本文档强烈推荐使用Web版式视图阅读

在菜单栏点击“视图”🡪“Web版式视图”完成设置

API参考

1.简介

Hi，您好，欢迎使用百度图像识别API服务。

本文档主要针对API开发者，描述百度图像识别接口服务的相关技术内容。如果您对文档内容有任何疑问，可以通过以下几种方式联系我们：

* 在百度云控制台内[提交工单](http://ticket.bce.baidu.com/#/ticket/create)，咨询问题类型请选择**人工智能服务**；
* 如有疑问，进入\*\*[AI社区交流](http://ai.baidu.com/forum/topic/list/171) \*\*:<http://ai.baidu.com/forum/topic/list/171>
* 具有免费调用额度的接口，超过每天的免费额度后会返回错误码：17，错误信息：Open api daily request limit reached（每天流量超限额）；

1.1.接口能力

| 产品功能 | 接口名称 | 接口能力简要描述 |
| --- | --- | --- |
| 通用图像分析 | 图像主体检测 | 识别图像中的主体具体坐标位置。 |
|  | 通用物体和场景识别 | 识别图片中的场景及物体标签，支持8w+标签类型。 |
| 细粒度图像识别 | 菜品识别 | 检测用户上传的菜品图片，返回具体的菜名、卡路里、置信度信息。 |
|  | 车型识别 | 检测用户上传的车辆图片，识别所属车型，包括车辆品牌及具体型号、颜色、年份、位置信息。 |
|  | logo商标识别 | 识别图片中包含的商品LOGO信息，返回LOGO品牌名称、在图片中的位置、置信度。 |
|  | 动物识别 | 检测用户上传的动物图片，返回动物名称、置信度信息。 |
|  | 植物识别 | 检测用户上传的植物图片，返回植物名称、置信度信息。 |
| EasyDL定制化图像识别 | 定制图像分类 | 自主训练图像分类模型，识别一张图中是否是某类物体/状态/场景，可以识别图片中主体单一的场景。 |
|  | 定制物体检测 | 自主训练物体检测模型，在一张图包含多个物体的情况下，定制识别出每个物体的位置、数量、名称。可以识别图片中有多个主体的场景。 |

|

1.2.请求格式

POST方式调用

**注意**：Content-Type为application/x-www-form-urlencoded，然后通过urlencode格式化请求体。

1.3.返回格式

JSON格式

1.4.请求限制

请求图片需经过base64编码：图片的base64编码指将一副图片数据编码成一串字符串，使用该字符串代替图像地址。您可以首先得到图片的二进制，然后用Base64格式编码即可。

**注意**：图片的base64编码是不包含图片头的，如（data:image/jpg;base64,）

**请求格式支持**：PNG、JPG、JPEG、BMP、GIF\*\*

| 接口名称 | 图片编码后大小限额 |
| --- | --- |
| 通用物体和场景识别 | base64编码后大小不超过4M，最短边至少30px，最长边最大4096px，长宽比3：1以内 |
| 菜品识别 | base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，长宽比3：1以内 |
| 车型识别 | base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，长宽比3：1以内 |
| logo商标识别 | base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，长宽比3：1以内 |
| 图像主体检测 | base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，长宽比3：1以内 |
| 动物识别 | base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，长宽比3：1以内 |
| 植物识别 | base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，长宽比3：1以内 |

2.调用方式

调用AI服务相关的API接口有两种调用方式，两种不同的调用方式采用相同的接口URL。

区别在于**请求方式**和**鉴权方法**不一样，请求参数和返回结果一致。

2.1.调用方式一

**请求URL数据格式**

向API服务地址使用POST发送请求，必须在URL中带上参数：

**access\_token:** 必须参数，参考“[Access Token获取](http://ai.baidu.com/docs#/Auth)”。

注意：access\_token的有效期为30天，需要每30天进行定期更换；

POST中参数按照API接口说明调用即可。

例如菜品识别API，使用HTTPS POST发送：

https://aip.baidubce.com/rest/2.0/image-classify/v2/dish?access\_token=24.f9ba9c5241b67688bb4adbed8bc91dec.2592000.1485570332.282335-8574074

**获取access\_token示例代码**

Bash

#!/bin/bash

curl -i -k 'https://aip.baidubce.com/oauth/2.0/token?grant\_type=client\_credentials&client\_id=【百度云应用的AK】&client\_secret=【百度云应用的SK】'

说明：方式一鉴权使用的Access\_token必须通过API Key和Secret Key获取。

**获取access\_token示例代码**

PHP

<?php

function request\_post($url = '', $param = '') {

if (empty($url) || empty($param)) {

return false;

}

$postUrl = $url;

$curlPost = $param;

$curl = curl\_init();//初始化curl

curl\_setopt($curl, CURLOPT\_URL,$postUrl);//抓取指定网页

curl\_setopt($curl, CURLOPT\_HEADER, 0);//设置header

curl\_setopt($curl, CURLOPT\_RETURNTRANSFER, 1);//要求结果为字符串且输出到屏幕上

curl\_setopt($curl, CURLOPT\_POST, 1);//post提交方式

curl\_setopt($curl, CURLOPT\_POSTFIELDS, $curlPost);

$data = curl\_exec($curl);//运行curl

curl\_close($curl);

return $data;

}

$url = 'https://aip.baidubce.com/oauth/2.0/token';

$post\_data['grant\_type'] = 'client\_credentials';

$post\_data['client\_id'] = '你的 Api Key';

$post\_data['client\_secret'] = '你的 Secret Key';

$o = "";

foreach ( $post\_data as $k => $v )

{

$o.= "$k=" . urlencode( $v ). "&" ;

}

$post\_data = substr($o,0,-1);

$res = request\_post($url, $post\_data);

var\_dump($res);

?>

说明：方式一鉴权使用的Access\_token必须通过API Key和Secret Key获取。

**获取access\_token示例代码**

Java

package com.baidu.ai.aip.auth;

import org.json.JSONObject;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

import java.util.List;

import java.util.Map;

/\*\*

\* 获取token类

\*/

public class AuthService {

/\*\*

\* 获取权限token

\* @return 返回示例：

\* {

\* "access\_token": "24.460da4889caad24cccdb1fea17221975.2592000.1491995545.282335-1234567",

\* "expires\_in": 2592000

\* }

\*/

public static String getAuth() {

// 官网获取的 API Key 更新为你注册的

String clientId = "百度云应用的AK";

// 官网获取的 Secret Key 更新为你注册的

String clientSecret = "百度云应用的SK";

return getAuth(clientId, clientSecret);

}

/\*\*

\* 获取API访问token

\* 该token有一定的有效期，需要自行管理，当失效时需重新获取.

\* @param ak - 百度云官网获取的 API Key

\* @param sk - 百度云官网获取的 Securet Key

\* @return assess\_token 示例：

\* "24.460da4889caad24cccdb1fea17221975.2592000.1491995545.282335-1234567"

\*/

public static String getAuth(String ak, String sk) {

// 获取token地址

String authHost = "https://aip.baidubce.com/oauth/2.0/token?";

String getAccessTokenUrl = authHost

// 1. grant\_type为固定参数

+ "grant\_type=client\_credentials"

// 2. 官网获取的 API Key

+ "&client\_id=" + ak

// 3. 官网获取的 Secret Key

+ "&client\_secret=" + sk;

try {

URL realUrl = new URL(getAccessTokenUrl);

// 打开和URL之间的连接

HttpURLConnection connection = (HttpURLConnection) realUrl.openConnection();

connection.setRequestMethod("GET");

connection.connect();

// 获取所有响应头字段

Map<String, List<String>> map = connection.getHeaderFields();

// 遍历所有的响应头字段

for (String key : map.keySet()) {

System.err.println(key + "--->" + map.get(key));

}

// 定义 BufferedReader输入流来读取URL的响应

BufferedReader in = new BufferedReader(new InputStreamReader(connection.getInputStream()));

String result = "";

String line;

while ((line = in.readLine()) != null) {

result += line;

}

/\*\*

\* 返回结果示例

\*/

System.err.println("result:" + result);

JSONObject jsonObject = new JSONObject(result);

String access\_token = jsonObject.getString("access\_token");

return access\_token;

