简介

Hi，您好，欢迎使用译图智讯文字识别OCR服务。 本文档主要针对API开发者，描述译图智讯OCR文字识别接口服务的相关技术内容。

接口能力

产品名称	接口调用对应的pid
人证识别	
身份证OCR识别	2
护照OCR识别	8
结婚证OCR识别	12
银行卡OCR识别	4
军官证OCR识别	11
外国人永久居留证OCR识别	53
出生证明OCR识别	52
临时身份证OCR识别	1020

产品名称	接口调用对应的pid
港澳台居民来往大陆通行证	1018
户口本户主页OCR识别	1002
户口本个人页OCR识别	1003
社保卡OCR识别	1039
往来港澳通行证正面OCR识别	1013
离婚证OCR识别	1005
往来港澳通行证背面OCR识别	1014
车辆证件	
VIN码OCR识别	1
行驶证OCR识别	5
驾驶证OCR识别	7
机动车发票OCR识别	61
车辆登记证OCR识别	62
车辆合格证OCR识别	68
车牌OCR识别	6
二手车发票OCR识别	60
道理运输许可证OCR识别	1076
电子保单OCR识别	1038
发票识别	
增值税发票OCR识别	21
火车票OCR识别	22
出租车票OCR识别	22
行程单OCR识别	22
定额发票OCR识别	22
电子承兑汇票OCR识别	1031
银行回单OCR识别	1037
房产证识别	
不动产OCR识别	501
不动产登记证OCR识别	1007
房产证OCR识别	1033
房产证（统一字段）OCR识别	1082
企业证件	

营业执照OCR识别	41
开户许可证OCR识别	1015
组织机构代码证OCR识别	1026
通用识别	
通用票据OCR识别	22
通用文字OCR识别	1001
通用表格OCR识别	1036
手写体OCR识别	1034
银行流水OCR识别	1066
证照自动分类OCR识别	666
证照自动分类	667

调用方式

请求格式

POST方式调用

接口地址

http://www.etoplive.com/ocr/PageOcrServlet

支持http和https两种协议调用

请求头

Header参数名称	Value
Content-Type	application/x-www-form-urlencoded

请求参数

参数名称	字段类型	字段描述	是否必选
file	String	图片文件的 BASE64 编码	否
urlPath	String	图片的url下载地址 (file 和urlPath参数必须 二个选一。)	否

参数名称	字段类型	字段描述	是否必选
pid	String	请查看简介里产品对应的pid编号	是
key	String	用户的Key 点击登录后查看	是
secret	String	用户的secret	是
requestId	String	调用方业务流水号(长度限制为255)	否
removeImgData	String	移除返回的裁切图base64值,默认值为1,设置为0才会返回裁切图的base64值	否
openSeal	String	识别印章内信息.默认值为0,设置为1开启识别(开启后识别时间会增加)	否
showPosition	String	是否返回字段位置信息.默认值为0,设置为1返回，用于通用票据	否
imageCheck	String	图像质量检测模块开启功能.此功能只针对二代身份证，现在五个维度，一位表示一个维度，现在检测维度包含：图像类型(原件、其他件(复印件、屏拍件)的区分)，图像完整性判断，光斑检测，模糊判断，PS检测，默认值为"00000"，0表示不开启，1表示开启	否
imageCheckVaule	String	是否对各维度分数达标进行判断，只有在启动ImageCheck功能时才有效，默认值为1，不判断，直接返回分值，设置值为0，如果分数不达标，直接返回错误码ErrorCode	否

