天翼云人脸识别

用户操作手册

天翼云科技有限公司

目 录

1 产品简介

1.1 产品定义

人脸识别是中国电信云公司自研AI平台提供的产品之一，通过自主研发图像识别人工 智能模型，提供人脸系列API服务-人脸检测，人脸属性识别，是否戴口罩识别，人脸活体检测，人脸比对。通过订购天翼云人脸识别产品，可将此服务快速高效的部署到您的应用中。

天翼云用户通过订购以上人脸识别的应用，无需进行模型开发仅需 API 接口对接到 AI 上云，即可轻松、高效、准确的实现云上产品的人脸识别。人脸识别将高效帮助用户识 别用户人脸信息，帮助用户在安防、人脸支付等领域实现智能化应用落地。

1.2 功能特性

1.2.1 人脸检测

通过用户输入人脸照片，高效快速地检测图像中的人脸，给出人脸的位置和可信度阈值，方便 用户后续应用。

1.2.2 人脸属性识别

在人脸检测基础上，分析检测后的人脸性别、年龄属性，扩展用户使用人脸检测服务的应用场 景。

1.2.3 人脸比对

通过对用户输入的两张人脸图像进行比对，分析是否为同一个人；同时可返回两张图像人脸的 相似度以及比对结果等。

1.2.4 人脸活体检测

基于深度学习方法，分析人脸图像的摩尔纹、成像畸形等信息，实现静默活体判断，有效防止 照片等非活体攻击。

1.2.5 是否戴口罩识别

准确高效识别图像中人脸是否正确佩戴口罩，支持多种类型口罩识别，并可对图片中多张人脸 实时检测，有效应对疫情场景。

1.3 产品优势

●识别高效且精准

支持实时识别，具备出色的多人脸检测能力。

能在复杂自然场景中提供高精度的人脸识别。

●支持戴口罩识别

可判断是否戴口罩，识别人脸数量、位置及是否佩戴口罩等关键信息。

对戴口罩人脸并可处理一定程度的遮挡，进行精准人脸匹配。

●支持活体检测和属性识别

有效对抗活体攻击，并可识别人脸属性。

支持扩展用户使用人脸检测服务的应用场景，完善场景解决方案。

●支持跨平台部署

支持服务端和移动端，覆盖多种业务场景。

提供易于集成的API，使开发人员能够轻松地将人脸属性识别功能整合到其应用程序中。

1.4 应用场景

1.4.1 门禁人脸识别

随着人们生活水平的提高，人们更加注重家居环境的安全，安防观念不断加强；伴随着这种需求的提 高，智能门禁系统应运而生，越来越多的企业、商铺、家庭都安装了各种各样的门禁系统。

1.4.2 市场营销-企业商务

面部识别技术在营销上主要有两方面的应用：首先，可以识别一个人的基本个人信息，例如性别、大致 年龄，以及他们看过什么，看了多久等。户外广告公司，例如 Val Morgan Outdoor（VMO），开始采用面部识别技术来收集消费者数据。其次，该技术可以用于识别已知的个人，例如小偷，或者已经加入系统的会员。这方面的应用已经引起一些服务提供商和零售商的注意。

1.4.3 商业银行

1.利用人脸识别技术防范网络风险

利用人脸识别技术准确认定持卡人的真实身份，确保持卡人的资金安全。另外，还可以通过人脸识别技 术进一步锁定不法分子，有利于公安机关快速破案。

2.脸识别技术在治理假钞方面的应用

银行的 ATM 机中没有假钞鉴别设备，只是在清机人员放入现金前做了鉴别，这样的措施并不够完善，

且容易造成银行与持卡人之间的纠纷。即使是现金存款机（CRS）有假钞鉴别功能，但往往因为假钞识别 特征提取的滞后，而被不法分子所利用。不法分子先存入假钞，然后马上在柜台或其他自助设备

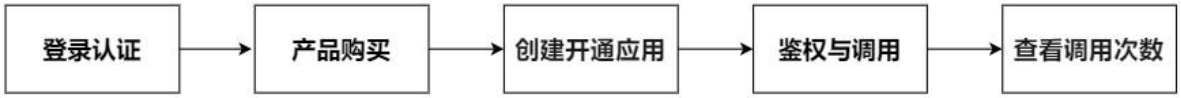
上提取　真钞，以此手段谋取不法利益。

2 快速入门

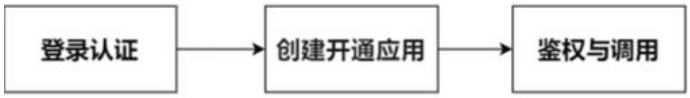
人脸识别服务，是基于人的脸部特征信息，利用计算机对人脸图像进行处理、分析和理解，进行身份识别的一种智能服务。

