
400	ModelArts.0203	Invalid token.	非法Token。	请检查Token是否正确。
400	ModelArts.4101	The token is empty.	Token为空。	发送的HTTP请求header中没有带x-auth-token的token请求认证信息，请检查发送的请求。

状态码	错误码	描述信息	说明	处理措施
400	ModelArts.4102	Failed to parse the token.	解析Token失败。	发送的HTTP请求header中x-auth-token的token请求认证信息错误，请检查发送的请求和token。
400	ModelArts.4103	The token is invalid.	Token不合法。	发送的HTTP请求header中x-auth-token的token请求认证信息错误，请检查发送的请求和token。
400	ModelArts.4104	The length of the request body is invalid.	请求body体长度不合法。	请检查请求body体长度。
400	ModelArts.4105	The JSON format of the request body is incorrect.	请求body体json格式不对。	请检查请求body体是否符合json格式。
400	ModelArts.4106	The account is restricted.	用户账号受限。	请检查用户资源状态，账号受限原因请参考账号中心相关描述。
400	ModelArts.4107	An exception occurred when obtaining the temporary AK/SK.	获取用户临时AK, SK异常。	请联系支持人员。
400	ModelArts.4201	The request URL does not contain the service ID.	请求url中需要包含服务ID。	请检查请求url中的服务ID。
400	ModelArts.4202	The request URL format is invalid.	请求url格式不合法。	请检查请求url格式。
400	ModelArts.4203	No access permission.	没有权限访问。	请检查访问权限。

状态码	错误码	描述信息	说明	处理措施
400	ModelArts.4204	The API is not subscribed to.	没有订阅该API。	请订阅该API。如已开通服务，请检查开通服务的区域与服务调用的一致，若一致请检查API的URL是否拼写正确，HTTP请求方法（POST，GET等）是否正确。
400	ModelArts.4601	The external URL is invalid.	外网下载地址不合法。	请检查输入的url下载地址参数格式。
400	ModelArts.4603	The file failed to be downloaded from the external URL.	下载外网URL文件失败。	请检查输入的url是否可下载。
400	ModelArts.4702	The OBS agency failed to be queried.	查询OBS委托失败。	请检查是否已开通给服务的OBS委托。
400	ModelArts.4703	The OBS URL is invalid.	OBS地址不合法。	请检查输入的OBS的url地址。
400	ModelArts.4704	Failed to obtain the OBS file.	获取OBS文件失败。	下载OBS文件失败，请检查对应的文件是否存在。
400	ModelArts.4705	The OBS file is oversized.	OBS文件大小超限。	请使用符合服务大小限制的文件作为输入。
400	ModelArts.4706	The OBS file does not exist.	OBS文件不存在。	下载OBS文件失败，请检查对应的文件是否存在。
400	其他	如若出现其他无法解决的错误码，请联系支持人员。		