



# 天翼云印刷文字识别产品

用户操作手册

天翼云科技有限公司

# 目 录

---

<b>1 产品简介</b> .....	<b>1</b>
1.1 产品定义 .....	1
1.2 功能特性 .....	1
1.2.1 通用型 OCR: .....	1
1.2.2 车牌识别 .....	1
1.2.3 身份证识别 .....	1
1.2.4 护照识别 .....	1
1.2.5 营业执照识别 .....	2
1.2.6 事业单位法人证书识别 .....	2
1.2.6 通信行程卡识别 .....	2
1.3 产品优势 .....	2
1.4 应用场景 .....	3
<b>2 快速入门</b> .....	<b>3</b>
2.1 创建印刷文字识别应用 .....	4
2.1.1 印刷文字识别产品订购入口 .....	4
2.1.2 创建印刷文字识别应用订单 .....	5
2.1.3 订单确认 .....	6
2.2 印刷文字识别用户控制台 .....	8
<b>3 用户指南</b> .....	<b>10</b>
3.1 产品价格 .....	10
3.2 产品购买 .....	11

---

3.3 产品续订 .....	13
3.4 产品退订 .....	13
<b>4 API 参考 .....</b>	<b>14</b>
4.1 API 概览 .....	14
4.1.1 概述 .....	14
本说明提供了印刷文字识别产品 API 的描述、语法、参数说明及示例等内容。 .....	14
4.1.2 API 概览 .....	14
4.1.3 状态码 .....	14
4.2 如何调用 API .....	16
4.1.1 终端节点 .....	16
4.1.2 构造请求 .....	16
4.1.3 认证鉴权 .....	17
4.1.4 Python 调用示例 .....	20
4.1.5 Java 调用示例 .....	23
4.3 API .....	24
4.3.1 通用型 OCR .....	24
4.3.2 身份证识别 .....	32
4.3.3 车牌识别 .....	40
4.3.4 护照识别 .....	49
4.3.5 营业执照识别 .....	60
4.3.6 事业单位法人证书识别 .....	67

---

4.3.7 通信行程卡识别.....	75
4.3 更新历史.....	81
<b>5 常见问题.....</b>	<b>81</b>
5.1 计费类.....	81
5.2 购买类.....	82
5.3 操作类.....	82
5.4 使用限制.....	84

---

# 1 产品简介

---

## 1.1 产品定义

印刷文字识别是中国电信云公司自研 AI 平台提供的产品之一，印刷文字识别（OCR）通俗来说是将图片上的文字内容提取出来并进行识别，最终转换为可编辑文本的功能。

天翼云依据客户的使用场景和需求，将产品分为了卡证类识别、汽车相关类识别、行业票据识别、资产类证件识别、行业文档识别、通用文字识别、自定义模版、教育类识别、仪器仪表类识别、票据混贴识别以及多卡证智能分类识别 11 大类，满足各种客户的识别需求。

## 1.2 功能特性

### 1.2.1 通用型 OCR：

通用型 OCR 是适用于多场景、多颜色的 OCR 文字检测和识别服务，识别准确率高，识别速度快。识别支持表格、文档、网络图片等任意格式图片上文字信息的自动化识别，自适应分析各种版面和表格，快速实现各种文档电子化。

### 1.2.2 车牌识别

针对图片中的车牌，进行 OCR 检测，返回检测到的车牌内容及车牌位置坐标。

### 1.2.3 身份证识别

针对图片中的身份证，进行 OCR 检测及后处理，支持返回身份证姓名、住址、身份证号等信息。

### 1.2.4 护照识别

识别护照图片的内容，并结构化输出姓名、性别、护照号码等字段。

### 1.2.5 营业执照识别

识别营业执照图片的内容，并结构化输出统一社会信用代码、名称、住所等字段。

### 1.2.6 事业单位法人证书识别

识别事业单位法人证书图片的内容，并结构化输出统一社会信用代码、名称、住所等字段。

### 1.2.6 通信行程卡识别

针对图片中的通信行程卡，进行 OCR 检测，返回检测到的行程卡颜色，电话号码，请求时间，到达城市，风险城市。接口只允许单张图片请求，不允许图片 list

## 1.3 产品优势

天翼云印刷文字识别产品具有识别精度高，使用场景丰富，服务质量稳定等优势

#### 识别精度高

海量标注样本进行训练，配合国内领先的人工智能算法，针对业务场景优化，打造高精度的印刷文字识别产品。

#### 数据安全

少人工干预，降低隐私风险，保护数据资产。

#### 快速高效

24h 服务无疲劳，识别速度高于人工 10 倍以上。

#### 使用便捷

提供标准 API 接口调用简单，便于被集成。

#### 使用场景丰富

依据客户需求，提供 8 大类别的使用场景，采用深度学习算法，

#### 服务质量稳定

依托天翼云平台，提供稳定稳定高可用的服务品质。

## 部署方式灵活

可以提供公共云、私有云、端服务等部署方式，满足客户部署需求。

## 1.4 应用场景

印刷文字识别主要用于办公场景，媒体场景，政务场景，交通场景，防疫场景

### 各类产品的应用场景

通用 OCR 识别应用场景：

文件存档的录入，客户通用文档文字识别，内容审核监管，搭配客户系统实现疫情健康码和行程卡上的文字识别。

车牌识别应用场景：

停车场，高速公路，城市道路

卡证识别应用场景：

防疫应用识别身份证，行程卡。注册登记审核，企业验证系统信息采集，比如互联网 APP、企业服务都需要进行证件审核。

### 各类场景中的可以实现的应用功能

办公场景：文件存档截图的录入，书籍照片的录入，客户通用文档文字识别，

媒体场景：结合图片进行内容审核监管

政务场景：政务云上各类卡证照的识别

交通场景：中国大陆车牌的识别

防疫场景：搭配客户系统实现疫情防控工作中的身份证识别等

# 2 快速入门

---

## 2.1 创建印刷文字识别应用

### 2.1.1 印刷文字识别产品订购入口

首先，用户使用天翼云网门户账号登录天翼云网门户后，点击菜单栏“产品”项，进入【人工智能】菜单，即可看到印刷文字识别产品相应服务。以通用型 OCR 为例，点击“通用性 OCR”名称即可进入“通用性 OCR”产品详情页。



#### 产品优势



点击【立即使用】按钮，进入 OCR 创建页面，点击“新的应用”。







## 我的能力

人脸识别	安全生产	图像识别	印刷文字识别	自然语言处理	内容安全
人脸属性识别					
API	状态	调用量限制	购买次数包	领取免费额度	
人脸属性识别	待付费	—	购买	立即领取	
人脸检测					
API	状态	调用量限制	购买次数包	领取免费额度	
人脸检测	待付费	—	购买	立即领取	
是否戴口罩识别					
API	状态	调用量限制	购买次数包	领取免费额度	
是否戴口罩识别	待付费	—	购买	立即领取	

## 2.1.3 订单确认

确认订单即可购买并开通服务

在“通用型 OCR” 列点击“购买”，进入产品规格及数量配置页面。

通用型 OCR

选择配置

规格	公测截止时间	单价	购买数量
1000次	2020-11-30 23:59:59	¥0.00	1
10000次	2020-11-30 23:59:59	¥0.00	0
10万次	2020-11-30 23:59:59	¥0.00	1
100万次	2020-11-30 23:59:59	¥0.00	0
1000万次	2020-11-30 23:59:59	¥0.00	0

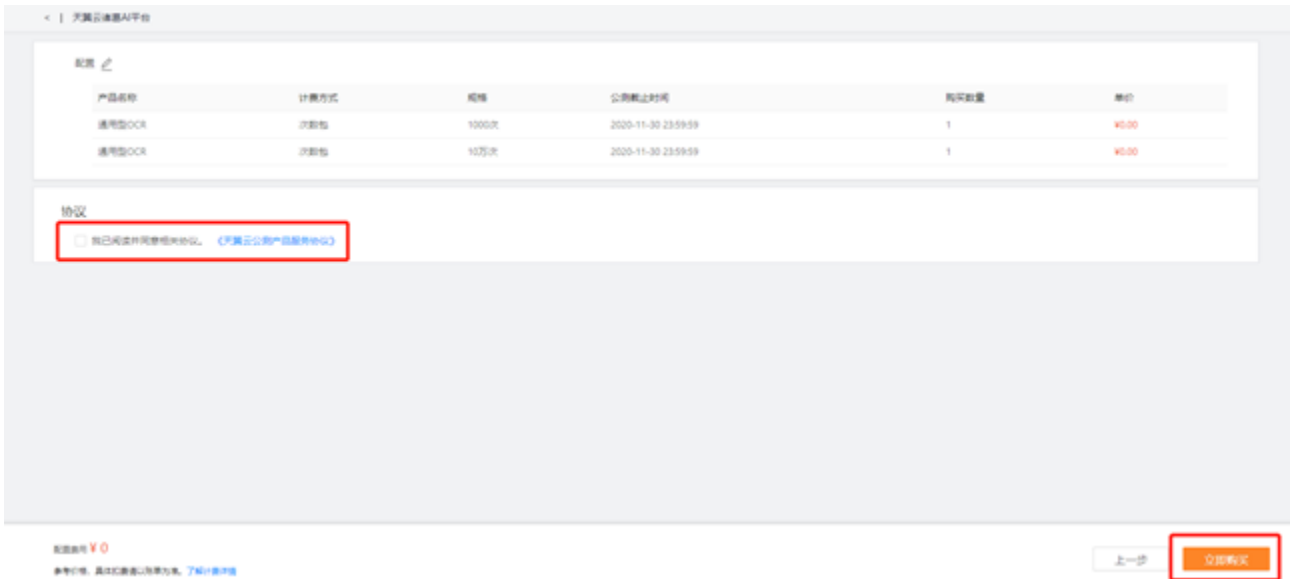
配置清单

规格	购买数量	价格
1000次	1	0
10万次	1	0

费用合计  
**¥0元**

下一步: 确认配置

用户根据需要选择要开通的资源包规格及数量，点击“下一步：确认配置”，进入配置确认界面。



用户勾选服务协议，点击“立即购买”，进入订单支付页面



用户确认订单费用是否准确，费用无误点击“立即支付”。完成支付后进入“支付结果”界面。



用户确认订单费用是否准确，费用无误点击“立即支付”。完成支付后进入“支付结果”界面。



## 2.2 印刷文字识别用户控制台

人工智能类产品采用统一的控制台，地址为 <https://ai.ctyun.cn/subscriber/dashboard>

点击页面右上角“控制台”，进入控制中心。



用户点击“我的应用”，可以查看已经开通的应用。



用户点击“应用监控”，可以查看所创建应用请求次数、响应时间与请求流量。



用户点击“应用告警”，可以了解应用使用过程中系统所反馈的异常情况。



## 3 用户指南

### 3.1 产品价格

印刷文字识别产品采用封顶资源包的方式订购。

印刷文字识别	规格（次）	标准价格（元）	有效期
通用 OCR 识别	10 万	450	一年
	100 万	3200	一年
	500 万	11500	一年
	1000 万	22000	一年
身份证识别	10 万	1350	一年
车牌识别	100 万	9600	一年
护照识别			
营业执照识别	500 万	39000	一年
事业单位法人证书识别	1000 万	78000	一年
通信行程卡识别			