} catch (Exception e) {

System.err.printf("获取token失败！");

e.printStackTrace(System.err);

}

return null;

}

}

说明：方式一鉴权使用的Access\_token必须通过API Key和Secret Key获取。

**获取access\_token示例代码**

Python

import ssl

# client\_id 为官网获取的AK， client\_secret 为官网获取的SK

host = 'https://aip.baidubce.com/oauth/2.0/token?grant\_type=client\_credentials&client\_id=【官网获取的AK】&client\_secret=【官网获取的SK】'

request = urllib2.Request(host)

request.add\_header('Content-Type', 'application/json; charset=UTF-8')

response = urllib2.urlopen(request)

content = response.read()

if (content):

print(content)

说明：方式一鉴权使用的Access\_token必须通过API Key和Secret Key获取。

**获取access\_token示例代码**

C++

#include <iostream>

#include <curl/curl.h>

#include <json/json.h>

#include "access\_token.h"

// libcurl库下载链接：https://curl.haxx.se/download.html

// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/

// 获取access\_token所需要的url

const std::string access\_token\_url = "https://aip.baidubce.com/oauth/2.0/token?grant\_type=client\_credentials";

// 回调函数获取到的access\_token存放变量

// static std::string access\_token\_result;

/\*\*

\* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在result中

\* @param 参数定义见libcurl库文档

\* @return 返回值定义见libcurl库文档

\*/

static size\_t callback(void \*ptr, size\_t size, size\_t nmemb, void \*stream) {

// 获取到的body存放在ptr中，先将其转换为string格式

std::string s((char \*) ptr, size \* nmemb);

// 开始获取json中的access token项目

Json::Reader reader;

Json::Value root;

// 使用boost库解析json

reader.parse(s,root);

std::string\* access\_token\_result = static\_cast<std::string\*>(stream);

\*access\_token\_result = root["access\_token"].asString();

return size \* nmemb;

}

/\*\*

\* 用以获取access\_token的函数，使用时需要先在百度云控制台申请相应功能的应用，获得对应的API Key和Secret Key

\* @param access\_token 获取得到的access token，调用函数时需传入该参数

\* @param AK 应用的API key

\* @param SK 应用的Secret key

\* @return 返回0代表获取access token成功，其他返回值代表获取失败

\*/

int get\_access\_token(std::string &access\_token, const std::string &AK, const std::string &SK) {

CURL \*curl;

CURLcode result\_code;

int error\_code = 0;

curl = curl\_easy\_init();

if (curl) {

std::string url = access\_token\_url + "&client\_id=" + AK + "&client\_secret=" + SK;

curl\_easy\_setopt(curl, CURLOPT\_URL, url.data());

curl\_easy\_setopt(curl, CURLOPT\_SSL\_VERIFYPEER, 0);

curl\_easy\_setopt(curl, CURLOPT\_SSL\_VERIFYHOST, 0);

std::string access\_token\_result;

curl\_easy\_setopt(curl, CURLOPT\_WRITEDATA, &access\_token\_result);

curl\_easy\_setopt(curl, CURLOPT\_WRITEFUNCTION, callback);

result\_code = curl\_easy\_perform(curl);

if (result\_code != CURLE\_OK) {

fprintf(stderr, "curl\_easy\_perform() failed: %s\n",

curl\_easy\_strerror(result\_code));

return 1;

}

access\_token = access\_token\_result;

curl\_easy\_cleanup(curl);

error\_code = 0;

} else {

fprintf(stderr, "curl\_easy\_init() failed.");

error\_code = 1;

}

return error\_code;

}

说明：方式一鉴权使用的Access\_token必须通过API Key和Secret Key获取。

**获取access\_token示例代码**

C#

﻿using System;

using System.Collections.Generic;

using System.Net.Http;

namespace com.baidu.ai

{

public static class AccessToken

{

// 调用getAccessToken()获取的 access\_token建议根据expires\_in 时间 设置缓存

// 返回token示例

public static String TOKEN = "24.adda70c11b9786206253ddb70affdc46.2592000.1493524354.282335-1234567";

// 百度云中开通对应服务应用的 API Key 建议开通应用的时候多选服务

private static String clientId = "百度云应用的AK";

// 百度云中开通对应服务应用的 Secret Key

private static String clientSecret = "百度云应用的SK";

public static String getAccessToken() {

String authHost = "https://aip.baidubce.com/oauth/2.0/token";

HttpClient client = new HttpClient();

List<KeyValuePair<String, String>> paraList = new List<KeyValuePair<string, string>>();

paraList.Add(new KeyValuePair<string, string>("grant\_type", "client\_credentials"));

paraList.Add(new KeyValuePair<string, string>("client\_id", clientId));

paraList.Add(new KeyValuePair<string, string>("client\_secret", clientSecret));

HttpResponseMessage response = client.PostAsync(authHost, new FormUrlEncodedContent(paraList)).Result;

String result = response.Content.ReadAsStringAsync().Result;

Console.WriteLine(result);

return result;

}

}

}

说明：方式一鉴权使用的Access\_token必须通过API Key和Secret Key获取。

**获取access\_token示例代码**

Node

var https = require('https');

var qs = require('querystring');

const param = qs.stringify({

'grant\_type': 'client\_credentials',

'client\_id': '您的 Api Key',

'client\_secret': '您的 Secret Key'

});

https.get(

{

hostname: 'aip.baidubce.com',

path: '/oauth/2.0/token?' + param,

agent: false

},

function (res) {

// 在标准输出中查看运行结果

res.pipe(process.stdout);

}

);

说明：方式一鉴权使用的Access\_token必须通过API Key和Secret Key获取。

2.2.调用方式二

**请求头域内容**

在请求的HTTP头域中包含以下信息：

* host（必填）
* x-bce-date （必填）
* x-bce-request-id（选填）
* authorization（必填）
* content-type（必填）
* content-length（选填）

作为示例，以下是一个标准的人脸识别的请求头域内容:

POST /rest/2.0/face/v1/detect HTTP/1.1

accept-encoding: gzip, deflate

x-bce-date: 2015-03-24T13:02:00Z

connection: keep-alive

accept: \*/\*

host: aip.baidubce.com

x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e

content-type: application/x-www-form-urlencoded

authorization: bce-auth-v1/46bd9968a6194b4bbdf0341f2286ccce/2015-03-24T13:02:00Z/1800/host;x-bce-date/994014d96b0eb26578e039fa053a4f9003425da4bfedf33f4790882fb4c54903

说明：方式二鉴权使用的[API认证机制](https://cloud.baidu.com/doc/Reference/AuthenticationMechanism.html)authorization必须通过百度云的[AK/SK](https://cloud.baidu.com/doc/Reference/GetAKSK.html)生成。

3.通用图像分析——通用物体和场景识别

3.1.接口描述

该请求用于通用物体及场景识别，即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片中的多个物体及场景标签。

3.2.请求说明

**请求示例**

HTTP 方法：POST

请求URL： https://aip.baidubce.com/rest/2.0/image-classify/v2/advanced\_general

URL参数：

| 参数 | 值 |
| --- | --- |
| access\_token | 通过API Key和Secret Key获取的access\_token,参考“[Access Token获取](http://ai.baidu.com/docs#/Auth)” |

Header如下：

| 参数 | 值 |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数，参数详情如下：

**请求参数**

| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| --- | --- | --- | --- | --- |
| image | true | string | - | Base64编码字符串，以图片文件形式请求时必填。(支持图片格式：jpg，bmp，png，jpeg)，图片大小不超过4M。最短边至少15px，最长边最大4096px。 |

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

3.3.返回说明

**返回参数**

| 字段 | 是否必选 | 类型 | 说明 |
| --- | --- | --- | --- |
| log\_id | 是 | uint64 | 唯一的log id，用于问题定位 |
| result\_num | 是 | unit32 | 返回结果数目，及result数组中的元素个数 |
| result | 是 | object() | 标签结果数组 |
| +keyword | 是 | string | 图片中的物体或场景名称 |
| +score | 是 | float | 置信度，0-1 |
| +root | 是 | string | 识别结果的上层标签，有部分钱币、动漫、烟酒等tag无上层标签 |