人脸识别以开放API的方式提供给用户，用户通过实时访问和调用API获取人脸处理结果，帮助用户自动进行人脸的识别、比对以及相似度查询等，打造智能化业务系统，提升业务效率。

- 商用/公测产品步骤流程



- 限免产品步骤流程



2.1 登录认证

2.1.1 注册天翼云账号

打开[天翼云官网](https://www.ctyun.cn/)，点击右上角【免费注册】按照操作提示完成账号注册，成为天翼云客户。

2.1.2 天翼云账号实名认证

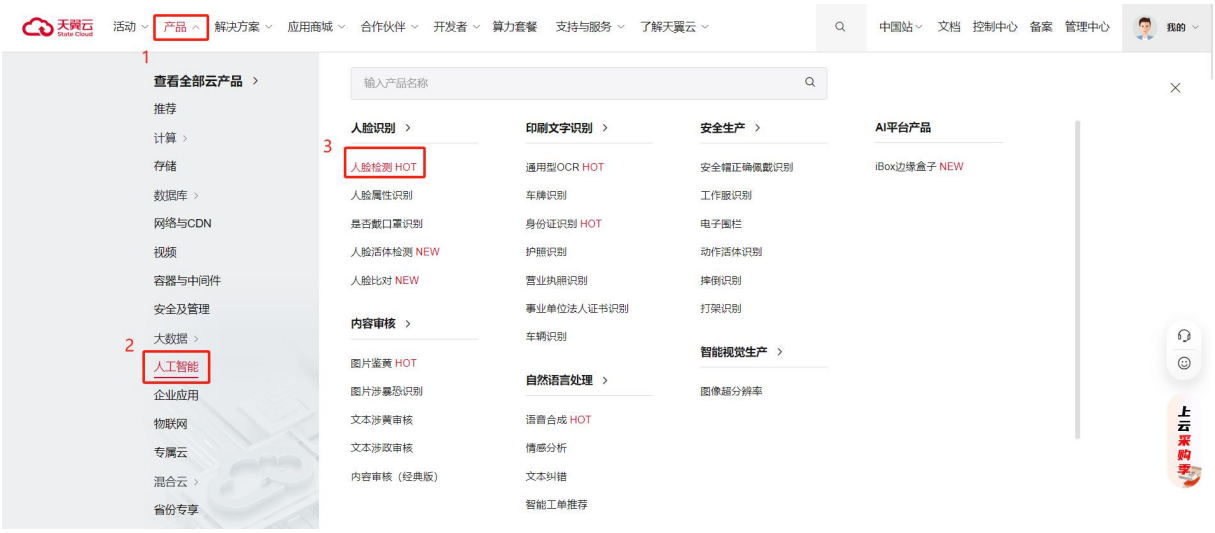
天翼云账号需要进行实名制认证后，才可以购买和使用产品，请务必完成实名认证操作。

1.进入[个人中心]页面，在左侧导航栏，点击【实名认证】，按照操作提示完成账号实名认证。

2.如果您是企业用户，推荐进行企业认证，以便获取更多便利。

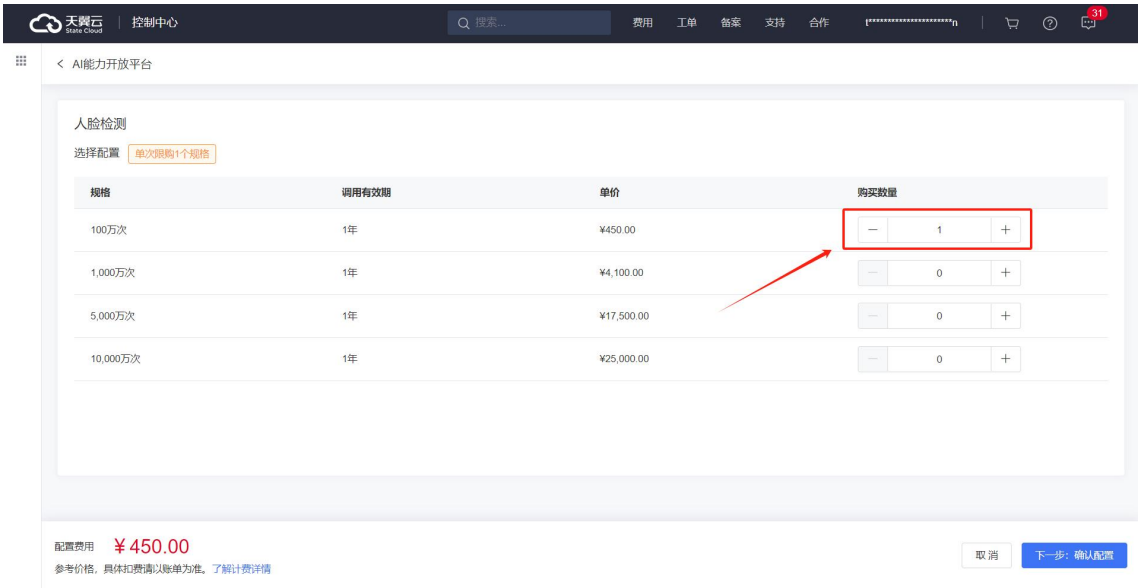
2.2 购买服务