注：护照识别，营业执照识别，事业单位法人证书识别 暂时为限时免费状态，转商后价格如上表。

备注：产品资源包价格以此表格为准，价格调整会在此处更新

订购说明：

1、为满足客户不同业务使用量需求，每类 API 设置五档套餐，如：用户评估认为应用每年身份证识别的 API 调用量大概为 10 万次，则可以购买一个 10 万次/年的 API 调用次数包，若大概为 50 万次，则可以购买五个 10 万次/年的 API 调用次数包

2、用户预先购买 N 个固定额度的资源包，后续使用过程中产生的接口调用次数从资源包中抵扣，有效期内未使用的资源不会流转至下一年。有效期内资源包中的次数用完之后，用户无法再调用 AI 产品 API。对于有效期内未使用的资源包剩余调用次数，本产品不支持退订。

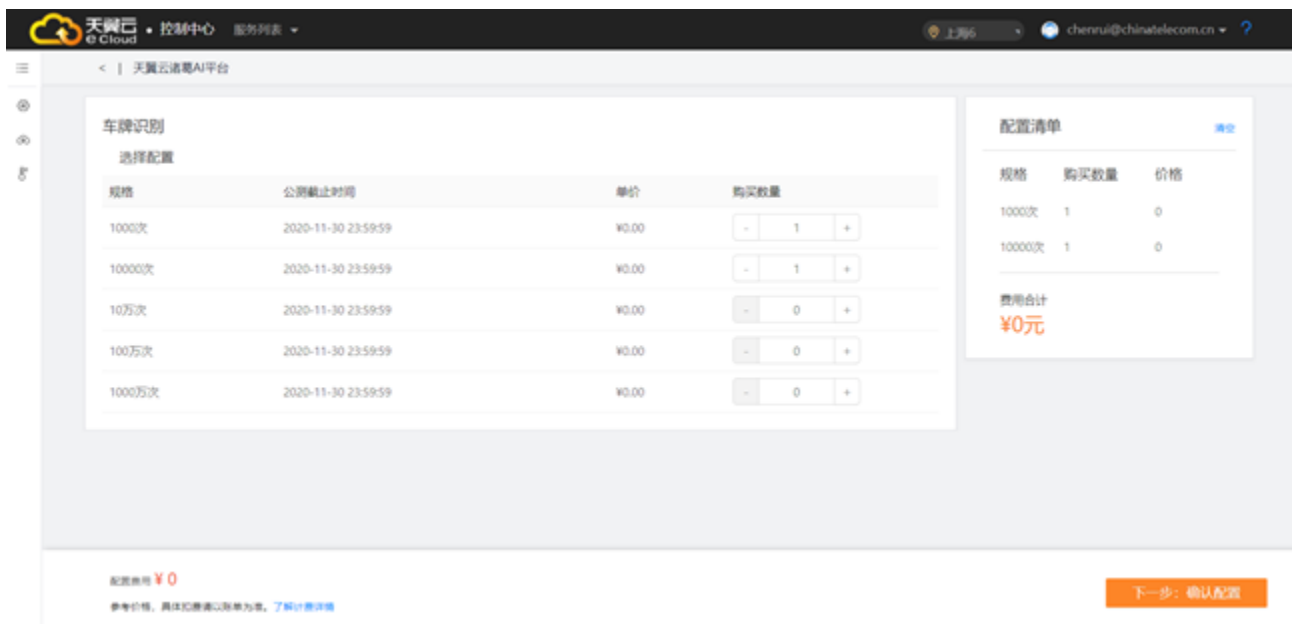
备注：产品价格以价格发文为准

## 3.2 产品购买

在控制台找到印刷文字识别页签，找到对应的产品，点击“购买”进行订购

以购买：车牌识别“产品为例：

在“车牌识别”列点击“购买”，进入产品规格及数量配置页面。



规格	公测截止时间	单价	购买数量
1000次	2020-11-30 23:59:59	¥0.00	- 1 +
10000次	2020-11-30 23:59:59	¥0.00	- 1 +
10万次	2020-11-30 23:59:59	¥0.00	- 0 +
100万次	2020-11-30 23:59:59	¥0.00	- 0 +
1000万次	2020-11-30 23:59:59	¥0.00	- 0 +

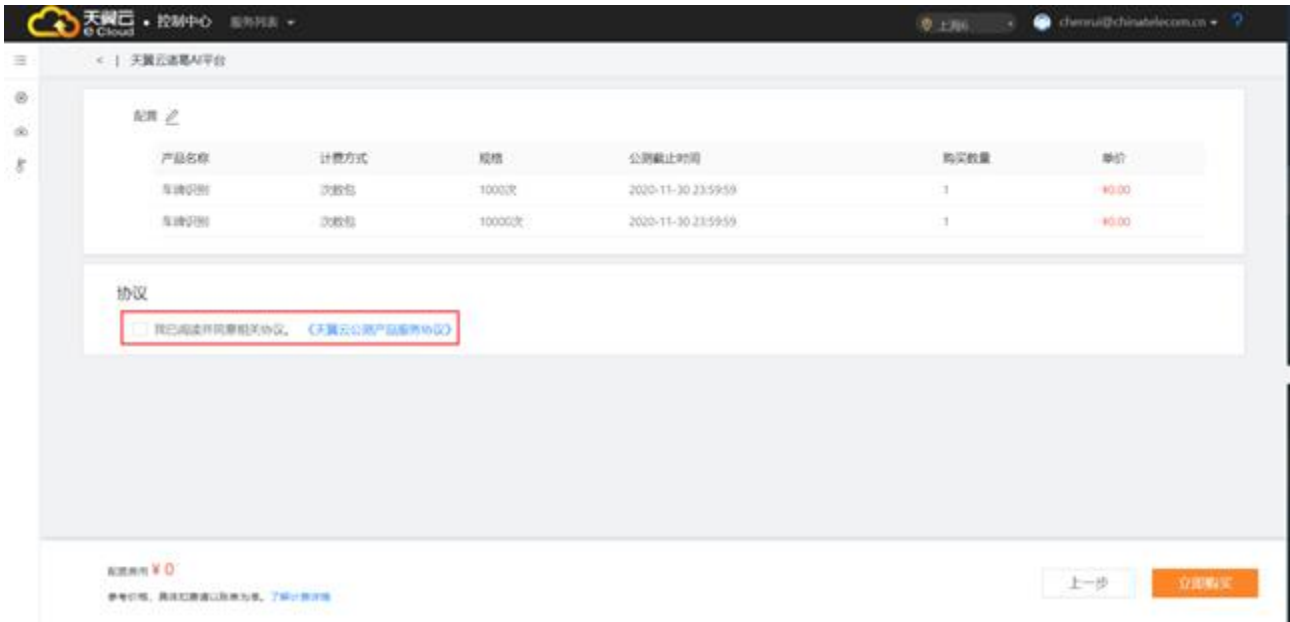
规格	购买数量	价格
1000次	1	0
10000次	1	0

费用合计  
**¥0元**

配置总价 ¥0  
参考价格，具体价格请以实际为准。了解详情

下一步：确认配置

用户根据需要选择要开通的资源包规格及数量，点击“下一步：确认配置”，进入配置确认界面。

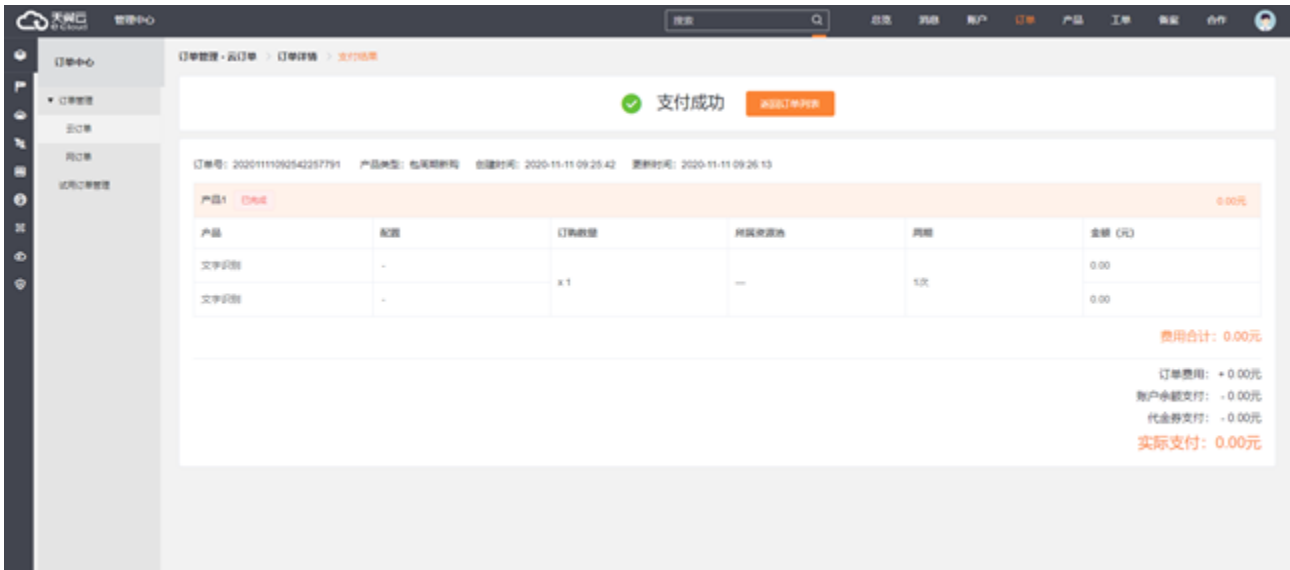


用户勾选服务协议，点击“立即购买”，进入订单支付页面



用户确认订单费用是否准确，费用无误点击“立即支付”。完成支付后进入“支付结果”界面。





### 3.3 产品续订

续订说明：当已订购的资源包订单即将到期或即将用完时，可通过订购新的资源包进行续订。

### 3.4 产品退订

退订说明：已购订单内的资源包不支持退订。

## 4.1 API 概览

### 4.1.1 概述

本说明提供了印刷文字识别产品 API 的描述、语法、参数说明及示例等内容。

### 4.1.2 API 概览

功能名称	描述
通用型 OCR	针对图片中的文字，进行 OCR 检测，返回检测到的文字内容及坐标信息。
身份证识别	针对图片中的身份证，进行 OCR 检测，返回检测到的姓名、身份证号码等信息
车牌识别	针对图片中的车牌，进行 OCR 检测，返回检测到的车牌内容及车牌位置坐标
护照识别	针对图片中的护照，进行 OCR 检测，返回检测到的护照号码、姓名、性别等信息
营业执照识别	针对图片中的营业执照，进行 OCR 检测，返回检测到的统一社会信用代码、名称、住所等信息
通信行程卡识别	针对图片中的营业执照，进行 OCR 检测，返回检测到的统一社会信用代码、名称、住所等信息