注意事项

- 1.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。

- 2.base64编码urlencode后大小不超过5M
- 3.图片的base64编码是不包含图片头的，如（）

调用示例

Java

```
package com.ocrweb.test;

import com.sun.org.apache.xml.internal.security.utils.Base64;

import okhttp3.FormBody;

import okhttp3.OkHttpClient;

import okhttp3.Request;

import okhttp3.Response;

import java.io.*;

import java.util.concurrent.TimeUnit;

public class OcrTest2 {

        /**

    *需要导入okhttp依赖

    *<dependency>

    *  <groupId>com.squareup.okio</groupId>

    *  <artifactId>okio</artifactId>

    *  <version>1.13.0</version>

    *</dependency>

    *<dependency>

    *  <groupId>com.squareup.okhttp3</groupId>

    *  <artifactId>okhttp</artifactId>

    *  <version>3.12.0</version>

    *</dependency>

    */

       public static void main(String[] args) {

           String strFilep = "D:\\LinuxTestPic\\2身份证\\2.jpg"; //图片本地存储的路径

           String pid = "2"; //产品pid

           String key = "";        //用户的key,登录之后能够查看到

           String secret = ""; //用户的secret

           String url = "http://etoplive.com/ocr/PageOcrServlet";

           File file = new File(strFilep);

           try (FileInputStream is = new FileInputStream(file)) { byte[] data = new byte[is.available()]; is.read(data);
```

```
                String filedata = com.sun.org.apache.xml.internal.security.utils.Base64.encode(data);
                OkHttpClient client = new OkHttpClient.Builder()
                        .connectTimeout(15, TimeUnit.SECONDS)// 连接超时时间
                        .writeTimeout(30, TimeUnit.SECONDS)// 写入超时时间
                        .readTimeout(60, TimeUnit.SECONDS)// 读取超时时间
                        .build();
                FormBody mBody = new FormBody.Builder()
                        .add("pid", pid)
                        .add("file", filedata)
                        .add("key", key)
                        .add("secret", secret)
                        .build();
                Request request = new Request.Builder().url(url).post(mBody).build();
                Response response = client.newCall(request).execute();
                String result = response.body().string();
                System.out.println(result);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
}
```

python

```
import base64
import requests

with open("D:\\LinuxTestPic\\2身份证\\2.jpg", 'rb') as f:
    base64_data = base64.b64encode(f.read())
    #获取解码后的base64值
    filedata = base64_data.decode()
    #pid 产品的pid,key,secret 登录之后能够查看到
    datas = {"pid": "2", "key": "","secret":"","file":filedata}
    r = requests.post("http://www.etoplive.com/ocr/PageOcrServlet", data=datas)
    if r.status_code ==200 :
         #识别结果
         print(r.text)
```

go

```
package main

import (
         "encoding/base64"
         "fmt"
         "io/ioutil"
         "net/http"
         "strings"
)

func main()  {
         img, err := ioutil.ReadFile("D:\\LinuxTestPic\\人证识别\\2身份证\\7.jpg")
         if err != nil {
                  panic(err)
         }
         base64Str := base64.StdEncoding.EncodeToString(img)

         var r http.Request
         r.ParseForm()
         //证件类型id
         r.Form.Add("pid", "2")
         //用户的key(用户登录能看到的凭证信息)
         r.Form.Add("key", "")
         //用户的secret
         r.Form.Add("secret", "")
         r.Form.Add("file", base64Str)
         res:=Post("http://www.etoplive.com/ocr/PageOcrServlet",r)
         fmt.Println(res)
}

func Post(url string, req http.Request) string {
         bodystr := strings.TrimSpace(req.Form.Encode())
         request, err := http.NewRequest("POST", url, strings.NewReader(bodystr))
         if err != nil {
                  panic(err)
         }
         request.Header.Set("Content-Type", "application/x-www-form-urlencoded")
         var resp *http.Response
         resp, err = http.DefaultClient.Do(request)
         result, _ := ioutil.ReadAll(resp.Body)
         return string(result)
}
```

node.js

```
const fs = require('fs');
const http = require('http');
const querystring = require('querystring');

//读取文件
let bitmap = fs.readFileSync('D:\\LinuxTestPic\\2身份证\\2.jpg');
// 获取base64编码
let basedata = Buffer.from(bitmap, 'binary').toString('base64');

var data = {
    pid: 2, //证件类型
    key: '',  //认证的key (登录后能看到)
    secret:'', //认证用的secret
    file:basedata  //图片的base值
};
var content = querystring.stringify(data);

var options = {
    hostname: 'www.etoplive.com',
    path: '/ocr/PageOcrServlet',
    method: 'POST',
    headers: {
        'Content-Type': 'application/x-www-form-urlencoded; charset=UTF-8'
    }
};

var req = http.request(options, function (res) {
    console.log('STATUS: ' + res.statusCode);
    res.setEncoding('utf8');
    res.on('data', function (chunk) {
        console.log('识别结果: ' + chunk);
    });
});
req.on('error', function (e) {
    console.log('problem with request: ' + e.message);
});
// 发送请求参数
req.write(content);
req.end();
```