**返回示例**

HTTP/1.1 200 OK

x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e

Cache-Control: no-cache

Server: BWS

Date: Tue, 18 Oct 2016 02:21:01 GMT

Content-Type: application/json;charset=UTF-8

{

"result": [

{

"score": 0.996184,

"root": "商品-电脑办公",

"keyword": "台式机"

},

{

"score": 0.799921,

"root": "Logo",

"keyword": "联想"

},

{

"score": 0.424538,

"root": "商品-电脑办公",

"keyword": "电脑"

},

{

"score": 0.213236,

"root": "商品-电脑办公",

"keyword": "服务器/工作站"

},

{

"score": 0.10919,

"root": "商品-数码产品",

"keyword": "台式电脑"

}

],

"result\_num": 5,

"log\_id": 2689001477

}

4.通用图像分析——图像主体检测

4.1.接口描述

用户向服务请求检测图像中的主体位置。

4.2.请求说明

**请求示例**

HTTP 方法：POST

请求URL： https://aip.baidubce.com/rest/2.0/image-classify/v1/object\_detect

URL参数：

| 参数 | 值 |
| --- | --- |
| access\_token | 通过API Key和Secret Key获取的access\_token,参考“[Access Token获取](http://ai.baidu.com/docs#/Auth)” |

Header如下：

| 参数 | 值 |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数，参数详情如下：

**请求参数**

| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| --- | --- | --- | --- | --- |
| image | true | string | - | 图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 |
| with\_face | false | number | - | 如果检测主体是人，主体区域是否带上人脸部分，0-不带人脸区域，其他-带人脸区域，裁剪类需求推荐带人脸，检索/识别类需求推荐不带人脸。默认取1，带人脸。 |

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

bash

curl -i -k 'https://aip.baidubce.com/rest/2.0/image-classify/v1/object\_detect?access\_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】&with\_face=1' -H 'Content-Type:application/x-www-form-urlencoded'

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

PHP

/\*\*

\* 发起http post请求(REST API), 并获取REST请求的结果

\* @param string $url

\* @param string $param

\* @return - http response body if succeeds, else false.

\*/

function request\_post($url = '', $param = '')

{

if (empty($url) || empty($param)) {

return false;

}

$postUrl = $url;

$curlPost = $param;

// 初始化curl

$curl = curl\_init();

curl\_setopt($curl, CURLOPT\_URL, $postUrl);

curl\_setopt($curl, CURLOPT\_HEADER, 0);

// 要求结果为字符串且输出到屏幕上

curl\_setopt($curl, CURLOPT\_RETURNTRANSFER, 1);

curl\_setopt($curl, CURLOPT\_SSL\_VERIFYPEER, false);

// post提交方式

curl\_setopt($curl, CURLOPT\_POST, 1);

curl\_setopt($curl, CURLOPT\_POSTFIELDS, $curlPost);

// 运行curl

$data = curl\_exec($curl);

curl\_close($curl);

return $data;

}

$token = '[调用鉴权接口获取的token]';

$url = 'https://aip.baidubce.com/rest/2.0/image-classify/v1/object\_detect?access\_token=' . $token;

$img = file\_get\_contents('[本地文件路径]');

$img = base64\_encode($img);

$bodys = array(

'image' => $img,

'with\_face' => 1

);

$res = request\_post($url, $bodys);

var\_dump($res);

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

Java

package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;

import com.baidu.ai.aip.utils.FileUtil;

import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/\*\*

\* 图像主体检测

\*/

public class ObjectDetect {

/\*\*

\* 重要提示代码中所需工具类

\* FileUtil,Base64Util,HttpUtil,GsonUtils请从

\* https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72

\* https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2

\* https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3

\* https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3

\* 下载

\*/

public static String detect() {

// 请求url

String url = "https://aip.baidubce.com/rest/2.0/image-classify/v1/object\_detect";

try {

// 本地文件路径

String filePath = "[本地文件路径]";

byte[] imgData = FileUtil.readFileByBytes(filePath);

String imgStr = Base64Util.encode(imgData);

String imgParam = URLEncoder.encode(imgStr, "UTF-8");

String param = "image=" + imgParam + "&with\_face=" + 1;

// 注意这里仅为了简化编码每一次请求都去获取access\_token，线上环境access\_token有过期时间， 客户端可自行缓存，过期后重新获取。

String accessToken = "[调用鉴权接口获取的token]";

String result = HttpUtil.post(url, accessToken, param);

System.out.println(result);

return result;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public static void main(String[] args) {

ObjectDetect.detect();

}

}

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

Python

import base64

import urllib

import urllib2

'''

图像主体检测

'''

request\_url = "https://aip.baidubce.com/rest/2.0/image-classify/v1/object\_detect"

# 二进制方式打开图片文件

f = open('[本地文件]', 'rb')

img = base64.b64encode(f.read())

params = {"image":img,"with\_face":1}

params = urllib.urlencode(params)

access\_token = '[调用鉴权接口获取的token]'

request\_url = request\_url + "?access\_token=" + access\_token

request = urllib2.Request(url=request\_url, data=params)

request.add\_header('Content-Type', 'application/x-www-form-urlencoded')

response = urllib2.urlopen(request)

content = response.read()

if content:

print content

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

C++

// libcurl库下载链接：https://curl.haxx.se/download.html

// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/

const static std::string request\_url = "https://aip.baidubce.com/rest/2.0/image-classify/v1/object\_detect";

static std::string detect\_result;

/\*\*

\* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中

\* @param 参数定义见libcurl文档

\* @return 返回值定义见libcurl文档

\*/

static size\_t callback(void \*ptr, size\_t size, size\_t nmemb, void \*stream) {

// 获取到的body存放在ptr中，先将其转换为string格式

detect\_result = std::string((char \*) ptr, size \* nmemb);

return size \* nmemb;

}

/\*\*

\* 图像主体检测

\* @return 调用成功返回0，发生错误返回其他错误码

\*/

int detect(std::string &json\_result, const std::string &access\_token) {

std::string url = request\_url + "?access\_token=" + access\_token;

CURL \*curl = NULL;

CURLcode result\_code;

int is\_success;

curl = curl\_easy\_init();

if (curl) {

curl\_easy\_setopt(curl, CURLOPT\_URL, url.data());

curl\_easy\_setopt(curl, CURLOPT\_POST, 1);

curl\_httppost \*post = NULL;

curl\_httppost \*last = NULL;

curl\_formadd(&post, &last, CURLFORM\_COPYNAME, "image", CURLFORM\_COPYCONTENTS, "【base64\_img】", CURLFORM\_END);

curl\_formadd(&post, &last, CURLFORM\_COPYNAME, "with\_face", CURLFORM\_COPYCONTENTS, "1", CURLFORM\_END);

curl\_easy\_setopt(curl, CURLOPT\_HTTPPOST, post);

curl\_easy\_setopt(curl, CURLOPT\_WRITEFUNCTION, callback);

result\_code = curl\_easy\_perform(curl);

if (result\_code != CURLE\_OK) {

fprintf(stderr, "curl\_easy\_perform() failed: %s\n",

curl\_easy\_strerror(result\_code));

is\_success = 1;

return is\_success;

}

json\_result = detect\_result;

curl\_easy\_cleanup(curl);

is\_success = 0;

} else {

fprintf(stderr, "curl\_easy\_init() failed.");

is\_success = 1;

}

return is\_success;

}

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

C#

﻿using System;

using System.IO;

using System.Net;

using System.Text;

using System.Web;

namespace com.baidu.ai

{

public class ObjectDetect

{

// 图像主体检测

public static string detect()

{

string token = "[调用鉴权接口获取的token]";

string host = "https://aip.baidubce.com/rest/2.0/image-classify/v1/object\_detect?access\_token=" + token;

Encoding encoding = Encoding.Default;

HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);

request.Method = "post";

request.KeepAlive = true;

// 图片的base64编码

string base64 = FileUtils.getFileBase64("[本地图片文件]");

String str = "image=" + HttpUtility.UrlEncode(base64) + "&with\_face=" + 1;

byte[] buffer = encoding.GetBytes(str);

request.ContentLength = buffer.Length;

request.GetRequestStream().Write(buffer, 0, buffer.Length);

HttpWebResponse response = (HttpWebResponse)request.GetResponse();

StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);

string result = reader.ReadToEnd();

Console.WriteLine("图像主体检测:");

Console.WriteLine(result);

return result;

}

}

}

4.3.返回说明

**返回参数**

| 字段 | 是否必选 | 类型 | 说明 |
| --- | --- | --- | --- |
| log\_id | 是 | uint64 | 唯一的log id，用于问题定位 |
| result | 否 | watermark-location | 裁剪结果 |
| +left | 否 | uint32 | 表示定位位置的长方形左上顶点的水平坐标 |
| +top | 否 | uint32 | 表示定位位置的长方形左上顶点的垂直坐标 |
| +width | 否 | uint32 | 表示定位位置的长方形的宽度 |
| +height | 否 | uint32 | 表示定位位置的长方形的高度 |

**返回示例**

HTTP/1.1 200 OK

x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e

Cache-Control: no-cache

Server: BWS

Date: Tue, 18 Oct 2016 02:21:01 GMT

Content-Type: application/json;charset=UTF-8

{

"log\_id": 895582300,

"result": {

"width": 486,

"top": 76,

"left": 134,

"height": 394

}

}

5.细粒度图像识别——菜品识别

5.1.接口描述

该请求用于菜品识别。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的菜品名称、卡路里信息、置信度。

5.2.请求说明

**请求示例**

HTTP 方法：POST

请求URL： https://aip.baidubce.com/rest/2.0/image-classify/v2/dish

URL参数：

| 参数 | 值 |
| --- | --- |
| access\_token | 通过API Key和Secret Key获取的access\_token,参考“[Access Token获取](http://ai.baidu.com/docs#/Auth)” |

Header如下：

| 参数 | 值 |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数，参数详情如下：

**请求参数**

| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| --- | --- | --- | --- | --- |
| image | 是 | string | - | Base64编码字符串，以图片文件形式请求时必填。(支持图片格式：jpg，bmp，png，jpeg)，图片大小不超过4M。最短边至少15px，最长边最大4096px。 |
| top\_num | 否 | unit32 | - | 返回结果top n,默认5. |
| filter\_threshold | 是 | float | - | 默认0.95，可以通过该参数调节识别效果，降低非菜识别率. |

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

bash

菜品识别

curl -i -k 'https://aip.baidubce.com/rest/2.0/image-classify/v2/dish?access\_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】&top\_num=5' -H 'Content-Type:application/x-www-form-urlencoded'

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

PHP

<?php

/\*\*

\* 发起http post请求(REST API), 并获取REST请求的结果

\* @param string $url

\* @param string $param

\* @return - http response body if succeeds, else false.

\*/

function request\_post($url = '', $param = '')

{

if (empty($url) || empty($param)) {

return false;

}

$postUrl = $url;

$curlPost = $param;

// 初始化curl

$curl = curl\_init();

curl\_setopt($curl, CURLOPT\_URL, $postUrl);

curl\_setopt($curl, CURLOPT\_HEADER, 0);

// 要求结果为字符串且输出到屏幕上

curl\_setopt($curl, CURLOPT\_RETURNTRANSFER, 1);

curl\_setopt($curl, CURLOPT\_SSL\_VERIFYPEER, false);

// post提交方式

curl\_setopt($curl, CURLOPT\_POST, 1);

curl\_setopt($curl, CURLOPT\_POSTFIELDS, $curlPost);

// 运行curl

$data = curl\_exec($curl);

curl\_close($curl);

return $data;

}

$token = '[调用鉴权接口获取的token]';

$url = 'https://aip.baidubce.com/rest/2.0/image-classify/v2/dish?access\_token=' . $token;

$img = file\_get\_contents('[本地文件路径]');

$img = base64\_encode($img);

$bodys = array(

'image' => $img,

'top\_num' => 5

);

$res = request\_post($url, $bodys);

var\_dump($res);

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

Java

package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;

import com.baidu.ai.aip.utils.FileUtil;

import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/\*\*

\* 菜品识别

\*/

public class Dish {

/\*\*

\* 重要提示代码中所需工具类

\* FileUtil,Base64Util,HttpUtil,GsonUtils请从

\* https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72

\* https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2

\* https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3

\* https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3

\* 下载

\*/

public static String dish() {

// 请求url

String url = "https://aip.baidubce.com/rest/2.0/image-classify/v2/dish";

try {

// 本地文件路径

String filePath = "[本地文件路径]";

byte[] imgData = FileUtil.readFileByBytes(filePath);

String imgStr = Base64Util.encode(imgData);

String imgParam = URLEncoder.encode(imgStr, "UTF-8");

String param = "image=" + imgParam + "&top\_num=" + 5;

// 注意这里仅为了简化编码每一次请求都去获取access\_token，线上环境access\_token有过期时间， 客户端可自行缓存，过期后重新获取。

String accessToken = "[调用鉴权接口获取的token]";

String result = HttpUtil.post(url, accessToken, param);

System.out.println(result);

return result;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public static void main(String[] args) {

Dish.dish();

}

}

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

Python

# encoding:utf-8

import base64

import urllib

import urllib2

'''

菜品识别

'''

request\_url = "https://aip.baidubce.com/rest/2.0/image-classify/v2/dish"

# 二进制方式打开图片文件

f = open('[本地文件]', 'rb')

img = base64.b64encode(f.read())

params = {"image":img,"top\_num":5}

params = urllib.urlencode(params)

access\_token = '[调用鉴权接口获取的token]'

request\_url = request\_url + "?access\_token=" + access\_token

request = urllib2.Request(url=request\_url, data=params)

request.add\_header('Content-Type', 'application/x-www-form-urlencoded')

response = urllib2.urlopen(request)

content = response.read()

if content:

print content

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

C++

#include <iostream>

#include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html

// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/

const static std::string request\_url = "https://aip.baidubce.com/rest/2.0/image-classify/v2/dish";

static std::string dish\_result;

/\*\*

\* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中

\* @param 参数定义见libcurl文档

\* @return 返回值定义见libcurl文档

\*/

static size\_t callback(void \*ptr, size\_t size, size\_t nmemb, void \*stream) {

// 获取到的body存放在ptr中，先将其转换为string格式

dish\_result = std::string((char \*) ptr, size \* nmemb);

return size \* nmemb;

}

/\*\*

\* 菜品识别

\* @return 调用成功返回0，发生错误返回其他错误码

\*/

int dish(std::string &json\_result, const std::string &access\_token) {

std::string url = request\_url + "?access\_token=" + access\_token;

CURL \*curl = NULL;

CURLcode result\_code;

int is\_success;

curl = curl\_easy\_init();

if (curl) {

curl\_easy\_setopt(curl, CURLOPT\_URL, url.data());

curl\_easy\_setopt(curl, CURLOPT\_POST, 1);

curl\_httppost \*post = NULL;

curl\_httppost \*last = NULL;

curl\_formadd(&post, &last, CURLFORM\_COPYNAME, "image", CURLFORM\_COPYCONTENTS, "【base64\_img】", CURLFORM\_END);

curl\_formadd(&post, &last, CURLFORM\_COPYNAME, "top\_num", CURLFORM\_COPYCONTENTS, "5", CURLFORM\_END);

curl\_easy\_setopt(curl, CURLOPT\_HTTPPOST, post);

curl\_easy\_setopt(curl, CURLOPT\_WRITEFUNCTION, callback);

result\_code = curl\_easy\_perform(curl);

if (result\_code != CURLE\_OK) {

fprintf(stderr, "curl\_easy\_perform() failed: %s\n",

curl\_easy\_strerror(result\_code));

is\_success = 1;

return is\_success;

}

json\_result = dish\_result;

curl\_easy\_cleanup(curl);

is\_success = 0;

} else {

fprintf(stderr, "curl\_easy\_init() failed.");

is\_success = 1;

}

return is\_success;

}

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

C#

﻿using System;

using System.IO;

using System.Net;

using System.Text;

using System.Web;

namespace com.baidu.ai

{

public class Dish

{

// 菜品识别

public static string dish()

{

string token = "[调用鉴权接口获取的token]";

string host = "https://aip.baidubce.com/rest/2.0/image-classify/v2/dish?access\_token=" + token;

Encoding encoding = Encoding.Default;

HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);

request.Method = "post";

request.KeepAlive = true;

// 图片的base64编码

string base64 = FileUtils.getFileBase64("[本地图片文件]");

String str = "image=" + HttpUtility.UrlEncode(base64) + "&top\_num=" + 5;

byte[] buffer = encoding.GetBytes(str);

request.ContentLength = buffer.Length;

request.GetRequestStream().Write(buffer, 0, buffer.Length);

HttpWebResponse response = (HttpWebResponse)request.GetResponse();

StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);

string result = reader.ReadToEnd();

Console.WriteLine("菜品识别:");

Console.WriteLine(result);

return result;

}

}

}

5.3.返回说明

**返回参数**

| 字段 | 是否必选 | 类型 | 说明 |
| --- | --- | --- | --- |
| log\_id | 是 | uint64 | 唯一的log id，用于问题定位 |
| result\_num | 否 | unit32 | 返回结果数目，及result数组中的元素个数 |
| result | 否 | array() | 菜品识别结果数组 |
| +name | 否 | string | 菜名，示例：鱼香肉丝 |
| +calorie | 否 | float | 卡路里，每100g的卡路里含量 |
| +probability | 否 | float | 识别结果中每一行的置信度值，0-1 |

**返回示例**

HTTP/1.1 200 OK

x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e

Cache-Control: no-cache

Server: BWS

Date: Tue, 18 Oct 2016 02:21:01 GMT

Content-Type: application/json;charset=UTF-8

{

"log\_id": 2471272194,

"result\_num": 5,

"result": [

            {

                "name": "烧烤（串类）",

                "calorie": 333.33,

                "probability": 0.35874313116074

            },

            {

                "name": "鱿鱼",

                "calorie": 333.33,

                "probability": 0.20610593259335

            },

            {

                "name": "板筋",

                "calorie": 333.33,

                "probability": 0.15860831737518

            },

            {

                "name": "鸡脆骨",

                "calorie": 333.33,

                "probability": 0.077698558568954

            },

            {

                "name": "麻辣烫",

                "calorie": 333.33,

                "probability": 0.041968926787376

            }

    ]

}

6.细粒度图像识别—车型识别

6.1.接口描述

该请求用于检测一张车辆图片的具体车型。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的车辆品牌及型号、颜色及年份、位置信息。

6.2.请求说明

**请求示例**

HTTP 方法：POST

请求URL： https://aip.baidubce.com/rest/2.0/image-classify/v1/car

URL参数：

| 参数 | 值 |
| --- | --- |
| access\_token | 通过API Key和Secret Key获取的access\_token,参考“[Access Token获取](http://ai.baidu.com/docs#/Auth)” |

Header如下：

| 参数 | 值 |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数，参数详情如下：

**请求参数**

| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| --- | --- | --- | --- | --- |
| image | true | string | - | 图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 |
| top\_num | false | uint32 | - | 返回结果top n，默认5。 |

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

bash

车型识别

curl -i -k 'https://aip.baidubce.com/rest/2.0/image-classify/v1/car?access\_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】&top\_num=5' -H 'Content-Type:application/x-www-form-urlencoded'

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

PHP

<?php

/\*\*

\* 发起http post请求(REST API), 并获取REST请求的结果

\* @param string $url

\* @param string $param

\* @return - http response body if succeeds, else false.

\*/

function request\_post($url = '', $param = '')

{

if (empty($url) || empty($param)) {

return false;

}

$postUrl = $url;

$curlPost = $param;

// 初始化curl

$curl = curl\_init();

curl\_setopt($curl, CURLOPT\_URL, $postUrl);

curl\_setopt($curl, CURLOPT\_HEADER, 0);

// 要求结果为字符串且输出到屏幕上

curl\_setopt($curl, CURLOPT\_RETURNTRANSFER, 1);

curl\_setopt($curl, CURLOPT\_SSL\_VERIFYPEER, false);

// post提交方式

curl\_setopt($curl, CURLOPT\_POST, 1);

curl\_setopt($curl, CURLOPT\_POSTFIELDS, $curlPost);

// 运行curl

$data = curl\_exec($curl);

curl\_close($curl);

return $data;

}

$token = '[调用鉴权接口获取的token]';

$url = 'https://aip.baidubce.com/rest/2.0/image-classify/v1/car?access\_token=' . $token;

$img = file\_get\_contents('[本地文件路径]');

$img = base64\_encode($img);

$bodys = array(

'image' => $img,

'top\_num' => 5

);

$res = request\_post($url, $bodys);

var\_dump($res);

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

Java

package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;

import com.baidu.ai.aip.utils.FileUtil;

import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/\*\*

\* 车型识别

\*/

public class Car {

/\*\*

\* 重要提示代码中所需工具类

\* FileUtil,Base64Util,HttpUtil,GsonUtils请从

\* https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72

\* https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2

\* https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3

\* https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3

\* 下载

\*/

public static String car() {

// 请求url

String url = "https://aip.baidubce.com/rest/2.0/image-classify/v1/car";

try {

// 本地文件路径

String filePath = "[本地文件路径]";

byte[] imgData = FileUtil.readFileByBytes(filePath);

String imgStr = Base64Util.encode(imgData);

String imgParam = URLEncoder.encode(imgStr, "UTF-8");

String param = "image=" + imgParam + "&top\_num=" + 5;

// 注意这里仅为了简化编码每一次请求都去获取access\_token，线上环境access\_token有过期时间， 客户端可自行缓存，过期后重新获取。

String accessToken = "[调用鉴权接口获取的token]";

String result = HttpUtil.post(url, accessToken, param);

System.out.println(result);

return result;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public static void main(String[] args) {

Car.car();

}

}

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

Python

# encoding:utf-8

import base64

import urllib

import urllib2

'''

车型识别

'''

request\_url = "https://aip.baidubce.com/rest/2.0/image-classify/v1/car"

# 二进制方式打开图片文件

f = open('[本地文件]', 'rb')

img = base64.b64encode(f.read())

params = {"image":img,"top\_num":5}

params = urllib.urlencode(params)

access\_token = '[调用鉴权接口获取的token]'

request\_url = request\_url + "?access\_token=" + access\_token

request = urllib2.Request(url=request\_url, data=params)

request.add\_header('Content-Type', 'application/x-www-form-urlencoded')

response = urllib2.urlopen(request)

content = response.read()

if content:

print content

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

C++

#include <iostream>

#include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html

// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/

const static std::string request\_url = "https://aip.baidubce.com/rest/2.0/image-classify/v1/car";

static std::string car\_result;

/\*\*

\* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中

\* @param 参数定义见libcurl文档

\* @return 返回值定义见libcurl文档

\*/

static size\_t callback(void \*ptr, size\_t size, size\_t nmemb, void \*stream) {

// 获取到的body存放在ptr中，先将其转换为string格式

car\_result = std::string((char \*) ptr, size \* nmemb);

return size \* nmemb;

}

/\*\*

\* 车型识别

\* @return 调用成功返回0，发生错误返回其他错误码

\*/

int car(std::string &json\_result, const std::string &access\_token) {

std::string url = request\_url + "?access\_token=" + access\_token;

CURL \*curl = NULL;

CURLcode result\_code;

int is\_success;

curl = curl\_easy\_init();

if (curl) {

curl\_easy\_setopt(curl, CURLOPT\_URL, url.data());

curl\_easy\_setopt(curl, CURLOPT\_POST, 1);

curl\_httppost \*post = NULL;

curl\_httppost \*last = NULL;

curl\_formadd(&post, &last, CURLFORM\_COPYNAME, "image", CURLFORM\_COPYCONTENTS, "【base64\_img】", CURLFORM\_END);

curl\_formadd(&post, &last, CURLFORM\_COPYNAME, "top\_num", CURLFORM\_COPYCONTENTS, "5", CURLFORM\_END);

curl\_easy\_setopt(curl, CURLOPT\_HTTPPOST, post);

curl\_easy\_setopt(curl, CURLOPT\_WRITEFUNCTION, callback);

result\_code = curl\_easy\_perform(curl);

if (result\_code != CURLE\_OK) {

fprintf(stderr, "curl\_easy\_perform() failed: %s\n",

curl\_easy\_strerror(result\_code));

is\_success = 1;

return is\_success;