### 4.1.3 状态码

- 1、 请求状态码

正常状态码	描述
200	请求成功
3**	请求转移
4**	客户端错误
5**	服务端错误

## 2、 全局请求返回错误码

错误码	描述
10002	生成签名时官网 ak 信息错误
10020	签名错误
40002	缺少 appkey 头
40006	无效的 appkey
40008	不支持的请求类型
40009	IP 未被 APP 授权
40010	IP 未被 API 授权
50000	服务内部错误

50001	服务未注册
50002	应用未开通
50003	API 中无效的 URL 请求
51001	购买服务已过期
51002	收费 API 未购买
51003	API 可用次数已不足

## 4.2 如何调用 API

### 4.1.1 终端节点

<https://ai-global.ctapi.ctyun.cn>

### 4.1.2 构造请求

请求地址: {终端节点地址}+{对应接口 URI}

请求头:

Key	Value(说明)
Content-Type	application/json
ctyun-eop-request-id	用户请求 id, 通过 uuid 生成, 形如 33dfa732-b27b-464f-b15a-21ed6845afd5

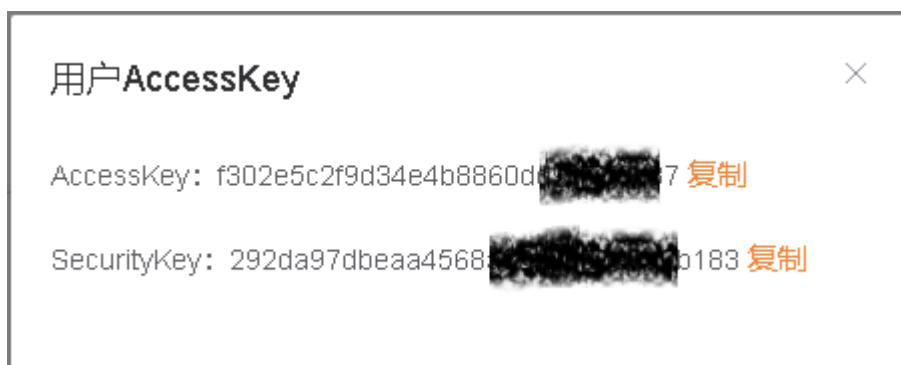
Eop- Authorization	由天翼云官网 accessKey 和 securityKey 经签名后生成，签名逻辑详见后续说明
eop-date	请求时间，形如 yyyyymmddTHHMMSSZ，例如 20211221T163014Z
host	终端节点域名
appkey	诸葛 AI 平台-控制台-我的应用中每个应用具有的 AppKey 信息，鉴权时需要加入 header

### 4.1.3 认证鉴权

#### 1 信息的获取

##### 云网平台获取

登录云网门户，在“控制台”->“个人中心”->“第三方账号绑定”，通过创建或者查看获取 ak, sk。



#### 2 基本签名流程

## ctyun-eop-ak/ctyun-eop-sk 基本签名流程

- 1、待签字符串：使用规范请求和其他信息创建待签字符串;
- 2、计算密钥：使用 HEADER、ctyun-eop-sk、ctyun-eop-ak 来创建 Hmac 算法的密钥;
- 3、计算签名：使用第三步的密钥和待签字符串在通过 hmacsha256 来计算签名。
- 4、签名应用：将生成的签名信息作为请求消息头添加到 HTTP 请求中。

### 3 创建待签名字符串

待签名字符串的构造规则如下：

待签名字符串=需要进行签名的 Header 排序后的组合列表 + "\n" + 排序的 query + "\n" + toHex(sha256(原封的 body))

需要进行签名的 Header 排序后的组合列表 (排序的 header)	header 以 header_name:header_value 来一个一个通过\n 拼接起来, EOP 是强制要求 ctyun-eop-request-id 和 eop-date 这个头作为 Header 中的一部分, 并且必须是待签名 Header 里的一个。需要进行签名算法的 Header 需要进行排序 (将它们的 header_name 以 26 个英文字母的顺序来排序), 将排序后得到的列表进行遍历组装成待签名的 header。
排序的 query	query 以&作为拼接, key 和值以=连接, 排序规则使用 26 个英文字母的顺序来排序, Query 参数全部都需要进行签名
toHex(sha256(原封的 body))	传进来的 body 参数进行 sha256 摘要, 对摘要出来的结果转十六进制

排序的 header 例子：

假设你需要将 ctyun-eop-request-id、eop-date、host 都要签名, 则待签名的 header 构造出来是：

```
ctyun-eop-request-id:123456789\neop-date:20210531T100101Z\nhost:1.1.1.1:9080\n
```

ctyun-eop-request-id、eop-date 和 host 的排序就是这个顺序, 如果你加入一个 ccad 的 header; 同时这个 header 也要是进行签名,则待签名的 header 组合：

```
ccda:123\nctyun-eop-request-id:123456789\neop-date:20210531T100101Z\nhost:1.1.1.1:9080\n
```

#### 4 构造动态密钥

发起请求时，需要构造一个 eop-date 的时间，这个时间的格式是 yyyyymmddTHHMMSSZ;言简意赅一些，就是年月日 T 时分秒 Z

- 1、先是拿你申请来的 ctyun-eop-sk 作为密钥，eop-date 作为数据，算出 ktime
- 2、拿 ktime 作为密钥，你申请来的 ctyun-eop-ak 数据，算出 kAk;
- 3、拿 kAk 作为密钥，eop-date 的年月日值作为数据；算出 kdate

eop-date	yyyyymmddTHHMMSSZ (20211221T163614Z) (年月日 T 时分秒 Z)
Ktime	使用 ctyun-eop-sk 作为密钥，eop-date 作为数据，算出 ktime; Ktime = hmacSha256(ctyun-eop-sk, eop-date)
kAk	使用 ktime 作为密钥，你申请来的 ctyun-eop-ak 数据，算出 kAk; kAk = hmacsha256(ktime,ctyun-eop-ak)
kdate	使用 kAk 作为密钥，eop-date 的年月日值作为数据；算出 kdate; kdate = hmacsha256(kAk, eop-date)

#### 5 签名应用及示例

由“构造动态密钥”和“创建待签名字符串”分别的出来的待签名字符串 string\_sigture、kdate 生成 Signature;

Signature	待签名字符串 string_sigture、kdate; 再根据 hmacsha256(kdate,string_sigture)得出的结果，再将结果进行 base64 编码得出 Sigture
Eop-Authorization	ctyun-eop-ak Header=你构造待签名字符串时的 header 排序 Signature (ctyun-eop-ak 后及 Signature 都有一个空格) header 排序以分号“;” 拼接 例子所述：你待签名的字符串 header 顺序是 eop-date 和 host; 那么 你加到 header 里的值就是

```
Eop-Authorization: ctyun-eop-ak Headers=eop-date;host Signature=xad01/ada
```

由上得到 Eop-Authorization, 然后将数据整合成 HEADER 放在 http\_client 内, 发出即可。

http\_client 所需请求头部如下:

```
Eop-Authorization: ctyun-eop-ak Headers= ctyun-eop-request-id;eop-date  
Signature=xad01/ada
```

```
eop-date:20211221T163614Z
```

```
ctyun-eop-request-id: 123456789
```

(注: 若需要进行签名的 Header 不止默认的 ctyun-eop-request-id 和 eop-date, 需要在 http\_client 的请求头部中加上, 并且 Eop-Authorization 中也需要增加)

## 4.1.4 Python 调用示例

```
import hmac  
  
import base64  
  
import hashlib  
  
import json  
  
import time  
  
import uuid  
  
import requests  
  
from urllib.parse import urlparse  
  
def sha256(content):  
    x = hashlib.sha256()  
    x.update(content.encode())  
    return x.hexdigest().upper()  
  
def hmac_sha256(key, content):  
    sign = hmac.new(key, content, digestmod="sha256").digest()
```



```
ret = base64.b64encode(sign)

return ret

# 计算签名

def get_signature(ak, sk, app_key, params):

    # 创建待签名字符串

    # 一、header 部分

    # 主要包括 3 个 header 需要作为签名内容: appkey、ctyun-eop-request-id、eop-date

    # 1. 首先通过 uuid 生成 ctyun-eop-request-id

    request_id = str(uuid.uuid1())

    # 2. 获取当前时间戳并对时间进行格式化

    now_time = time.localtime()

    eop_date = time.strftime("%Y%m%dT%H%M%SZ", now_time)

    eop_date_simple = time.strftime("%Y%m%d", now_time)

    # 3. 对 header 部分按照字母顺序进行排序并格式化

    camp_header = "appkey:{0}\nctyun-eop-request-id:{1}\neop-date:{2}\n".format(app_key, request_id, eop_date)

    # 二、query 部分

    # 对 url 的 query 部分进行排序

    parsed_url = urlparse(request_url)

    query = parsed_url.query

    query_params = sorted(query.split("&"))

    after_query = ""

    for query_param in query_params:

        if len(after_query) < 1:

            after_query += query_param

        else:

            after_query += "&" + query_param

    # 三、body 参数进行 sha256 摘要

    # sha256 body

    content_hash = sha256(json.dumps(params)).lower()

    # 完成创建待签名字符串

    pre_signature = camp_header + "\n" + after_query + "\n" + content_hash
```

```
# 构造动态密钥

k_time = hmac_sha256(sk.encode("utf-8"), eop_date.encode("utf-8"))

k_ak = hmac_sha256(base64.b64decode(k_time), ak.encode("utf-8"))

k_date = hmac_sha256(base64.b64decode(k_ak), eop_date_simple.encode("utf-8"))

# 签名的使用

signature = hmac_sha256(base64.b64decode(k_date), pre_signature.encode("utf-8"))

# 将数据整合得到真正的 header 中的内容

sign_header = "{0} Headers=appkey;ctyun-eop-request-id;eop-date Signature={1}".format(ak, signature.decode())

# 返回 request-id eop-date 和 sign_header

return request_id, eop_date, sign_header

# 向服务发送请求

def do_post(url, headers, params):

    response = requests.post(url, data=json.dumps(params), headers=headers)

    try:

        print(response.status_code)

        print(response.json())

    except AttributeError:

        print("请求失败")

if __name__ == '__main__':

    # 请求地址

    request_url = "https://ai-global.ctapi.ctyun.cn/v1/aiop/api/2f6hqix09mv4/face/PERSON/person/detectFaceFromBase64"

    # 官网 accessKey

    ctyun_ak = accessKey

    # 官网 securityKey

    ctyun_sk = 'securityKey'

    # 诸葛 AI 官网-控制台-我的应用中获取的 appKey

    ai_app_key = 'appKey'

    # body 内容从本地文件中获取

    # 打开图片文件

    f = open('test.jpeg', 'rb')

    img_base64 = base64.b64encode(f.read()).decode()
```

```
# body 内容

params = {"imageContent": img_base64}

# 调用 get_signature 方法获取签名

request_id, eop_date, sign_header = get_signature(ctyun_ak, ctyun_sk, ai_app_key, params)

# 生成请求 header

# 请求 header

headers = {

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

    'ctyun-eop-request-id': request_id,

    'appkey': ai_app_key,

    'Eop-Authorization': sign_header,

    'eop-date': eop_date,

    'host': 'ai-global.ctapi.ctyun.cn'

}

print("请求头部:")

print(headers)

# 执行 post 请求

do_post(request_url, headers, params)
```