C#

```
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Drawing;
namespace ConsoleApp2
{
    class Program
    {

        static void Main(string[] args)
        {
            String basedata;
            MemoryStream ms = new MemoryStream();
            try
            {
                Bitmap bmp = new Bitmap("D:\\LinuxTestPic\\2身份证\\2.jpg");
                bmp.Save(ms, System.Drawing.Imaging.ImageFormat.Jpeg);
                byte[] arr = new byte[ms.Length];
                ms.Position = 0;
                ms.Read(arr, 0, (int)ms.Length);
                basedata = Convert.ToBase64String(arr);
```

```
            }
            finally
            {
                ms.Close();
            }
            String pid = "2"; //证件类型id
            String key = ""; //用户的key(用户登录能看到的凭证信息)
            String secret = ""; //用户的secret
            string postData = string.Format("pid={0}&key={1}&secret={2}&file={3}", pid, key, secret, basedata);
            byte[] data = Encoding.UTF8.GetBytes(postData);

            HttpWebRequest request = (HttpWebRequest)WebRequest.Create("http://www.etoplive.com/ocr/PageOcrServlet");
            request.Method = "POST";
            request.ContentLength = data.Length;
            request.ContentType = "application/x-www-form-urlencoded";
            Stream myRequestStream = request.GetRequestStream();
            myRequestStream.Write(data, 0, data.Length);
            myRequestStream.Close();

            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            //读取识别结果
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.UTF8);
            string content = reader.ReadToEnd();
            Console.WriteLine(content);
            Console.Read();

        }
    }
}
```

php

```php
<?php
//取消PHP的内存限制
ini_set('memory_limit', '-1');
ini_set('user_agent', 'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)');

/**
 * 文件转成base64
 */
function base64EncodeFile($file)
{
    $base64_file = '';
    if (file_exists($file)) {
        $base64_file = base64_encode(file_get_contents($file));
    }
    return $base64_file;
}

/**
 * 发送post请求
 */
function send_post($url, $data)
{
    $postdata = http_build_query($data);
    $options = array(
        'http' => array(
            'method' => 'POST',
            'header'  => 'Content-type: application/x-www-form-urlencoded',
            'timeout' => 15 * 60 ,// 超时时间（单位:s）
            'content' => $postdata
        )
    );
    $context = stream_context_create($options);
    $result = file_get_contents($url, false, $context);
    return $result;

}

$filedata = base64EncodeFile("D:\\2身份证\\2.jpg");

$post_data = array(
    'pid' => '2', //产品pid
    'key' => '', //用户的key,登录之后能够查看到
    'secret' => '', //用户的secret
    'file' => $filedata
);
$res = send_post("http://www.etoplive.com/ocr/PageOcrServlet", $post_data);
echo $res;
```

OC

```
 //MARK:pid
    NSString * pidString = [NSString stringWithFormat:@"pid=%zd", pid];
    NSMutableData *postData = [[NSMutableData alloc] initWithData:[pidString dataUsingEncoding:NSUTF8StringEncoding]];

    //MARK:file
    NSData * imageData = UIImageJPEGRepresentation(image, 0.9);

    //图片->base64
    NSString *image64 = [imageData base64EncodedStringWithOptions:NSDataBase64Encoding64CharacterLineLength];

    image64 = [image64 stringByReplacingOccurrencesOfString:@"\r\n" withString:@""];
    NSString *newString = CFBridgingRelease(CFURLCreateStringByAddingPercentEscapes(kCFAllocatorDefault,
(CFStringRef)image64, nil, CFSTR(":/?#[]@!$&' ()*+,;="), kCFStringEncodingUTF8));
    NSString * fileData = [NSString stringWithFormat:@"&file=%@", newString];

    [postData appendData:[fileData dataUsingEncoding:NSUTF8StringEncoding]];

    //MARK:KEY
    NSString * ocrKey = [NSString stringWithFormat:@"&key=%@", SERVICEKEY];
    [postData appendData:[ocrKey dataUsingEncoding:NSUTF8StringEncoding]];

    //MARK:SECRET
    NSString * ocrSecret = [NSString stringWithFormat:@"&secret=%@", SERVICESECRET];
    [postData appendData:[ocrSecret dataUsingEncoding:NSUTF8StringEncoding]];

    NSString * SERVICEURL = @"http://www.etoplive.com/ocr/PageOcrServlet";

    NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL:[NSURL URLWithString:SERVICEURL]
cachePolicy:NSURLRequestUseProtocolCachePolicy timeoutInterval:15.0];
    [request setHTTPMethod:@"POST"];
    //    [request setAllHTTPHeaderFields:headers];
    [request setHTTPBody:postData];

    NSURLSession *session = [NSURLSession sharedSession];
    NSURLSessionDataTask *dataTask = [session dataTaskWithRequest:request completionHandler:^(NSData *data, NSURLResponse
*response, NSError *error) {
        if (error) {
            NSLog(@"error:%@", error);

        } else {

            NSString * strJson = [[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding];
            NSLog(@"json%@", strJson);


        }
    }];
    [dataTask resume];
```