}

json\_result = car\_result;

curl\_easy\_cleanup(curl);

is\_success = 0;

} else {

fprintf(stderr, "curl\_easy\_init() failed.");

is\_success = 1;

}

return is\_success;

}

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

C#

﻿using System;

using System.IO;

using System.Net;

using System.Text;

using System.Web;

namespace com.baidu.ai

{

public class Car

{

// 车型识别

public static string car()

{

string token = "[调用鉴权接口获取的token]";

string host = "https://aip.baidubce.com/rest/2.0/image-classify/v1/car?access\_token=" + token;

Encoding encoding = Encoding.Default;

HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);

request.Method = "post";

request.KeepAlive = true;

// 图片的base64编码

string base64 = FileUtils.getFileBase64("[本地图片文件]");

String str = "image=" + HttpUtility.UrlEncode(base64) + "&top\_num=" + 5;

byte[] buffer = encoding.GetBytes(str);

request.ContentLength = buffer.Length;

request.GetRequestStream().Write(buffer, 0, buffer.Length);

HttpWebResponse response = (HttpWebResponse)request.GetResponse();

StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);

string result = reader.ReadToEnd();

Console.WriteLine("车型识别:");

Console.WriteLine(result);

return result;

}

}

}

6.3.返回说明

**返回参数**

| 字段 | 是否必选 | 类型 | 说明 |
| --- | --- | --- | --- |
| log\_id | 否 | uint64 | 唯一的log id，用于问题定位 |
| color\_result | 是 | string | 颜色 |
| result | 否 | car-result() | 车型识别结果数组 |
| +name | 否 | string | 车型名称，示例：宝马x6 |
| +score | 否 | double | 置信度，示例：0.5321 |
| +year | 否 | string | 年份 |
| location\_result | 否 | string | 车在图片中的位置信息 |

**返回示例**

HTTP/1.1 200 OK

x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e

Cache-Control: no-cache

Server: BWS

Date: Tue, 18 Oct 2016 02:21:01 GMT

Content-Type: application/json;charset=UTF-8

{

"log\_id": 3490021330,

"result": [{

"name": "别克昂科威",

"score": 0.83903276920319,

"year": "2014-2017"

},

{

"name": "通用(别克昂科拉)Mokka",

"score": 0.010498280636966,

"year": "2017"

},

{

"name": "奔驰gla级",

"score": 0.0083939796313643,

"year": "2016-2017"

}, {

"name": "日产楼兰",

"score": 0.0076470440253615,

"year": "2011-2017"

},

{

"name": "MG锐腾",

"score": 0.0065533421002328,

"year": "2016-2017"

}],

"color\_result": "棕色",

"location\_result":

{

"left": 257,

"top": 663,

"width": 1229,

"height": 795

}

}

7.细粒度图像识别—logo商标识别

7.1.接口描述

该请求用于检测和识别图片中的品牌LOGO信息。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片中LOGO的名称、位置和置信度。 当效果欠佳时，可以建立子库（在[控制台](https://console.bce.baidu.com/ai/#/ai/imagerecognition/overview/index)创建应用并申请建库）并通过调用logo入口接口完成自定义logo入库，提高识别效果。

7.2.logo商标识别

7.2.1.请求说明

**请求示例**

HTTP 方法：POST

请求URL： https://aip.baidubce.com/rest/2.0/image-classify/v2/logo

URL参数：

| 参数 | 值 |
| --- | --- |
| access\_token | 通过API Key和Secret Key获取的access\_token,参考“[Access Token获取](http://ai.baidu.com/docs#/Auth)” |

Header如下：

| 参数 | 值 |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数，参数详情如下：

**请求参数**

| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| --- | --- | --- | --- | --- |
| image | 是 | string | - | 图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式. |
| custom\_lib | 否 | boolean | - | 是否只检索用户子库，true则只检索用户子库，false(默认)为检索底库+用户子库 |

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

Java

package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;

import com.baidu.ai.aip.utils.FileUtil;

import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/\*\*

\* logo商标识别

\*/

public class Logo {

/\*\*

\* 重要提示代码中所需工具类

\* FileUtil,Base64Util,HttpUtil,GsonUtils请从

\* https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72

\* https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2

\* https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3

\* https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3

\* 下载

\*/

public static String logo() {

// 请求url

String url = "https://aip.baidubce.com/rest/2.0/image-classify/v2/logo";

try {

// 本地文件路径

String filePath = "[本地文件路径]";

byte[] imgData = FileUtil.readFileByBytes(filePath);

String imgStr = Base64Util.encode(imgData);

String imgParam = URLEncoder.encode(imgStr, "UTF-8");

String param = "image=" + imgParam + "&custom\_lib=" + true;

// 注意这里仅为了简化编码每一次请求都去获取access\_token，线上环境access\_token有过期时间， 客户端可自行缓存，过期后重新获取。

String accessToken = "[调用鉴权接口获取的token]";

String result = HttpUtil.post(url, accessToken, param);

System.out.println(result);

return result;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public static void main(String[] args) {

Logo.logo();

}

}

7.2.2.返回说明

**返回参数**

| 字段 | 是否必选 | 类型 | 说明 |
| --- | --- | --- | --- |
| log\_id | 是 | uint64 | 唯一的log id，用于问题定位 |
| result\_num | 是 | uint32 | 识别结果数，标识返回结果数目 |
| result | 否 | array() | 返回结果数组，每一项为一个识别出的logo |
| +location | 是 | object | 位置信息（左起像素位置、上起像素位置、像素宽、像素高） |
| ++left | 是 | number | 坐起像素位置 |
| ++top | 是 | number | 上起像素位置 |
| ++width | 是 | number | 像素宽 |
| ++height | 是 | number | 像素高 |
| +name | 是 | string | 识别的品牌名称 |
| +probability | 是 | number | 分类结果置信度（0--1.0） |
| +type | 是 | number | type=0为1千种高优商标识别结果;type=1为2万类logo库的结果；其它type为自定义logo库结果 |

**返回示例**

HTTP/1.1 200 OK

x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e

Cache-Control: no-cache

Server: BWS

Date: Tue, 18 Oct 2016 02:21:01 GMT

Content-Type: application/json;charset=UTF-8

{

"log\_id": 843411868,

"result\_num": 1,

"result": [

{

"type": 0,

"name": "科颜氏",

"probability": 0.99998807907104,

"location": {

"width": 296,

"top": 20,

"height": 128,

"left": 23

}

}

]

}

7.3.logo商标识别—添加

使用入库接口请先在[控制台](https://console.bce.baidu.com/ai/#/ai/imagerecognition/overview/index)创建应用并申请建库，建库成功后方可正常使用。

**添加logo图片规范**

1. logo商标入库需要待识别商标的紧致包围盒图像，下面是一组示例： 
2. 每种商标不同场景多入库一些图像有助于获得更好的识别效果。如上图中“美素佳儿”图标。
3. 入库的图像最好是每种商标的最小识别单元。例如入库下面第一个图像的效果不如将下面后三个商标图片入库的效果好。

入库后识别效果一般：



分别入库后效果更好



**不适合入库的图片示例**

1. 尽量不要含有背景，像下面图入库效果肯定不佳，需要裁剪。



1. 不含logo的图像入库，不会返回该图像，因为算法先回做商标检测，如果没有商标，返回未识别结果，如下图。 

**请求示例**

HTTP 方法：POST

请求URL： https://aip.baidubce.com/rest/2.0/realtime\_search/v1/logo/add

URL参数：

| 参数 | 值 |
| --- | --- |
| access\_token | 通过API Key和Secret Key获取的access\_token,参考“[Access Token获取](http://ai.baidu.com/docs#/Auth)” |

Header如下：

| 参数 | 值 |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数，参数详情如下：

**请求参数**

| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| --- | --- | --- | --- | --- |
| image | 是 | string | - | 图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式. |
| brief | 是 | string | - | 检索时带回。此处要传对应的name与code字段，name长度小于100B，code长度小于150B。示例："{"name": "abc", "code":"ddd"}" |

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

bash

PHP

Java

Python

C++

C#

logo商标识别—添加

curl -i -k 'https://aip.baidubce.com/rest/2.0/realtime\_search/v1/logo/add?access\_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】&custom\_lib={'name': 'abc', 'code': 'abc'}' -H 'Content-Type:application/x-www-form-urlencoded'