## 4.1.5 Java 调用示例

java 的鉴权 sdk 及调用示例

请参考官网附件

ai-api-sign-java-sdk-1.0.0.zip

## 4.3 API

### 4.3.1 通用型 OCR

#### 1、接口描述

针对图片中的文字，进行 OCR 检测，返回检测到的文字内容及坐标信息。

#### 2、请求方法

POST

#### 3、接口要求

- 图片格式限制：图片格式支持 png、jpg、jpeg、bmp 格式
- 图片大小限制：单张图片大小不超过 10MB，批量请求单次不超过 50 张
- 图片尺寸限制：图片像素尺寸应大于 32x32，小于 5000x5000

#### 4、请求 URL

/v1/aiop/api/2f3p1pnxpm8/ocrdetect/ocr/v1/image.json

#### 5、请求参数

请求头 header 参数

参数	是否必填	参数类型	说明	示例	下级对象
Content-	是	string	json 格式	"application/json"	

参数	是否必填	参数类型	说明	示例	下级对象
Type					
appkey	是	string	诸葛 AI 应用 appkey	"562b89493b1a40e1b97ea05e50d8170"	

### 请求体 body 参数

参数	是否必填	参数类型	说明	示例	下级对象
data	是	list	一张或多张图像 Base64 编码数据构成的 list。注意：图片需要使用常规 base64 编码方式	["_9j_4AAQSkZJRg..."]	

## 6、请求代码示例

```

Curl -X POST "https://ai-global.ctapi.ctyun.cn/v1/aiop/api/2f3plnpxqm8/ocrdetect/ocr/v1/image.json"
-H "Content-Type: application/json"
-H "ctyun-eop-request-id:33dfa732-b27b-464f-b15a-21ed6845afd5"
-H "appkey:XXX"
-H "Eop-Authorization:XXX"
-H "eop-date:20211109T104641Z"
-H "host:ai-global.ctapi.ctyun.cn"

```

```
--data '{"data":["AAAAAAAAA..."]}'
```

复制

## 7、返回值说明

### 请求成功返回响应参数

参数	是否必填	参数类型	说明	示例	下级对象
code	是	int	请求响应状态码	0	
message	是	object	请求信息结构体， message["success"] 代表请求 list 中的成功数量， message["fail"]代表 请求 list 中的失败数量。	"success"	
result	是	object	识别的结果，按照列表形式排列，每个元素为图片对应的鉴定结果键值对，元素排序按照传入顺序排列，元素中包括识别到的文本行数，位置以及文本行内容		result

### 表 message

参数	是否必填	参数类型	说明	示例	下级对象
success	否	int	成功标识	1	
fail	否	int	错误标识	0	

**表 result**

参数	是否必填	参数类型	说明	示例	下级对象
text_line	否	int	图片识别到的文字行数	-	
detail	否	object	按照列表形式排列，每个元素为图片中每一个识别到的文本行的具体信息，包含位置以及文本行内容	-	detail

**表 detail**

参数	是否必填	参数类型	说明	示例	下级对象
name	否	string	序号	-	
text	否	string	识别到的文字	-	
box	否	list	文字所在的坐标为 8 个浮点数构成	-	

参数	是否必填	参数类型	说明	示例	下级对象
			的数组，8 个浮点数依次分别表示左上、右上、右下、左下四个点的 x 坐标，y 坐标。		

### 请求失败返回响应参数

参数	是否必填	参数类型	说明	示例	下级对象
code	是	int	错误码，参见本文档错误码说明部分	4003	
message	是	string/object	返回对应的可读消息，值为 error	"error"	message
details	是	string	返回对应的错误信息	"请求中未包 data 字段"	
result	否	object	返回对应的错误编		result



参数	是否必填	参数类型	说明	示例	下级对象
			码和错误信息		

表 message

参数	是否必填	参数类型	说明	示例	下级对象
success	否	int	成功标识	0	
fail	否	int	错误标识	1	

表 result

参数	是否必填	参数类型	说明	示例	下级对象
errorCode	是	string	错误码	"4008"	
errorMessage	是	string	错误信息	"base64 数据处理异常"	

## 8、返回值示例

### 请求成功返回值示例

```
{
  'code': 0,
  'message': {'success': 1, 'fail': 0},
  'result': [
```

```
{
  'text_line': 2,
  'detail': [
    {
      'name': '0',
      'text': '社会主义核心价值观',
      'box': [
        43.3963039677,
        157.3534469991,
        610.5843128949,
        169.5419591656,
        609.1529514034,
        236.1498488994,
        41.9649424762,
        223.9613367328
      ]
    },
    {
      'name': '1',
      'text': '富强、民主、文明、和谐',
      'box': [
        84.6885821036,
        244.0079416001,
        573.8530589198,
        248.174096093,
        573.4117596933,
        299.9887625014,
        84.2472828771,
        295.8226080086
      ]
    }
  ]
}
```

复制

### 请求失败返回值示例

示例 1:

```
{
  "code": 4003,
  "message": "error",
  "details": "请求中未包 data 字段"
}
```

示例 2:

```
{
  'code': 0,
  'message': {'success': 0, 'fail': 1},
  'result': [{'err_code': 4008, 'err_msg': 'base64 数据处理异常'}]
}
```

复制

## 9、状态码

状态码	描述
200	表示请求成功。

## 10、错误码说明

4 位错误码。4 开头为业务错误码，5 开头为服务错误码。

错误码	错误信息	错误描述
4001	请求 JSON 处理异常	请求 JSON 处理异常
4002	body 传入不是字典	body 传入不是字典
4003	请求中未包 data 字段	请求中未包 data 字段
4004	data 字段图片数据不是 list 格式	data 字段图片数据不是 list 格式
4005	图片 list 内图片数量为 0	图片 list 内图片数量为 0
4006	图片 list 内图片数量超过 50 张	图片 list 内图片数量超过 50 张
4008	base64 数据处理异常	base64 数据处理异常
4009	请求文件格式不合法，仅支持 jpeg/png/jpg/bmp 格式	请求文件格式不合法，仅支持 jpeg/png/jpg/bmp 格式
4010	单张图片大小超过 10M	单张图片大小超过 10M
4011	图片尺寸不符合要求,分辨率长宽尺寸	图片尺寸不符合要求,分辨率长宽尺寸

错误码	错误信息	错误描述
	应不高于 5000 不低于 32	应不高于 5000 不低于 32
5000	OCR 服务接口异常，请联系管理员	OCR 服务接口异常，请联系管理员

## 11、base64 编码规则：使用常规的 safe base64 编码方式

- python 中推荐使用 `base64.urlsafe_b64encode()` 函数进行编码。
- java 中推荐使用 `BASE64.getUrlEncoder().encodeToString()` 函数进行编码。

### 4.3.2 身份证识别

#### 1、 接口描述

针对图片中的身份证，进行 OCR 检测，返回检测到的姓名、身份证号码等信息

#### 2、 请求方法

POST

#### 3、 接口要求

- 图片格式限制：目前仅支持 png、jpg、jpeg、bmp 格式。
- 图片大小限制：图片单张大小不超过 10MB，批量请求单次不超过 50 张。
- 图片尺寸限制：图片像素尺寸应大于 32x32，小于 5000x5000。

#### 4、 请求 URL

```
/v1/aiop/api/2f3os7qq79xc/IdentityCard/ocr/v1/idcard.json
```

复制

#### 5、 请求参数

请求头 header 参数

参数	是否必填	参数类型	说明	示例	下级对象
Content-Type	是	string	json 格式	"application/json"	
appkey	是	string	诸葛 AI 应用 appkey	"562b89493b1a40e1b97ea05e50dd8170"	

### 请求体 body 参数

参数	是否必填	参数类型	说明	示例	下级对象
data	是	list	一张或多张图像 Base64 编码数据构成的 list。注意：图片需要使用常规 base64 编码方式	["_9j_4AAQSkZJRg..."]	