1.点击【产品-人工智能】，根据需要选择人脸识别下的对应产品（以人脸检测为例）。

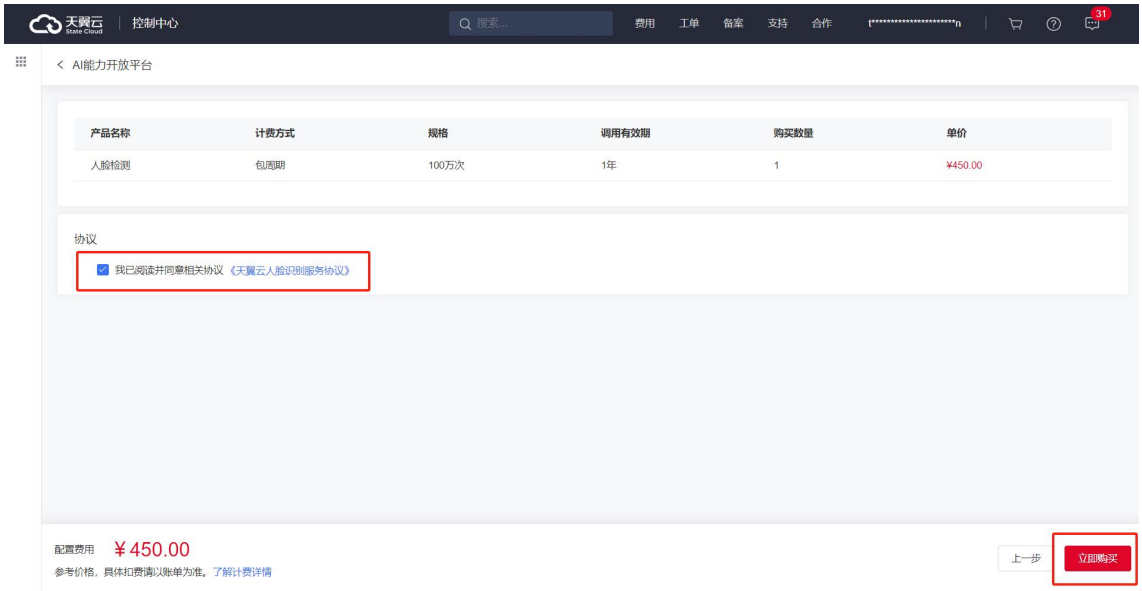


2.跳转到服务详情页后，点击【立即购买】。

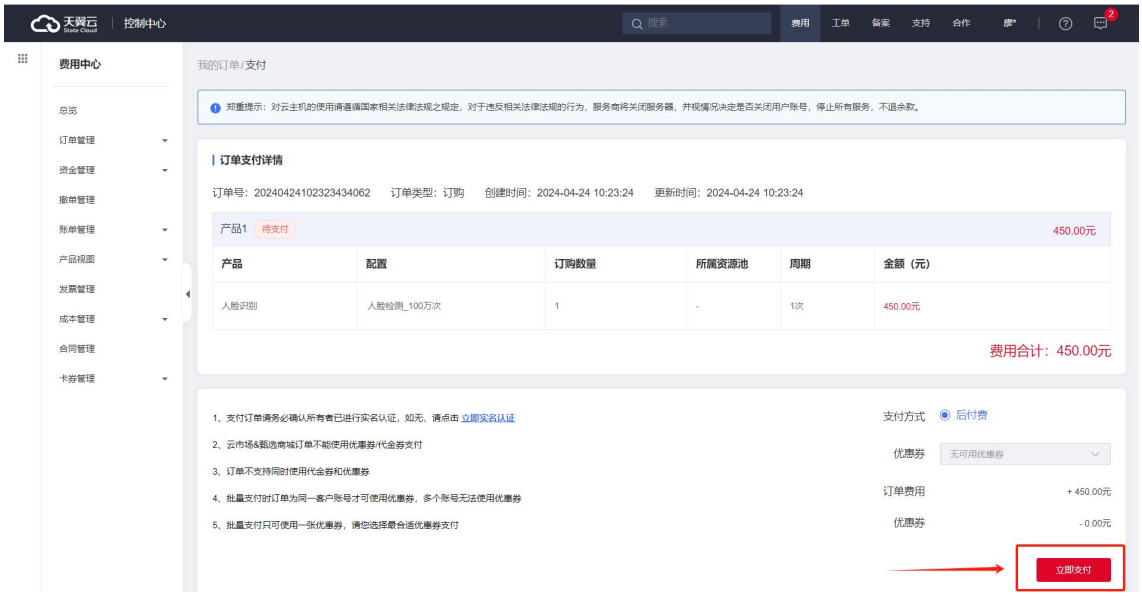
3.选择自己需要购买的资源包规格，点击购买数量，点击【下一步：确认配置】。

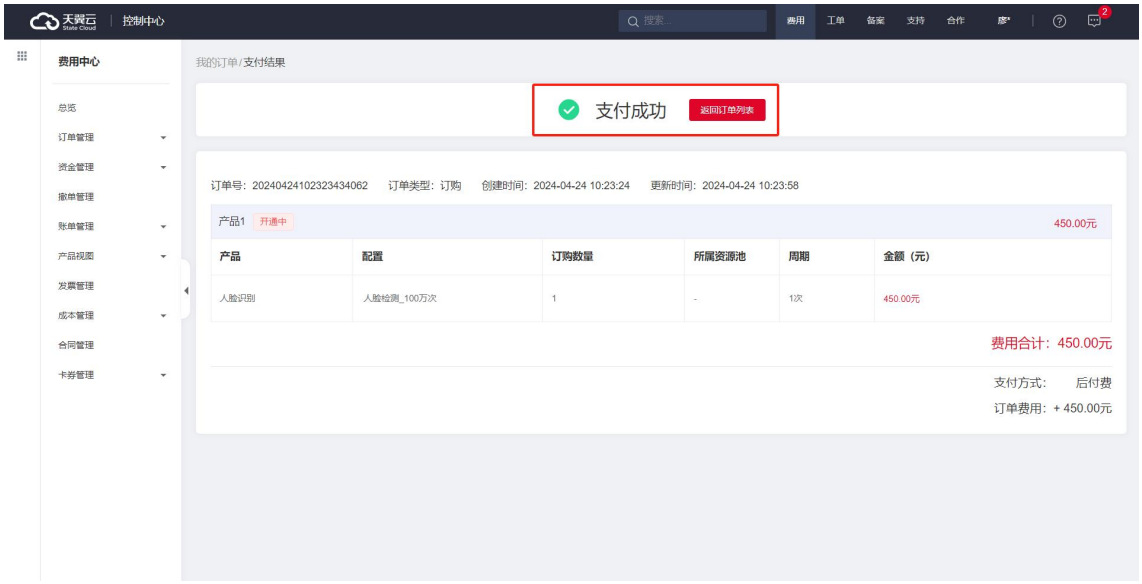