7.3.1.返回说明

**返回参数**

| 字段 | 是否必选 | 类型 | 说明 |
| --- | --- | --- | --- |
| log\_id | 是 | uint64 | 唯一的log id，用于问题定位 |
| cont\_sign | 是 | string | 输入图片签名，可用于删除 |

**返回示例**

HTTP/1.1 200 OK

x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e

Cache-Control: no-cache

Server: BWS

Date: Tue, 18 Oct 2016 02:21:01 GMT

Content-Type: application/json;charset=UTF-8

{

"log\_id": 972934653,

"cont\_sign": "217113248,419422301"

}

7.4.logo商标识别—删除

使用删除接口请先在[控制台](https://console.bce.baidu.com/ai/#/ai/imagerecognition/overview/index)创建应用并申请建库，建库成功后先调用入库接口完成logo图片入库，删除接口用户在已入库的logo图片中删除图片。

**请求示例**

HTTP 方法：POST

请求URL： https://aip.baidubce.com/rest/2.0/realtime\_search/v1/logo/delete

URL参数：

| 参数 | 值 |
| --- | --- |
| access\_token | 通过API Key和Secret Key获取的access\_token,参考“[Access Token获取](http://ai.baidu.com/docs#/Auth)” |

Header如下：

| 参数 | 值 |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数，参数详情如下：

**请求参数**

| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| --- | --- | --- | --- | --- |
| image | 是 | string | - | base64编码后的图片数据（和cont\_sign二选一）。要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式. |
| cont\_sign | 是 | string | - | 图片签名（和image二选一，image优先级更高） |

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

**提示二**：部分语言依赖的类或库，请在代码注释中查看下载地址。

bash

PHP

Java

Python

C++

C#

logo商标识别—删除

curl -i -k 'https://aip.baidubce.com/rest/2.0/realtime\_search/v1/logo/delete?access\_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'

7.4.1.返回说明

**返回参数**

| 字段 | 是否必选 | 类型 | 说明 |
| --- | --- | --- | --- |
| log\_id | 是 | uint64 | 唯一的log id，用于问题定位 |

**返回示例**

HTTP/1.1 200 OK

x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e

Cache-Control: no-cache

Server: BWS

Date: Tue, 18 Oct 2016 02:21:01 GMT

Content-Type: application/json;charset=UTF-8

{

"log\_id": 50488417

}

8.细粒度图像识别—动物识别

8.1.接口描述

该请求用于识别一张图片，即对于输入的一张图片（可正常解码，且长宽比较合适），输出动物识别结果。

8.2.请求说明

**请求示例**

HTTP 方法：POST

请求URL： https://aip.baidubce.com/rest/2.0/image-classify/v1/animal

URL参数：

| 参数 | 值 |
| --- | --- |
| access\_token | 通过API Key和Secret Key获取的access\_token,参考“[Access Token获取](http://ai.baidu.com/docs#/Auth)” |

Header如下：

| 参数 | 值 |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数，参数详情如下：

**请求参数**

| 参数名称 | 是否必选 | 类型 | 默认值 | 说明 |
| --- | --- | --- | --- | --- |
| image | 是 | string |  | 图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 |
| top\_num | 否 | integer | 6 | 返回预测得分top结果数，默认为6 |

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

Java

package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;

import com.baidu.ai.aip.utils.FileUtil;

import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/\*\*

\* 动物识别

\*/

public class Animal {

/\*\*

\* 重要提示代码中所需工具类

\* FileUtil,Base64Util,HttpUtil,GsonUtils请从

\* https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72

\* https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2

\* https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3

\* https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3

\* 下载

\*/

public static String animal() {

// 请求url

String url = "https://aip.baidubce.com/rest/2.0/image-classify/v1/animal";

try {

// 本地文件路径

String filePath = "[本地文件路径]";

byte[] imgData = FileUtil.readFileByBytes(filePath);

String imgStr = Base64Util.encode(imgData);

String imgParam = URLEncoder.encode(imgStr, "UTF-8");

String param = "image=" + imgParam + "&top\_num=" + 6;

// 注意这里仅为了简化编码每一次请求都去获取access\_token，线上环境access\_token有过期时间， 客户端可自行缓存，过期后重新获取。

String accessToken = "[调用鉴权接口获取的token]";

String result = HttpUtil.post(url, accessToken, param);

System.out.println(result);

return result;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public static void main(String[] args) {

Animal.animal();

}

}

8.3.返回说明

**返回参数**

| 参数 | 类型 | 是否必须 | 说明 |
| --- | --- | --- | --- |
| log\_id | uint64 | 是 | 唯一的log id，用于问题定位 |
| result | arrry(object) | 是 | 识别结果数组 |
| +name | string | 是 | 动物名称，示例：蒙古马 |
| +score | uint32 | 是 | 置信度，示例：0.5321 |

**返回示例**

{

"log\_id": 304829369,

"result": [

{

"name": "蒙古马",

"score": 0.41471540927887

},

{

"name": "伊犁马",

"score": 0.39988899230957

},

{

"name": "哈萨克马",

"score": 0.39988899230957

}

...

]

}

9.细粒度图像识别—植物识别

9.1.接口描述

该请求用于识别一张图片，即对于输入的一张图片（可正常解码，且长宽比较合适），输出植物识别结果。

9.2.请求说明

**请求示例**

HTTP 方法：POST

请求URL： https://aip.baidubce.com/rest/2.0/image-classify/v1/plant

URL参数：

| 参数 | 值 |
| --- | --- |
| access\_token | 通过API Key和Secret Key获取的access\_token,参考“[Access Token获取](http://ai.baidu.com/docs#/Auth)” |

Header如下：

| 参数 | 值 |
| --- | --- |
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数，参数详情如下：

**请求参数**

| 参数名称 | 是否必选 | 类型 | 说明 |
| --- | --- | --- | --- |
| image | 是 | string | 图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 |

**请求代码示例**

**提示一**：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

Java

package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;

import com.baidu.ai.aip.utils.FileUtil;

import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/\*\*

\* 植物识别

\*/

public class Plant {

/\*\*

\* 重要提示代码中所需工具类

\* FileUtil,Base64Util,HttpUtil,GsonUtils请从

\* https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72

\* https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2

\* https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3

\* https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3

\* 下载

\*/

public static String plant() {

// 请求url

String url = "https://aip.baidubce.com/rest/2.0/image-classify/v1/plant";

try {

// 本地文件路径

String filePath = "[本地文件路径]";

byte[] imgData = FileUtil.readFileByBytes(filePath);

String imgStr = Base64Util.encode(imgData);

String imgParam = URLEncoder.encode(imgStr, "UTF-8");

String param = "image=" + imgParam;

// 注意这里仅为了简化编码每一次请求都去获取access\_token，线上环境access\_token有过期时间， 客户端可自行缓存，过期后重新获取。

String accessToken = "[调用鉴权接口获取的token]";

String result = HttpUtil.post(url, accessToken, param);

System.out.println(result);

return result;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public static void main(String[] args) {

Plant.plant();

}

}

9.3.返回说明

**返回参数**

| 参数 | 类型 | 是否必须 | 说明 |
| --- | --- | --- | --- |
| log\_id | uint64 | 是 | 唯一的log id，用于问题定位 |
| result | arrry(object) | 是 | 植物识别结果数组 |
| +name | string | 是 | 植物名称，示例：吉娃莲 |
| +score | uint32 | 是 | 置信度，示例：0.5321 |

**返回示例**

{

"log\_id": 304829369,

"result": [

{

"name": "吉娃莲",

"score": 0.41471540927887

},

{

"name": "番杏科",

"score": 0.39988899230957

},

{

"name": "花月夜",

"score": 0.39988899230957

}

...