## 6、 请求代码示例

```
Curl -X POST "https://ai-global.ctapi.ctyun.cn/v1/aiop/api/2f3os7qq79xc/IdentityCard/ocr/v1/idcard.json"
-H "Content-Type: application/json"
-H "ctyun-eop-request-id:33dfa732-b27b-464f-b15a-21ed6845afd5"
```

```

-H "appkey:XXX"
-H "Eop-Authorization:XXX"
-H "eop-date:20211109T104641Z"
-H "host:ai-global.ctapi.ctyun.cn"
--data '{"data":["AAAAAAAAA..."]}'
复制

```

## 7、返回值说明

### 请求成功返回响应参数

参数	是否必填	参数类型	说明	示例	下级对象
code	是	int	请求响应状态码	0	
message	是	object	请求信息结构体， message["success"] 代表请求 list 中的成功数量， message["fail"]代表请求 list 中的失败数量。	"success"	
result	是	object	识别的结果，按照列表形式排列，每个元素为图片对应的鉴定结果键值对，元素排序按照传入顺序排列，元素中包括身份证正反面信息，以及姓名、性别、身份证		result

参数	是否必填	参数类型	说明	示例	下级对象
			号码等信息		

**表 message**

参数	是否必填	参数类型	说明	示例	下级对象
success	否	int	成功标识	1	
fail	否	int	错误标识	0	

**表 result**

参数	是否必填	参数类型	说明	示例	下级对象
note	否	string	身份的正面或反面信息，国徽面为正面：front，人像面为反面：back	-	
detail	否	object	返回结果键值集合，包含姓名、性别、身份证号码等信息	-	detail
idcard_face	否	string	身份照片的人脸图片	-	

参数	是否必填	参数类型	说明	示例	下级对象
			Base64 字符串		
demo_image	否	string	无效字段（可忽略）	-	
risk	否	string	判定身份证是原件还是复印件	-	

### 表 detail

参数	是否必填	参数类型	说明	示例	下级对象
name	否	string	姓名	-	
sex	否	string	性别	-	
nation	否	string	民族	-	
birthday	否	string	出生日期	-	
idn	否	string	身份证号	-	
addr	否	string	住址	-	
org	否	string	签发机关	-	
validperiod	否	string	有效期限	-	

### 请求失败返回响应参数



参数	是否必填	参数类型	说明	示例	下级对象
code	是	int	错误码，放置 API 对应的错误码	4003	
message	是	string/object	请求失败时返回值为 'error' 或 {'success':0,'fail':1}	"error"	message
details	否	string	返回对应的错误信息	"请求中未包含 data 字段"	
result	否	object	返回对应的错误编码和错误信息		result

**表 message**

参数	是否必填	参数类型	说明	示例	下级对象
success	否	int	成功标识	0	
fail	否	int	错误标识	1	

**表 result**

参数	是否必填	参数类型	说明	示例	下级对象
errorCode	是	string	错误码	"4008"	
errorMessage	是	string	错误信息	"base64 数据处理异常"	

## 8、返回值示例

### 请求成功返回值示例

```
{  
  
  'code': 0,  
  'message': {'success': 1, 'fail': 0},  
  'result':  
  [{ 'note': 'back',  
    'detail':  
    { 'name': '李创',  
      'sex': '男',  
      'nation': '壮',  
      'birthday': '2002年1月30日',  
      'addr': '云南省文山壮族苗族自治州文山市秉烈乡小平坝村民委坡头寨村二组 046 号',  
      'idn': '532621200201302715',  
      'org': '',  
      'validperiod': ''},  
      'idcard_face': 'no face',  
      'demo_image': 'demo_image',  
      'risk': 'normal'}]}  
}
```

复制

### 请求失败返回值示例

示例 1:

```
{
  "code": 4004,
  "message": "error",
  "details": "data 字段图片数据不是 list 格式"
}
```

示例 2:

```
{
  'code': 0,
  'message': {'success': 0, 'fail': 1},
  'result': [{'err_code': 4008, 'err_msg': 'base64 数据处理异常'}]
}
```

复制

## 9、状态码

状态码	描述
200	表示请求成功。

## 10、错误码说明

4 开头为业务错误码，5 开头为服务错误码。

错误码	错误信息	错误描述
4001	请求 JSON 处理异常	请求 JSON 处理异常
4002	body 传入不是字典	body 传入不是字典
4003	请求中未包 data 字段	请求中未包 data 字段
4004	data 字段图片数据不是 list 格式	data 字段图片数据不是 list 格式

错误码	错误信息	错误描述
4005	图片 list 内图片数量为 0	图片 list 内图片数量为 0
4006	图片 list 内图片数量超过 50 张	图片 list 内图片数量超过 50 张
4008	base64 数据处理异常	base64 数据处理异常
4009	请求文件格式不合法, 仅支持 jpeg/png/jpg/bmp 格式	请求文件格式不合法, 仅支持 jpeg/png/jpg/bmp 格式
4010	单张图片大小超过 10M	单张图片大小超过 10M
4011	图片尺寸不符合要求, 分辨率长宽尺寸应不高于 5000 不低于 32	图片尺寸不符合要求, 分辨率长宽尺寸应不高于 5000 不低于 32
5000	OCR 服务接口异常, 请联系管理员	OCR 服务接口异常, 请联系管理员

## 11、base64 编码规则：使用常规的 safe base64 编码方式

- python 中推荐使用 `base64.urlsafe_b64encode()` 函数进行编码。
- java 中推荐使用 `BASE64.getUrlEncoder().encodeToString()` 函数进行编码。

### 4.3.3 车牌识别

#### 1、 接口描述

对图片中的车牌（仅限中国大陆境内的蓝牌、黄牌（单层）、新能源绿牌），进行 OCR 检测，返回检测到的车牌内容及车牌位置坐标。

#### 2、 请求方法

POST

### 3、 接口要求

- 图片格式限制：目前仅支持 jpg、jpeg、png、bmp 等常见格式。
- 图片大小限制：图片单张大小不超过 10MB，批量请求单次不超过 50 张。
- 图片尺寸限制：图片像素尺寸应大于 32x32，小于 5000x5000。

### 4、 请求 URL

```
/v1/aiop/api/2gt54ed8660w/driven-plate-ocr/platelicense.json
```

复制

### 5、 请求参数

#### 请求头 header 参数

参数	是否必填	参数类型	说明	示例	下级对象
Content-Type	是	string	json 格式	"application/json"	
appkey	是	string	诸葛 AI 应用 appkey	"562b89493b1a40e1b97ea05e50dd8170"	

#### 请求体 body 参数

参数	是否必填	参数类型	说明	示例	下级对象
data	是	list	一张或多张图像 Base64 编码字符串构成的 list。注意：图片需要使用常规 safe base64 编码方式	["_9j_4AAQSkZJRg..."]	

## 6、 请求代码示例

Curl -X POST

```
"https://ai-global.ctapi.ctyun.cn/v1/aiop/api/2gt54ed8660w/driven-plate-ocr/platelicense.json"  
-H "Content-Type: application/json"  
-H "ctyun-eop-request-id:33dfa732-b27b-464f-b15a-21ed6845afd5"  
-H "appkey:XXX"  
-H "Eop-Authorization:XXX"  
-H "eop-date:20211109T104641Z"  
-H "host:ai-global.ctapi.ctyun.cn"  
--data '{"data":["_9j_4AAQSkZJRg..."]}'
```

复制

## 7、 返回值说明

请求成功返回响应参数

参数	是否必填	参数类型	说明	示例	下级对象
code	是	int	请求响应状态码，返回 0 表示成功，返回错误代码参考下面的错误代码列表。	0	
message	是	object	请求信息结构体， message["success"] 代表请求 list 中的成功数量， message["fail"]代表 请求 list 中的失败数量。	"success"	
result	是	object	识别的结果，按照列表形式排列，每个元素为图片对应的鉴定结果键值对，元素排序按照传入顺序排列，元素中包括识别到的车牌数目，位置以及车牌内容及其识别的置信度		result

表 message

参数	是否必填	参数类型	说明	示例	下级对象
success	否	int	成功标识	1	
fail	否	int	错误标识	0	

**表 result**

参数	是否必填	参数类型	说明	示例	下级对象
number of plate licenses	是	int	图片中识别到的车牌数量		
detail	是	list	图片内识别到的车牌 list 的具体信息，list 内每一项包括车牌位置、牌照类型、车牌内容、置信度等		detail

**表 detail**

参数	是否必填	参数类型	说明	示例	下级对象
box	是	list	识别到的车牌 8 坐标位置信息，包含四个数组元素，依次分别对应车		



参数	是否必填	参数类型	说明	示例	下级对象
			牌左上、右上、右下、左下四个点，每个数组元素包含两个整数，代表坐标点的 x 坐标，y 坐标		
plate_type	是	string	识别到的车牌种类信息，现阶段只支持中国大陆境内的蓝牌、黄牌（单层）、新能源绿牌	"蓝牌"	
platelicense	是	string	识别到的车牌内容	"冀 D137G5"	
confidence	是	float	识别到的车牌置信度	1.0	

#### 请求失败返回响应参数

参数	是否必填	参数类型	说明	示例	下级对象
code	是	int	错误码，放置 API 对应的错误码	4003	
messag	是	string/obje	请求失败时返回值为'error'或	"error	messag

参数	是否必填	参数类型	说明	示例	下级对象
e		ct	{'success':0,'fail':1}	"	e
details	否	string	返回对应的错误描述	"请求中未包data字段"	
result	否	object	返回对应的错误编码和错误信息		result

表 message

参数	是否必填	参数类型	说明	示例	下级对象
success	否	int	成功标识	0	
fail	否	int	错误标识	1	

表 result

参数	是否必填	参数类型	说明	示例	下级对象
errorCode	是	string	错误码	"4008"	

参数	是否必填	参数类型	说明	示例	下级对象
errorMessage	是	string	错误信息	"base64 数据处理异常"	

## 8、 返回值示例

### 请求成功返回值示例

```
{
  "code": 0,
  "message": {"success": 1, "fail": 0},
  "result": [{
    "number of plate licenses": 1,
    "detail": [{
      "box": [
        [123, 191], [214, 195], [213, 226], [122, 222]
      ],
      "plate_type": "蓝牌",
      "platelicense": "浙 C80000",
      "confidence": 0.99
    }]
  }]
}
```

复制

### 请求失败返回值示例

示例 1:

```
{
  "code": 4004,
  "message": "error",
  "details": "data 字段图片数据不是 list 格式"
}
```

示例 2:

```
{
  'code': 0,
```

```
'message': {'success': 0, 'fail': 1},  
'result': [{'err_code': 4008, 'err_msg': 'base64 数据处理异常'}]  
}
```

复制

## 9、状态码

状态码	描述
200	表示请求成功。

## 10、错误码说明

4 开头为业务错误码，5 开头为服务错误码。

错误码	错误信息	错误描述
4001	请求 JSON 处理异常	请求 JSON 处理异常
4002	body 传入不是字典	body 传入不是字典
4003	请求中未包 data 字段	请求中未包 data 字段
4004	data 字段图片数据不是 list 格式	data 字段图片数据不是 list 格式
4005	图片 list 内图片数量为 0	图片 list 内图片数量为 0
4006	图片 list 内图片数量超过 50 张	图片 list 内图片数量超过 50 张
4008	base64 数据处理异常	base64 数据处理异常
4009	请求文件格式不合法，仅支持 jpeg/png/jpg/bmp 格式	请求文件格式不合法，仅支持 jpeg/png/jpg/bmp 格式