4.点击勾选同意相关协议，点击【立即购买】。



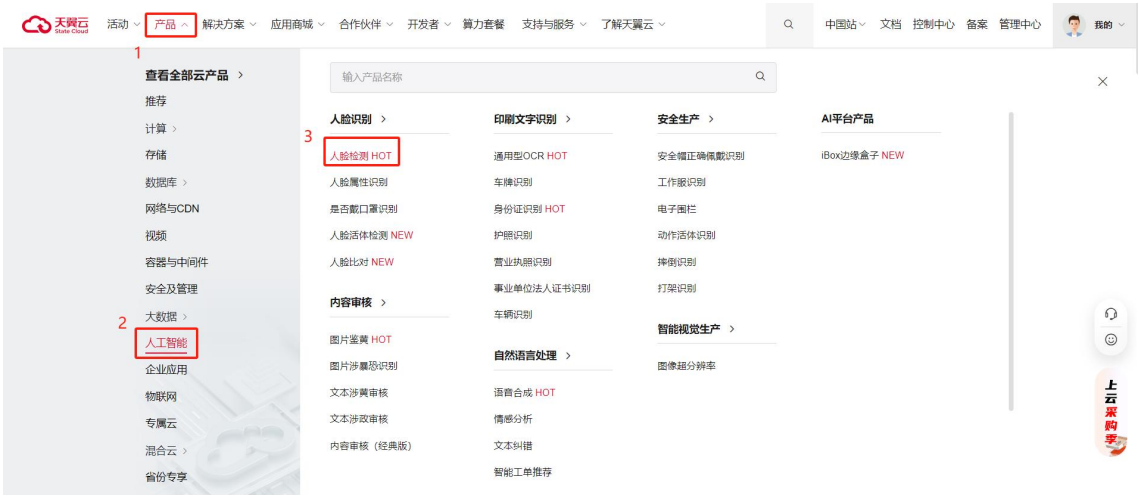
5.支付订单。点击【立即支付】，跳转到支付成功界面即购买成功。





2.3 开通服务

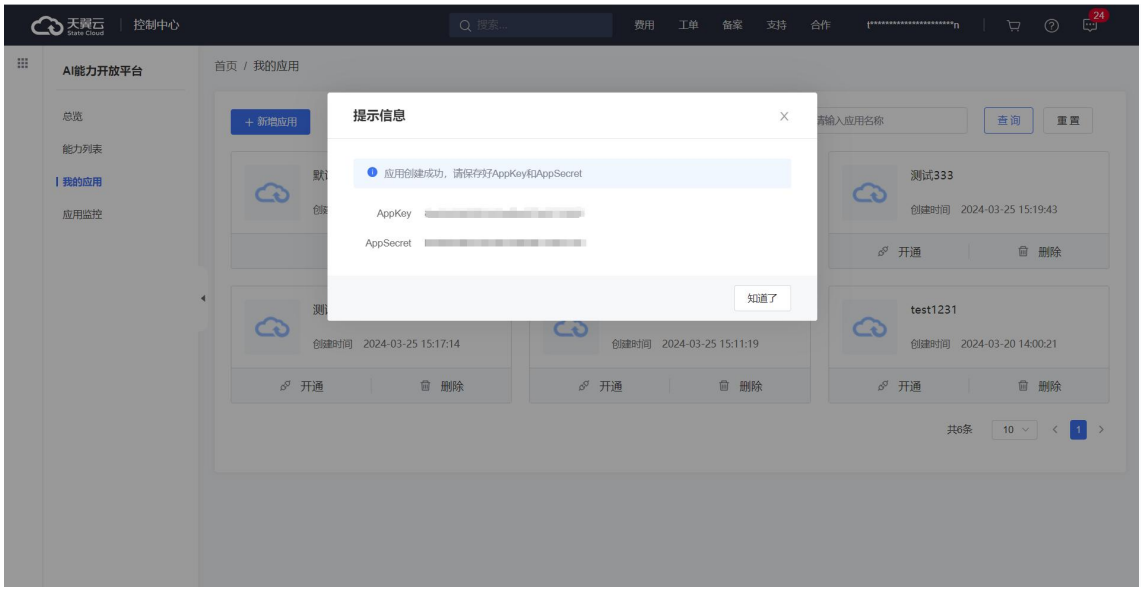
1.点击【产品-人工智能】，根据需要选择人脸识别下的对应产品（以人脸检测为例）。