]

}

10.EasyDL定制化图像识别

可视化操作界面自动化训练图像识别模型，支持定制图像分类和物体检测两类图像识别模型。 详情访问：[定制化图像训练平台](http://ai.baidu.com/customize/common/welcome)进行训练。

**图像分类** 识别一张图中是否是某类物体/状态/场景。可以识别图片中主体单一的场景。 [应用场景] 1、图片内容检索，可定制识别各类物体 2、UGC内容审核 3、工业分拣或质检

**物体检测**： 在一张图包含多个物体的情况下，定制识别出每个物体的位置、数量、名称。可以识别图片中有多个主体的场景。 [应用场景] 1、物体计数，如零售行业货品摆放统计、医疗细胞计数 2、一图多物识别，识别一张图中多个物体

10.1.定制图像分类

10.1.1.接口描述

基于自定义训练出的图像分类模型，实现个性化图像识别。 详情访问：[定制化图像训练平台](http://ai.baidu.com/customize/common/welcome)进行训练。 更多训练模型过程中的常见问题查看 [http://ai.baidu.com/docs#/FAQ/504affa3](http://ai.baidu.com/customize/common/welcome) 如有其它问题，欢迎加入QQ群185289236进行咨询。

10.1.2.请求说明

**请求示例**

HTTP 方法：POST

请求URL： 请首先在[定制化图像训练平台](http://ai.baidu.com/customize/common/welcome)进行自定义模型训练，完成训练后可在服务列表中查看并获取url。

URL参数：

| 参数 | 值 |
| --- | --- |
| access\_token | 通过API Key和Secret Key获取的access\_token,参考“[Access Token获取](http://ai.baidu.com/docs#/Auth)” |

Header如下：

| 参数 | 值 |
| --- | --- |
| Content-Type | application/json |

**注意**：如果出现336001的错误码很可能是因为请求方式错误，与其他图像识别服务不同的是定制化图像识别服务以json方式请求。

Body请求示例：

{

"image": "<base64数据>",

"top\_num": 5

}

Body中放置请求参数，参数详情如下：

**请求参数**

| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| --- | --- | --- | --- | --- |
| image | 是 | string | - | 图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注意请去掉头部** |
| top\_num | 否 | number | - | 返回分类数量，默认为6个 |

10.1.3.返回说明

**返回参数**

| 字段 | 是否必选 | 类型 | 说明 |
| --- | --- | --- | --- |
| log\_id | 是 | number | 唯一的log id，用于问题定位 |
| results | 否 | array(object) | 分类结果数组 |
| +name | 否 | string | 分类名称 |
| +score | 否 | number | 置信度 |

10.2.定制物体检测

10.2.1.接口描述

基于自定义训练出的物体检测模型，实现个性化图像识别。 详情访问：[定制化图像训练平台](http://ai.baidu.com/customize/common/welcome)进行训练。 更多训练模型过程中的常见问题查看 [http://ai.baidu.com/docs#/FAQ/504affa3](http://ai.baidu.com/customize/common/welcome) 如有其它问题，欢迎加入QQ群185289236进行咨询。

10.2.2.请求说明

**请求示例**

HTTP 方法：POST

请求URL： 请首先在[定制化图像训练平台](http://ai.baidu.com/customize/common/welcome)进行自定义模型训练，完成训练后申请上线，上线成功后可在服务列表中查看并获取url。

URL参数：

| 参数 | 值 |
| --- | --- |
| access\_token | 通过API Key和Secret Key获取的access\_token,参考“[Access Token获取](http://ai.baidu.com/docs#/Auth)” |

Header如下：

| 参数 | 值 |
| --- | --- |
| Content-Type | application/json |

**注意**：如果出现336001的错误码很可能是因为请求方式错误，与其他图像识别服务不同的是定制化图像识别服务以json方式请求。

Body请求示例：

{

"image": "<base64数据>",

}

Body中放置请求参数，参数详情如下：

**请求参数**

| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |
| --- | --- | --- | --- | --- |
| image | true | string | - | 图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注意请去掉头部** |
| threshold | false | number | - | 默认值为0.3，请在我的模型列表-模型效果查看推荐阈值 |

10.2.3.返回说明

**返回参数**

| 字段 | 是否必选 | 类型 | 说明 |
| --- | --- | --- | --- |
| log\_id | 是 | number | 唯一的log id，用于问题定位 |
| results | 否 | array(object) | 识别结果数组 |
| +name | 否 | string | 分类名称 |
| +score | 否 | number | 置信度 |
| +location | 否 |  |  |
| ++left | 否 | number | 检测到的目标主体区域到图片左边界的距离 |
| ++top | 否 | number | 检测到的目标主体区域到图片上边界的距离 |
| ++width | 否 | number | 检测到的目标主体区域的宽度 |
| ++height | 否 | number | 检测到的目标主体区域的高度 |

11.错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

* \*\*error\_code：\*\*错误码。
* \*\*error\_msg：\*\*错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

{

"error\_code": 110,

"error\_msg": "Access token invalid or no longer valid"

}

需要重新获取新的Access Token再次请求即可。

| 错误码 | 错误信息 | 描述 |
| --- | --- | --- |
| 1 | Unknown error | 服务器内部错误，请再次请求， 如果持续出现此类错误，请通过QQ群（649285136）或工单联系技术支持团队。 |
| 2 | Service temporarily unavailable | 服务暂不可用，请再次请求， 如果持续出现此类错误，请通过QQ群（649285136）或工单联系技术支持团队。 |
| 3 | Unsupported openapi method | 调用的API不存在，请检查后重新尝试 |
| 4 | Open api request limit reached | 集群超限额 |
| 6 | No permission to access data | 无权限访问该用户数据 |
| 17 | Open api daily request limit reached | 每天请求量超限额 |
| 18 | Open api qps request limit reached | QPS超限额 |
| 19 | Open api total request limit reached | 请求总量超限额 |
| 100 | Invalid parameter | 无效的access\_token参数，请检查后重新尝试 |
| 110 | Access token invalid or no longer valid | access\_token无效 |
| 111 | Access token expired | access token过期 |
| 282000 | internal error | 服务器内部错误，请再次请求， 如果持续出现此类错误，请通过QQ群（649285136）或工单联系技术支持团队。 |
| 216100 | invalid param | 请求中包含非法参数，请检查后重新尝试 |
| 216101 | not enough param | 缺少必须的参数，请检查参数是否有遗漏 |
| 216102 | service not support | 请求了不支持的服务，请检查调用的url |
| 216103 | param too long | 请求中某些参数过长，请检查后重新尝试 |
| 216110 | appid not exist | appid不存在，请重新核对信息是否为后台应用列表中的appid |
| 216200 | empty image | 图片为空，请检查后重新尝试 |
| 216201 | image format error | 上传的图片格式错误，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片 |
| 216202 | image size error | 上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096\*4096，请重新上传图片 |
| 216203 | image size error | 上传的图片base64编码有误，请校验base64编码方式，并重新上传图片 |
| 216630 | recognize error | 识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（649285136）或工单联系技术支持团队。 |
| 216631 | recognize bank card error | 识别银行卡错误，出现此问题的原因一般为：您上传的图片非银行卡正面，上传了异形卡的图片或上传的银行卡正品图片不完整 |
| 216633 | recognize idcard error | 识别身份证错误，出现此问题的原因一般为：您上传了非身份证图片或您上传的身份证图片不完整 |
| 216634 | detect error | 检测错误，请再次请求，如果持续出现此类错误，请通过QQ群（649285136）或工单联系技术支持团队。 |
| 282003 | missing parameters: {参数名} | 请求参数缺失 |
| 282005 | batch  processing error | 处理批量任务时发生部分或全部错误，请根据具体错误码排查 |
| 282006 | batch task  limit reached | 批量任务处理数量超出限制，请将任务数量减少到10或10以下 |
| 282114 | url size error | URL长度超过1024字节或为0 |
| 282808 | request id: xxxxx not exist | request id xxxxx 不存在 |
| 282809 | result type error | 返回结果请求错误（不属于excel或json） |
| 282810 | image recognize error | 图像识别错误 |
| 283300 | Invalid argument | 入参格式有误，可检查下图片编码、代码格式是否有误 |
| 336000 | Internal error | 服务器内部错误，请再次请求， 如果持续出现此类错误，请通过QQ群（185289236）或工单联系技术支持团队 |
| 336001 | Invalid Argument | 入参格式有误，比如缺少必要参数、图片base64编码错误等等，可检查下图片编码、代码格式是否有误。有疑问请通过QQ群（185289236）或工单联系技术支持团队 |