错误码	错误信息	错误描述
4010	单张图片大小超过 10M	单张图片大小超过 10M
4011	图片尺寸不符合要求,分辨率长宽尺寸应不高于 5000 不低于 32	图片尺寸不符合要求,分辨率长宽尺寸应不高于 5000 不低于 32
5000	OCR 服务接口异常, 请联系管理员	OCR 服务接口异常, 请联系管理员

## 11、base64 编码规则：使用常规的 safe base64 编码方式

- python 中推荐使用 `base64.urlsafe_b64encode()` 函数进行编码。
- java 中推荐使用 `BASE64.getUrlEncoder().encodeToString()` 函数进行编码。
- 

### 4.3.4 护照识别

#### 1、接口描述

针对新版护照图片，进行 OCR 检测，返回检测到的护照号码、姓名、性别等信息。本接口不适用于老版护照图片。

备注：接口只允许单张图片请求，不允许图片 list。

#### 2、请求方法

POST

#### 3、接口要求

- 目前仅支持 base64 编码方式请求，暂不支持 url 方式请求。一次请求仅支持一条数据。
- 图片格式限制：目前仅支持 jpg、jpeg、png、bmp 等常见格式。
- 图片大小限制：图片大小不超过 7 MB。
- 图片尺寸限制：图片像素尺寸应不小于 32x32，不大于 5000x5000。

## 4、请求 URL

/v1/aiop/api/2k8jsx0rxmo/passport/ocr/v1/Passport.json

## 5、请求参数

### 请求头 header 参数

参数	是否必填	参数类型	说明	示例	下级对象
Content-Type	是	string	json 格式	"application/json"	
appkey	是	string	诸葛 AI 应用 appkey	"562b89493b1a40e1b97ea05e50dd8170"	

### 请求体 body 参数

参数	是否必填	参数类型	说明	示例	下级对象
Action	是	string	算法名称，只能为 "PassportOCR"	PassportOCR	

参数	是否必填	参数类型	说明	示例	下级对象
ImageData	是	string	图片 Base64 编码数据，图片需要使用常规 safe base64 编码方式，不包含前缀 "data:img/jpg;base64,"	"_9j_4AAQSkZJRgA"	

## 6、请求代码示例

```

Curl -X POST
"https://ai-global.ctapi.ctyun.cn/v1/aiop/api/2k8jsx0rzm0/passport/ocr/v1/Passport.json"
-H "Content-Type: application/json"
-H "ctyun-eop-request-id:33dfa732-b27b-464f-b15a-21ed6845afd5"
-H "appkey:XXX"
-H "Eop-Authorization:XXX"
-H "eop-date:20211109T104641Z"
-H "host:ai-global.ctapi.ctyun.cn"
--data '{"Action": "PassportOCR", "ImageData": "_9j_4AAQSkZJRgA.....Zuv_9k="}'
复制

```

## 7、返回值说明

### 请求成功返回响应参数

参数	是否必填	参数类型	说明	示例	下级对象
code	是	int	返回状态，返回 0 表示成功，返回错误代码参考下面的错误代码列表。	0	
message	是	string	如果 code 为 0，返回 success；如果 code 非 0，则返回对应的可读错误信息。	"success"	
result	是	object	返回的新版护照识别结果对象		result

**表 result**

参数	是否必填	参数类型	说明	示例	下级对象
license_type	是	string	护照的整体判断信息， "New_Passport"表示图片是新版护照， "Not	"New_Passport"	



参数	是否必填	参数类型	说明	示例	下级对象
			Passport"表示图片不是护照		
detail	是	object	返回结果键值集合，包含护照号码、姓名、性别等信息		
detail.country_code	是	string	国家码	"CHN"	
detail.passport_no	是	string	护照号码	"EXXXX7053"	
detail.name	是	string	姓名	"陈 XX"	
detail.name_pinyin	是	string	姓名拼音	"CHEN,XXXXXX XX"	
detail.sex	是	string	性别	"男/M"	
detail.nationality	是	string	国籍	"中国 /CHINESE"	

参数	是否必填	参数类型	说明	示例	下级对象
detail.date_of_birth	是	string	出生日期	"19XXX01"	
detail.plate_of_birth	是	string	出生地点	"山东/SHANDONG"	
detail.date_of_issue	是	string	签发日期	"2013XX24"	
detail.plate_of_issue	是	string	签发地点	"上海/SHANGHAI"	
detail.date_of_expiry	是	string	有效期至	"2023XX23"	
detail.authority	是	string	签发机关	"公安部出入境管理局"	
detail.mrzcode1	是	string	MRZ 代码 1		
detail.mrzcode2	是	string	MRZ 代码 2		

#### 请求失败返回响应参数



复制

### 请求失败返回值示例

```
{  
  "code": 400003,  
  "message": "body 内容为空",  
  "details": "Body 请求数据为空，没有包含内容"  
}
```

复制

## 9、状态码

状态码	描述
200	表示请求成功。

## 10、错误码说明

6 位错误码。4 开头为业务错误码，5 开头为服务错误码。

错误码	错误信息	错误描述
400003	Body 内容为空	Body 请求数据为空，没有包含内容
400006	必传的参数未传	必须的参数（Action、ImageData）未传
400008	请求体的字段类型错误	请求体的字段（Action、ImageData）类型错误，类型只能为字符串，不能为其他类型
400009	Action 值设置错误	Action 值设置错误
400010	ImageData 字段值为空字符	ImageData 字段值为空字符
400011	ImageData 字段	ImageData 字段的 base64 字符串转换字节码

错误码	错误信息	错误描述
	base64 数据处理异常	异常
400012	请求文件格式不合法	仅支持 jpeg/png/jpg/bmp 格式
400013	图片文件大小不符合要求	该文件大小不符合要求，静态图片要求小于 7M
400014	图片解码错误	字节码解码为图片错误
400015	图片尺寸不符合要求	分辨率长宽尺寸应不高于 5000 不低于 32
500001	服务接口异常，请联系管理员	需要联系管理员处理

## 11、支持的证件示例

本接口支持新版护照图片，如图所示：





## 4.3.5 营业执照识别

### 1、接口描述

针对营业执照图片，进行 OCR 检测，返回检测到的统一社会信用代码，注册号，名称，类型，住所，法定代表人，注册资金，成立日期，营业期限，实收资本。

### 2、请求方法

POST

### 3、接口要求

- 图片格式限制：图片格式支持 png、jpg、jpeg、bmp 格式
- 图片大小限制：单张图片大小不超过 7MB
- 图片尺寸限制：图片像素尺寸应不小于 32x32，不大于 5000x5000

### 4、请求 URL

```
/v1/aiop/api/2k8jo3ghhjwg/businesslicense/ocr/v1/BusinessLicense.json
```

复制

### 5、请求参数

请求头 header 参数

参数	是否必填	参数类型	说明	示例	下级对象
Content-Type	是	string	json 格式	"application/json"	



参数	是否必填	参数类型	说明	示例	下级对象
appkey	是	string	诸葛 AI 应用 appkey	"562b89493b1a40e1b97ea05e50d8170"	

### 请求体 body 参数

参数	是否必填	参数类型	说明	示例	下级对象
Action	是	string	请求参数放置于 http body 中，action 代表需要请求的具体算法，只能为 "BusinessCardOCR"	"BusinessCardOCR"	
ImageData	是	string	一张图像 Base64 编码数据 string。图片需要使用常规 safe base64 编码方式，不包含前缀 "data:img/jpg;base64,"		

## 6、请求代码示例

```
Curl -X POST "https://ai-global.ctapi.ctyun.cn/v1/aiop/api/2k8jo3ghhjwg/businesslicense/ocr/v1/BusinessLicense.json"
-H "Content-Type: application/json"
-H "ctyun-eop-request-id:33dfa732-b27b-464f-b15a-21ed6845afd5"
-H "appkey:XXX"
-H "Eop-Authorization:XXX"
-H "eop-date:20211109T104641Z"
-H "host:ai-global.ctapi.ctyun.cn"
--data '{"Action": "BusinessCardOCR", "ImageData": "9j_4AAQSkZJRgABAQAAQABAAD_4QAwRXhpZgAATUOAKgAAA..."}'
```

复制

## 7、返回值说明

### 请求成功返回响应参数

参数	是否必填	参数类型	说明	示例	下级对象
code	是	int	返回状态，返回 0 表示成功，返回错误代码参考下面的错误代码列表	0	
message	是	string	如果 code 为 0，返回 success；如果 code 非 0，则返回对应的可读错误信息	success	
result	是	object	识别的结果字典，字典内包营业执照的整体判断信息以及统一社会信用代码、名称、住所等		result

参数	是否必填	参数类型	说明	示例	下级对象
			详细键值对信息		

表 result

参数	是否必填	参数类型	说明	示例	下级对象
license_type	是	string	营业执照的整体判断信息， 'Business License'表示图片是营业执照， 'Not Business License'表示图片不是营业执照	"Business License"	
detail	是	object	返回结果键值集合，包含统一社会信用代码、名称、住所等信息	-	detail

表 detail

参数	是否必填	参数类型	说明	示例	下级对象
union_id	是	string	统一社会信用代码	-	
regist_no	是	string	注册号	-	
name	是	string	名称	-	
type	是	string	类型	-	
addr	是	string	住所	-	
legal_rep	是	string	法定代表人	-	
regist_captial	是	string	注册资金	-	
establish_date	是	string	成立日期	-	
operation_period	是	string	营业期限	-	
real_captial	否	string	实收资本（检测到双列营业执照返回此项，单列营业执照无此项信息）	-	

#### 请求失败返回响应参数

参数	是否必填	参数类型	说明	示例	下级对象
code	是	int	错误码，参见本	400009	

参数	是否必填	参数类型	说明	示例	下级对象
			文档错误码说明部分		
message	是	string	返回对应的错误信息	"Action 值设置错误"	
details	是	string	返回对应的错误描述	"Action 值设置错误"	

## 8、返回值示例

### 请求成功返回值示例

```
{
  "code": 0,
  "message": "success",
  "result": {
    "license_type": "Business License",
    "detail": {
      "union_id": "",
      "regist_no": "32050XXXXXX8749",
      "name": "苏州市 XXXX 有限公司",
      "type": "有限公司（自然人控股）",
      "addr": "苏州市相城区 XXXXXXXXX",
      "legal_rep": "潘 X",
      "regist_captial": "200 万元人民币",
      "establish_date": "2005 年 6 月 9 日",
      "operation_period": "2005 年 06 月 09 日至 2015 年 06 月 08 日",
      "real_captial": "200 万元人民币"
    }
  }
}
```