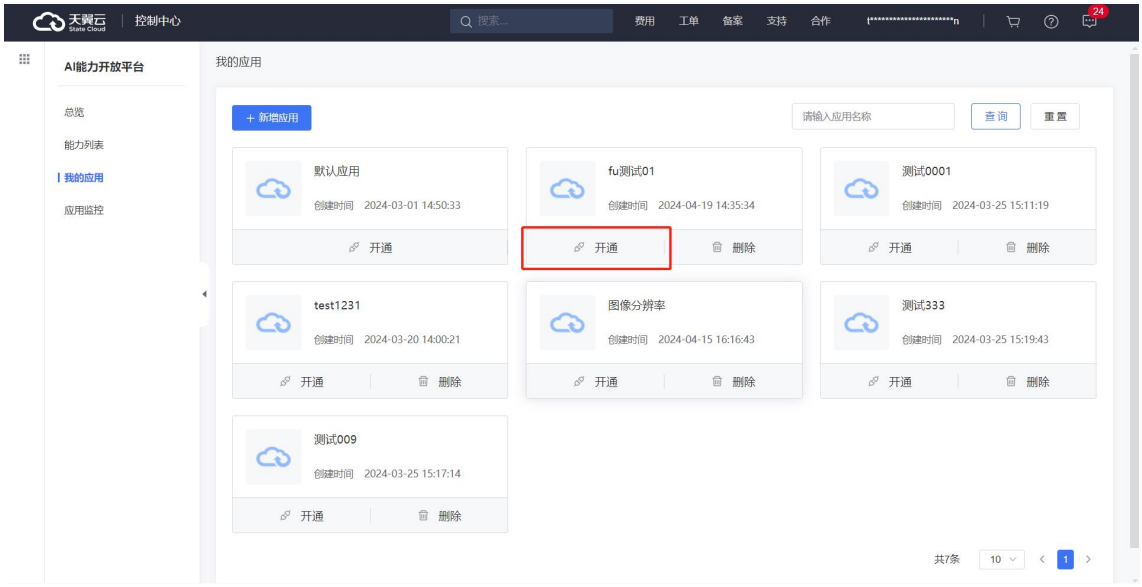
2.跳转到服务详情页后，点击【立即开通】。

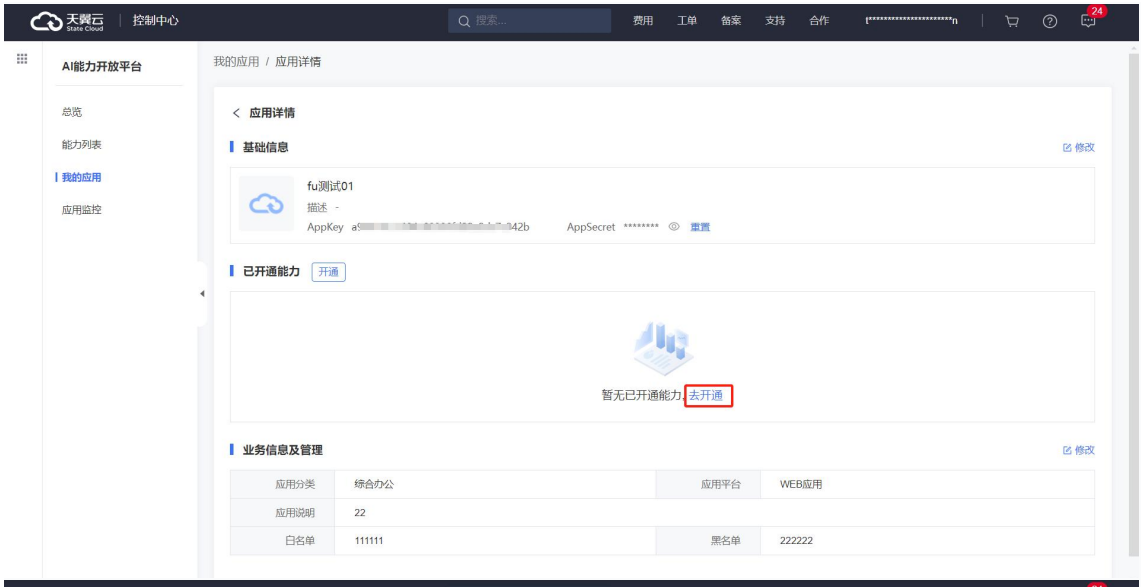
3.进入控制台【我的应用】，点击【新增应用】，页面出现新建应用的弹窗，填写应用名称及应用概述，点击【确定】按钮。

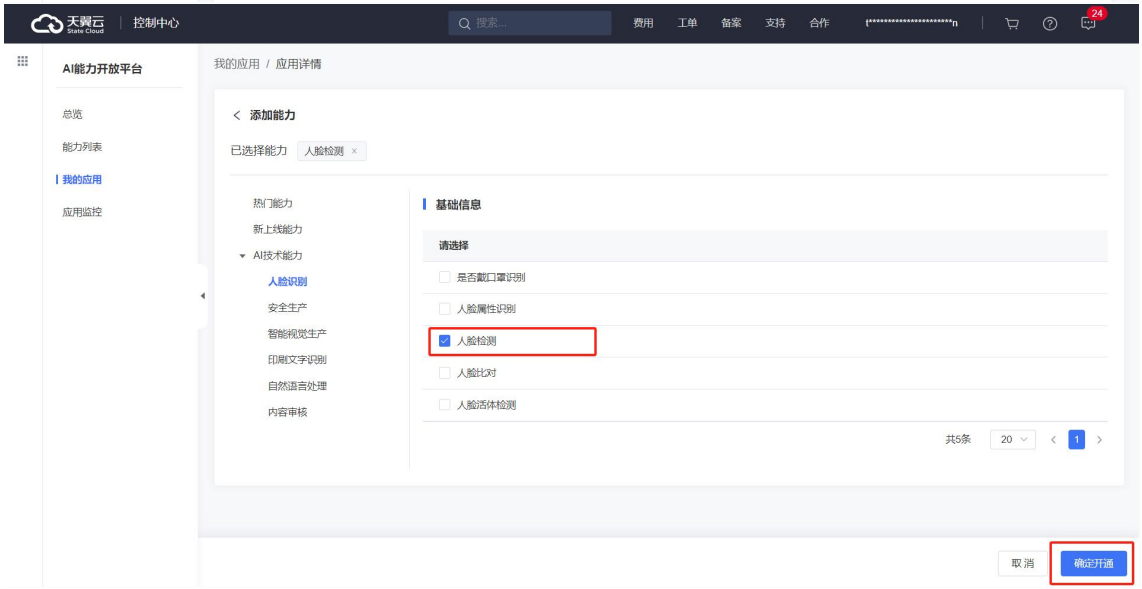
4.再次弹出小弹窗，提示应用创建成功，请保存好AppKey和AppSecret，点击【知道了】即新应用创建成功。