Swift

```
            //MARK:pid
            let pid = 4
            let pidString: String = String(format: "pid=%zd", pid)
            var postData : Data = pidString.data(using: .utf8)!


            //MARK:data
            let imageData: Data = image.jpegData(compressionQuality: 0.9)!
            //图片->base64
            var image64 :String = imageData.base64EncodedString(options: .lineLength64Characters)

            image64 = image64.replacingOccurrences(of: "\r\n", with: "")


//                NSString *newString = CFBridgingRelease(CFURLCreateStringByAddingPercentEscapes(kCFAllocatorDefault,
(CFStringRef)image64, nil, CFSTR(":/?#[]@!$&' ()*+,;="), kCFStringEncodingUTF8));
//                let newString: String = CFBridgingRelease(CFURLCreateStringByAddingPercentEscapes(kCFAllocatorDefault,
(CFString)image64, nil, CFSTR(":/?#[]@!$&' ()*+,;="), kCFStringEncodingUTF8));
            let fileString: String = String(format: "&file=%@", image64)
            //file
            let fileData: Data = fileString.data(using: .utf8)!
            postData.append(fileData)

            //MARK:key
            let SERVICEKEY = ""
            let ocrKey: String = String(format: "&key=%@", SERVICEKEY)
            let keyData: Data = ocrKey.data(using: .utf8)!
            postData.append(keyData)


            //MARK:secret
            let SERVICESECRET = ""
            let ocrSecret: String = String(format: "&secret=%@", SERVICESECRET)
            let secretData: Data = ocrSecret.data(using: .utf8)!
            postData.append(secretData)

            let url: URL = URL.init(string: "http://www.etoplive.com/ocr/PageOcrServlet")!


            var request: URLRequest = URLRequest(url: url, cachePolicy: .useProtocolCachePolicy, timeoutInterval: 15.0)
            request.httpMethod = "POST"

            request.httpBody = postData

            let session: URLSession = URLSession.shared
            let dataTask: URLSessionTask = session.dataTask(with: request, completionHandler: { (data, response, error) in
                if (error) != nil {
                    print(error!)

                } else {
                    print(data!)
                    let strJson: String = String.init(data: data!, encoding: .utf8) ?? ""

                    print(strJson)



                }
            })

            dataTask.resume()
```

错误码

返回示例

```
{
    "ErrorCode": "0"
}
```

若服务器返回的ErrorCode等于0,则请求成功,其它则请求失败!

错误编码	描述
0	识别成功
3	非该类型证件
16	不支持的图像文件
17	未定义的产品类型
19	识别失败
-1	用户已屏蔽
-2	用户key或secret验证错误。
-3	服务次数不足。
-4	用户未找到。
-5	未支持的识别类型。
-6	file参数未提供。
-7	未提供keyh或secret参数
-8	服务器IO错误.
-49	请求超时
-51	图片错误, 请换一张再重试
-52	系统繁忙, 请一会后访问
-55	客户端断开连接

返回数据样例

请参考官网最新在线文档： http://www.etoplive.com/apidoc.do