复制

## 请求失败返回值示例

```
{  
  "code": 400009,  
  "message": "Action 值设置错误",  
  "details": "Action 值设置错误"  
}
```

复制

## 9、状态码

状态码	描述
200	表示请求成功。

## 10、错误码说明

6 位错误码。4 开头为业务错误码，5 开头为服务错误码。

错误码	错误信息	错误描述
400003	body 内容为空	Body 请求数据为空，没有包含内容
400004	body 请求体非 json 格式	body 内容需要符合 json 要求
400005	请求体类型错误	请求体需为字典，不能为其他类型
400006	必传的参数未传	必须的参数 ('Action'、'ImageData') 未传
400007	传递非法参数	body 字典内有除 ('Action'、'ImageData') 外的参数
400008	请求体的字段类型错误	请求体的字段 ('Action'、'ImageData') 类型错误，类型只能为字符串，不能为其他类型

错误码	错误信息	错误描述
400009	Action 值设置错误	Action 值设置错误
400010	'ImageData'字段值为空字符	'ImageData'字段的 base64 字符串转换字节码异常
400011	'ImageData'字段 base64 数据处理异常	'ImageData'字段 base64 数据处理异常
400012	请求文件格式不合法	仅支持 jpeg/png/jpg/bmp 格式
400013	图片文件大小不符合要求	该文件大小不符合要求,静态图片要求小于 7M
400014	图片解码错误	字节码解码为图片错误
400015	图片尺寸不符合要求	分辨率长宽尺寸应不高于 5000 不低于 32
500001	服务接口异常, 请联系管理员	需要联系管理员处理

## 11、base64 编码规则：使用常规的 safe base64 编码方式

- python 中推荐使用 `base64.urlsafe_b64encode()` 函数进行编码。
- java 中推荐使用 `BASE64.getUrlEncoder().encodeToString()` 函数进行编码。

### 4.3.6 事业单位法人证书识别

#### 1、接口描述

针对事业单位法人证书图片，进行 OCR 检测，返回检测到的统一社会信用代码、名称、住所、法定代表人、开办资金、举办单位、有效期、经费来源等信息。

## 2、请求方法

POST

## 3、接口要求

- 图片格式限制：目前仅支持 png、jpg、jpeg、bmp 格式。
- 图片大小限制：单张图片大小不超过 7 MB。
- 图片尺寸限制：图片像素尺寸应不小于 32x32，不大于 5000x5000。

## 4、请求 URL

```
/v1/aiop/api/2k8jrue2nk84/shiyefaren/ocr/v1/Shiyefaren.json
```

复制

## 5、请求参数

### 请求头 header 参数

参数	是否必填	参数类型	说明	示例	下级对象
Content-Type	是	string	json 格式	"application/json"	
appkey	是	string	诸葛 AI 应用 appkey	"562b89493b1a40e1b97ea05e50dd8170"	

### 请求体 body 参数



参数	是否必填	参数类型	说明	示例	下级对象
Action	是	string	请求参数放置于 http body 中，action 代表需要请求的具体算法，只能为 "SyfrLicenseOCR"	"SyfrLicenseOCR"	
ImageData	是	string	一张图像 Base64 编码数据 string。图片需要使用常规 safe base64 编码方式		

## 6、请求代码示例

```
Curl -X POST "https://ai-global.ctapi.ctyun.cn/v1/aiop/api/2k8jrue2nk84/shiyefaren/ocr/v1/Shiyefaren.json"
-H "Content-Type: application/json"
-H "ctyun-eop-request-id:33dfa732-b27b-464f-b15a-21ed6845afd5"
-H "appkey:XXX"
-H "Eop-Authorization:XXX"
-H "eop-date:20211109T104641Z"
-H "host:ai-global.ctapi.ctyun.cn"
--data '{"Action": "SyfrLicenseOCR", "ImageData": "9j_4AAQSkZJRgABAQAAQABAAD_4QAwRXhpZgAATUOAKgAAA..."}'
```

复制

## 7、返回值说明

### 请求成功返回响应参数

参数	是否必填	参数类型	说明	示例	下级对象
code	是	int	返回状态，返回 0 表示成功，返回错误代码参考下面的错误代码列表	0	
message	是	string	如果 code 为 0，返回 success；如果 code 非 0，则返回对应的可读错误信息	"success"	
result	是	object	识别的结果字典，字典内包事业单位法人证书的整体判断信息以及统一社会信用代码、名称、住所等详细键值对信息		result

表 result

参数	是否必填	参数类型	说明	示例	下级对象
license_type	是	string	事业单位法人证书的整体判断信息，'Shiyefaren License'表示图片是事业	-	

参数	是否必填	参数类型	说明	示例	下级对象
			单位法人证书, 'Not Shiyefaren License'表示图片不是事业单位法人证书		
detail	是	object	返回结果键值集合, 包含统一社会信用代码、名称、住所等信息	-	detail

### 表 detail

参数	是否必填	参数类型	说明	示例	下级对象
union_id	是	string	统一社会信用代码	-	
name	是	string	名称	-	
addr	是	string	住所	-	
legal_rep	是	string	法定代表人	-	
regist_capital	是	string	开办资金	-	
organization	是	string	举办单位	-	
operation_period	是	string	有效期	-	

参数	是否必填	参数类型	说明	示例	下级对象
money_source	是	string	经费来源		

#### 请求失败返回响应参数

参数	是否必填	参数类型	说明	示例	下级对象
code	是	int	错误码，参见本文档错误码说明部分	400009	
message	是	string	返回对应的错误信息	"Action 值设置错误"	
details	是	string	返回对应的错误描述	"Action 值设置错误"	

## 8、返回值示例

#### 请求成功返回值示例

```
{
  "code": 0,
  "message": "success",
  "result": {
    "license_type": "Shiyefaren License",
    "detail": {
      "union_id": "12350XXXXXXXXX6547U",
      "name": "福建中国 XXXXXX",
      "addr": "泉州市 XXXXXX 北侧",
      "legal_rep": "林 XX",
      "regist_captial": "¥60 万元",

```

```
"organization": "中国 XXXXXX 委员会",  
"operation_period": "自 2016 年 04 月 05 日至 2021 年 04 月 05 日",  
"money_source": "财政拨款"  
}  
}  
}
```

复制

### 请求失败返回值示例

```
{  
  "code": 400009,  
  "message": "Action 值设置错误",  
  "details": "Action 值设置错误"  
}
```

复制

## 9、状态码

状态码	描述
200	表示请求成功。

## 10、错误码说明

6 位错误码。4 开头为业务错误码，5 开头为服务错误码。

错误码	错误信息	错误描述
400003	body 内容为空	Body 请求数据为空，没有包含内容
400004	body 请求体非 json 格式	body 内容需要符合 json 要求
400005	请求体类型错误	请求体需为字典，不能为其他类型

错误码	错误信息	错误描述
400006	必传的参数未传	必须的参数 ('Action'、'ImageData') 未传
400007	传递非法参数	body 字典内有除 ('Action'、'ImageData') 外的参数
400008	请求体的字段类型错误	请求体的字段 ('Action'、'ImageData') 类型错误, 类型只能为字符串, 不能为其他类型
400009	Action 值设置错误	Action 值设置错误
400010	'ImageData' 字段值为空字符	'ImageData' 字段值为空字符
400011	'ImageData' 字段 base64 数据处理异常	'ImageData' 字段的 base64 字符串转换字节码异常
400012	请求文件格式不合法	仅支持 jpeg/png/jpg/bmp 格式
400013	图片文件大小不符合要求	该文件大小不符合要求, 静态图片要求小于 7M
400014	图片解码错误	字节码解码为图片错误
400015	图片尺寸不符合要求	分辨率长宽尺寸应不高于 5000 不低于 32
500001	服务接口异常, 请联系管理员	需要联系管理员处理

## 11、base64 编码规则：使用常规的 safe base64 编码方式

- python 中推荐使用 `base64.urlsafe_b64encode()` 函数进行编码。
- java 中推荐使用 `BASE64.getUrlEncoder().encodeToString()` 函数进行编码。

## 4.3.7 通信行程卡识别

### 1、接口描述

针对图片中的通信行程卡，进行 OCR 检测，返回检测到的行程卡颜色，电话号码，请求时间，到达城市，风险城市。

备注：接口只允许单张图片请求，不允许图片 list。

### 2、请求方法

POST

### 3、接口要求

- 目前仅支持 base64 编码方式请求，暂不支持 url 方式请求。一次请求仅支持一条数据。
- 图片格式限制：目前仅支持 jpg、jpeg、png、bmp 等常见格式。
- 图片大小限制：图片大小不超过 7 MB。
- 图片尺寸限制：图片像素尺寸应不小于 32x32，不大于 5000x5000。

### 4、请求 URL

/v1/aiop/api/2xfwz87g6s4c/travel-card/ocr/detect

### 5、请求参数

请求头 header 参数

参数	是否必填	参数类型	说明	示例
Content-Type	是	string	json 格式	"application/json"

参数	是否必填	参数类型	说明	示例
appkey	是	string	诸葛 AI 应用 appkey	"562b89493b1a40e1b97ea05e50dd8170"

### 请求体 body 参数

参数	是否必填	参数类型	说明	示例
Action	是	string	算法名称, 只能为 "RecognizeTravelCardOCR"	"RecognizeTravelCardOCR"
ImageData	是	string	图片 Base64 编码数据, 图片需要使用常规 safe base64 编码方式, 不包含前缀 "data: img/ jpg; base64, "	"_9j_4AAQSkZJRgA..."