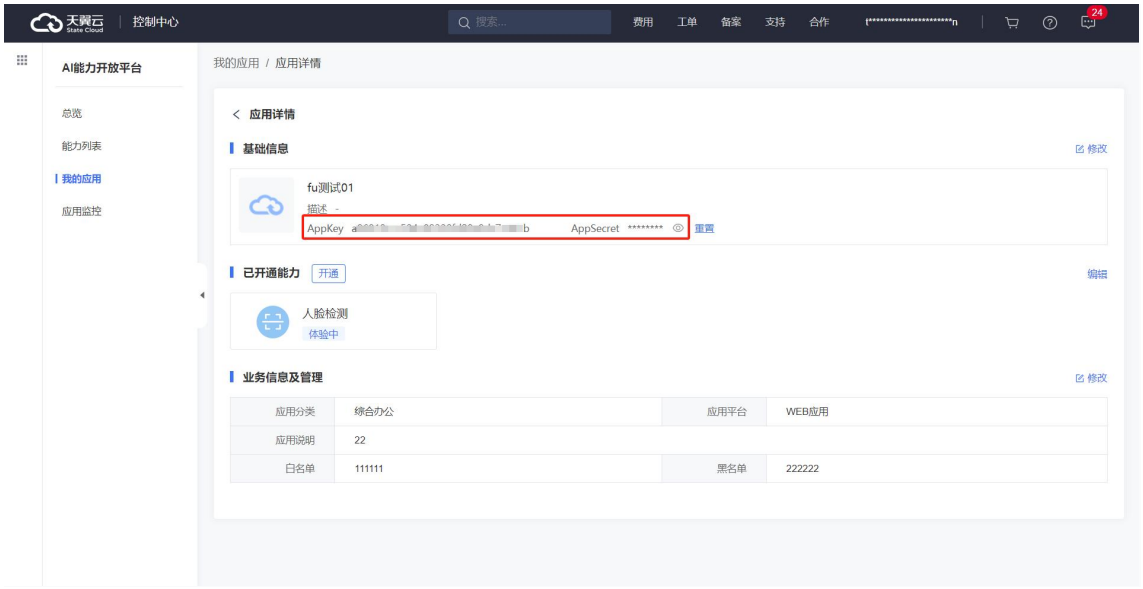
5.根据需要选择应用，点击【开通】，跳转到应用详情页。并点击【去开通】，选择需要的能力。


控制中心
费用
工单
备案
支持
合作
AI能力开放平台
总览
能力列表
我的应用
应用监控
我的应用
+ 新增应用
请输入应用名称
查询
重置
默认应用
创建时间 2024-03-01 14:50:33
开通
fu测试01
创建时间 2024-04-19 14:35:34
开通
删除
测试0001
创建时间 2024-03-25 15:11:19
开通
删除
test1231
创建时间 2024-03-20 14:00:21
开通
删除
图像分辨率
创建时间 2024-04-15 16:16:43
开通
删除
测试333
创建时间 2024-03-25 15:19:43
开通
删除
测试009
创建时间 2024-03-25 15:17:14
开通
删除
共7条
10
1


控制中心
费用
工单
备案
支持
合作
AI能力开放平台
总览
能力列表
我的应用
应用监控
我的应用 / 应用详情
应用详情
基础信息
修改
fu测试01
描述 -
AppKey
AppSecret
重置
已开通能力
开通
暂无已开通能力,去开通
业务信息及管理
修改
应用分类
综合办公
应用平台
WEB应用
应用说明
22
白名单
111111
黑名单
222222


控制中心
费用
工单
备案
支持
合作
AI能力开放平台
总览
能力列表
我的应用
应用监控
我的应用 / 应用详情
添加能力
已选择能力
人脸检测
热门能力
新上线能力
AI技术能力
人脸识别
安全生产
智能视觉生产
印刷文字识别
自然语言处理
内容审核
基础信息
请选择
是否戴口罩识别
人脸属性识别
人脸检测
人脸比对
人脸活体检测
共5条
20
1
取消
确定开通