## 6、请求代码示例

```
Curl -X POST
"https://ai-global.ctapi.ctyun.cn/v1/aiop/api/2xfwz87g6s4c/travel-card/ocr/detect"
-H "Content-Type: application/json"
-H "ctyun-eop-request-id:33dfa732-b27b-464f-b15a-21ed6845afd5"
-H "appkey:XXX"
-H "Eop-Authorization:XXX"
-H "eop-date:20211109T104641Z"
-H "host:ai-global.ctapi.ctyun.cn"
--data '{"Action": "RecognizeTravelCardOCR", "ImageData": "_9j_4AAQSkZJRgA..."}'
```



## 7、返回值说明

请求成功返回响应参数

参数	是否必填	参数类型	说明	示例	下级对象
code	是	int	返回状态，返回 0 表示成功，返回错误代码参考下面的错误代码列表。	0	
message	是	string	如果 code 为 0，返回 success；如果 code 非 0，则返回对应的可读错误信息。	"success"	
result	是	object	返回的通信行程卡识别结果对象		result

表 result

参数	是否必填	参数类型	说明	示例	下级对象
Color	是	string	行程卡颜色	"绿色"	
Telephone	是	string	电话号码	"1895828"	
Time	是	string	图片时间	"2022. 04. 23 20:12:53"	

参数	是否必填	参数类型	说明	示例
ReachedCity	是	list	到达城市，按照列表形式排列	["河北省廊坊市", "重庆市"]
Notes	是	string	备注	"（注：*表示当前该城市存在中风险或高风险地区，并不表示用户实际到访过这些中高风险地区。）"
RiskAreas	是	list	风险城市，按照列表形式排列	["重庆市"]

#### 请求失败返回响应参数

参数	是否必填	参数类型	说明	示例
code	是	int	错误码，放置 API 对应的错误码	400003
message	是	string	返回对应的错误信息	"body 内容为空"
details	是	string	返回对应的错误描述	"Body 请求数据为空，没有包含内容"

## 8、返回值示例

请求成功返回值示例

```
{
  "code": 0,
  "message": "success",
  "result": {
    "Color": "绿色",
    "Telephone": "189****5828",
    "Time": "2022.04.23 20:12:53",
    "ReachedCity": ["河北省廊坊市", "重庆市"],
    "Notes": "（注：*表示当前该城市存在中风险或高风险地区，并不表示用户实际到访过这些中高风险地区。）",
    "RiskAreas": ["重庆市"]
  }
}
```

请求失败返回值示例

```
{
  "code": 400003,
  "message": "body 内容为空",
  "details": "Body 请求数据为空，没有包含内容"
}
```

## 9、状态码

状态码	描述
200	表示请求成功。

## 10、错误码说明

6 位错误码。4 开头为业务错误码，5 开头为服务错误码。

错误码	错误信息	错误描述
400003	body 内容为空	Body 请求数据为空，没有包含内容

错误码	错误信息	错误描述
400004	body 请求体非 json 格式	body 内容需要符合 json 要求
400005	请求体类型错误	请求体需为字典，不能为其他类型
400006	必传的参数未传	必须的参数 ('Action'、'ImageData') 未传
400007	传递非法参数	body 字典内有除 ('Action'、'ImageData') 外的参数
400008	请求体的字段类型错误	请求体的字段 ('Action'、'ImageData') 类型错误，类型只能为字符串，不能为其他类型
400009	Action 值设置错误	Action 值设置错误
400010	'ImageData' 字段值为空字符	'ImageData' 字段值为空字符
400011	'ImageData' 字段 base64 数据处理异常	'ImageData' 字段的 base64 字符串转换字节码异常
400012	请求文件格式不合法	仅支持 jpeg/png/jpg/bmp 格式
400013	图片文件大小不符合要求	该文件大小不符合要求, 静态图片要求小于 7M
400014	图片解码错误	字节码解码为图片错误
400015	图片尺寸不符合要求	分辨率长宽尺寸应不高于 5000 不低于 32
500001	服务接口异常, 请联系管理员	需要联系管理员处理

## 11、base64 编码规则：使用常规的 safe base64 编码方式

- python 中推荐使用 `base64.urlsafe_b64encode()` 函数进行编码。

- java 中推荐使用 `BASE64.getUrlEncoder().encodeToString()` 函数进行编码。

## 4.3 更新历史

更新日期	更新内容
2020-12-11	第一次正式发布。
2022-10-25	第二次正式发布。本次更新说明如下：修改 API 文档格式；
2022-11-29	第三次正式发布。本次更新说明如下：修改 API 文档格式；
.....	.....

# 5 常见问题

## 5.1 计费类

### 印刷文字识别支持哪些计费方式？

我们只提供封顶资源包的计费方式。

### 印刷文字识别是否支持续订？

可以支持，当已订购的资源包订单即将到期或即将用完时，可通过订购新的资源包进行续订。

### 印刷文字识别是否支持退订？

不支持，已购订单内的资源包不支持退订。

## 5.2 购买类

**印刷文字识别服务是否支持代金券付款？**

支持。

**印刷文字识别服务是否支持试用？**

在天翼云官网订购，可以支持一个包含有限次数的资源包（100/1000/1 万次，根据能力不同）试用。

**印刷文字识别服务如何开通？**

- 1, 在“通用型 OCR” 列点击“购买”，进入产品规格及数量配置页面。
- 2, 用户根据需要选择要开通的资源包规格及数量，点击“下一步：确认配置”，进入配置确认界面。
- 3, 用户勾选服务协议，点击“立即购买”，进入订单支付页面
- 4, 用户确认订单费用是否准确，费用无误点击“立即支付”。完成支付后进入“支付结果”界面，即开通成功。

## 5.3 操作类

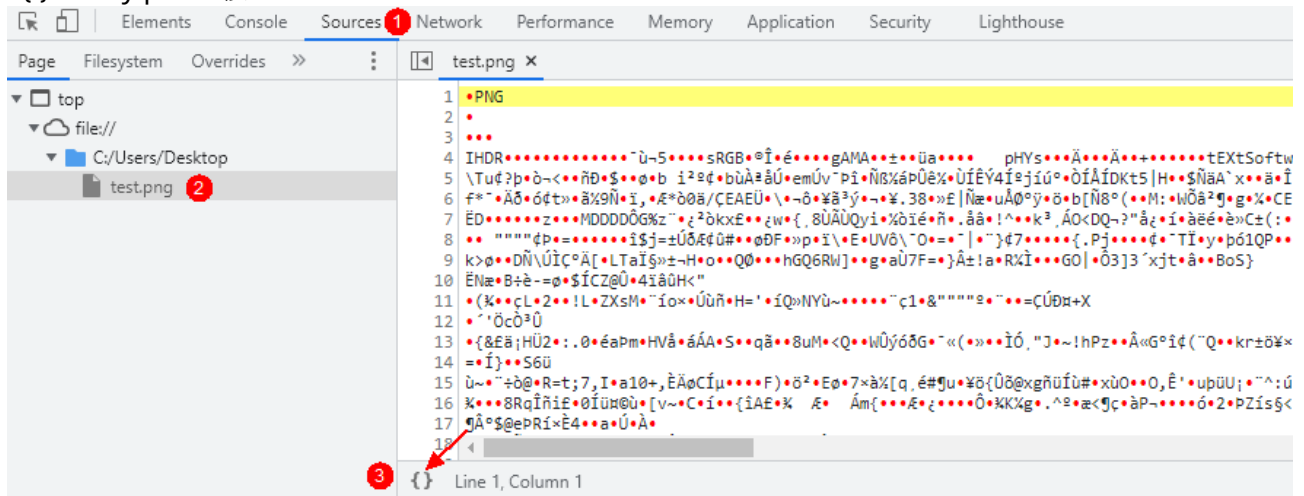
**印刷文字识别在什么时候进入可使用状态？**

当您支付费用且系统扣款成功后，将自动为您开通服务。

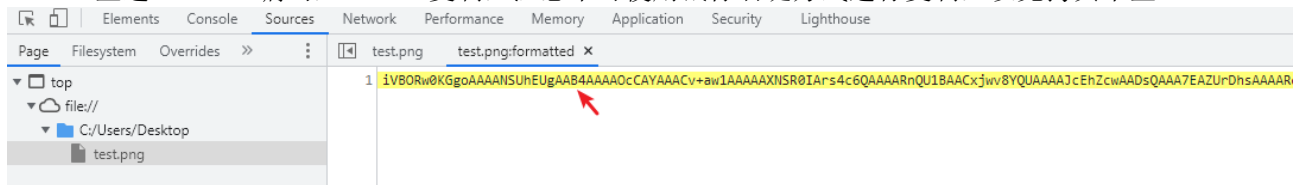
**如何获取图片 base64 编码？**

1. 将需转码的图片拖拽至 Chrome 浏览器中，浏览器上显示该图片。

- 按下“F12”，在弹出窗口中选择“Sources”，在左侧导航树中选择需要编码的图片所在路径，单击“{} Pretty print”按钮。



- 图片的 base64 编码显示在右侧界面中，如下图箭头中内容，选中图片的 base64 编码信息，Ctrl+A 全选 base64 编码，Ctrl+C 复制，注意不可使用鼠标右键方式进行复制，以免拷贝不全。



### 印刷文字识别 API 是否可以混用？

不建议混用，例如有专门的身份证识别，就不建议使用“通用 OCR”识别身份证图片，不同接口底层算法不同，混用会导致识别效果差。

### 印刷文字识别服务支持批量识别吗？

OCR 服务只支持调用一次接口识别一张图片，批量识别需要进行二次开发，编码循环调用 API，实现批量调用服务识别图片。

### 印刷文字识别服务识别结果可以转化为 Word、TXT、pdf 吗？

OCR 提取之后返回的结果是 JSON 格式，需要用户通过编程，将结果保存为 Word、TXT、pdf 格式。

### 印刷文字识别的并发是多少？

公有云服务，线上用户资源共享，并发量会根据线上用户的调用情况动态调整。

如遇到突发高峰导致的并发量不够用的情况，您可以尝试以下两种解决方法：

- 1.通过重试机制，碰到并发错误可以延时一小段时间（如 2-5s）重试请求。
- 2.后端检查上一请求结果，上一个请求返回后再发送下一次，避免请求过于频繁。
- 3.如果需要更大的常态性并发需求，请通过天翼云客服联系我们。

## 5.4 使用限制

**印刷文字识别前端展示页面对于上传的图片是否有要求？**

- (1) 图片单张大小不超过 2MB；
- (2) 图片格式支持 jpg、png、bmp 等。

**印刷文字识别 API 接口对于上传的图片是否有要求？**

- (1) 图片单张大小不超过 7MB，单次 1 张；
- (2) 图片像素尺寸大于 32x32 ， 小于 5000x5000；
- (3) 图片格式支持 png、jpg、jpeg、bmp 格式

**印刷文字识别是 HTTP GET 请求还是 HTTP POST 请求？**

HTTP POST 请求。

**印刷文字识别的请求图像放置在 http 请求的哪部分？**

请求图片应当放置于 http 的 body 中，不能放置于 query 或者 header 中

**印刷文字识别的 api 接口是否支持单次请求多张图片？**

接口只允许单张图片请求，不允许图片 list，且按请求中的图片数量来进行计费。