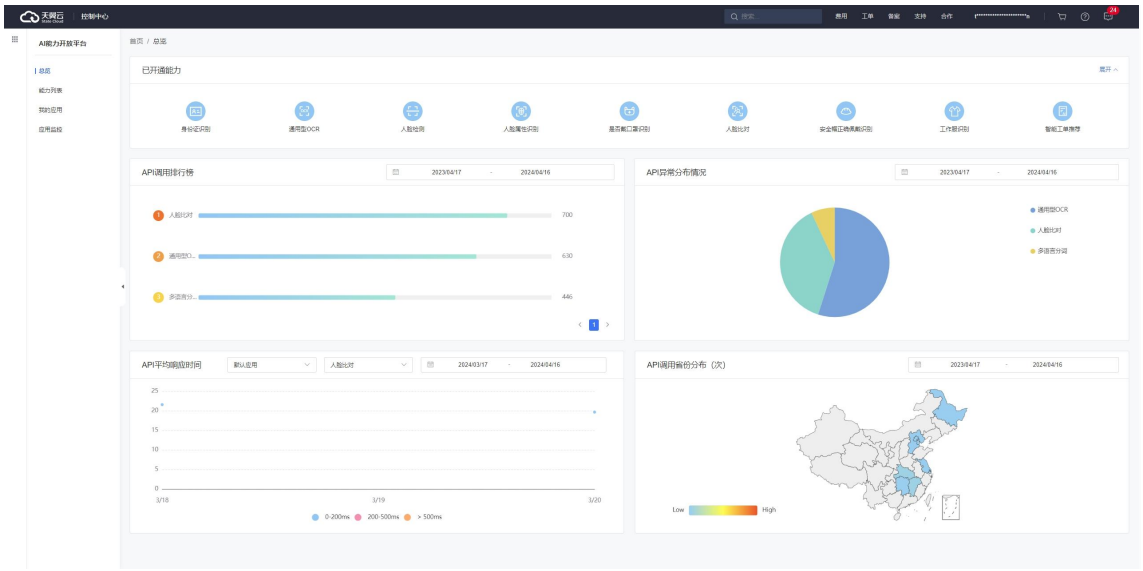
6.确认开通，并获取应用的AppKey、AppSecret。



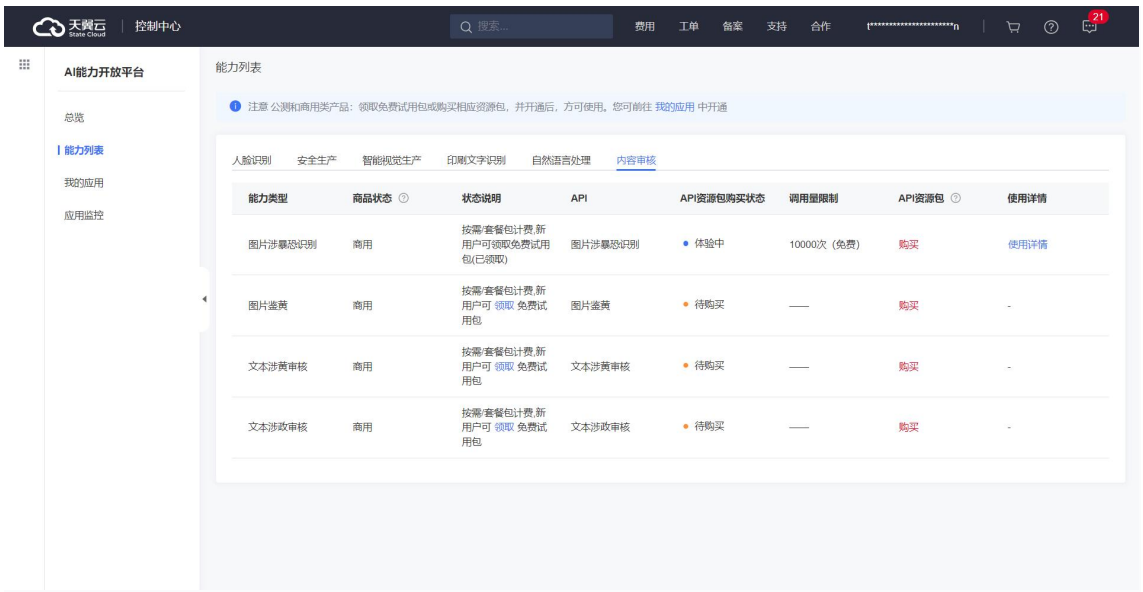
2.4 人脸识别用户控制台

1.点击产品详情页左上角的【管理控制台】，页面跳转到控制台总览页面。

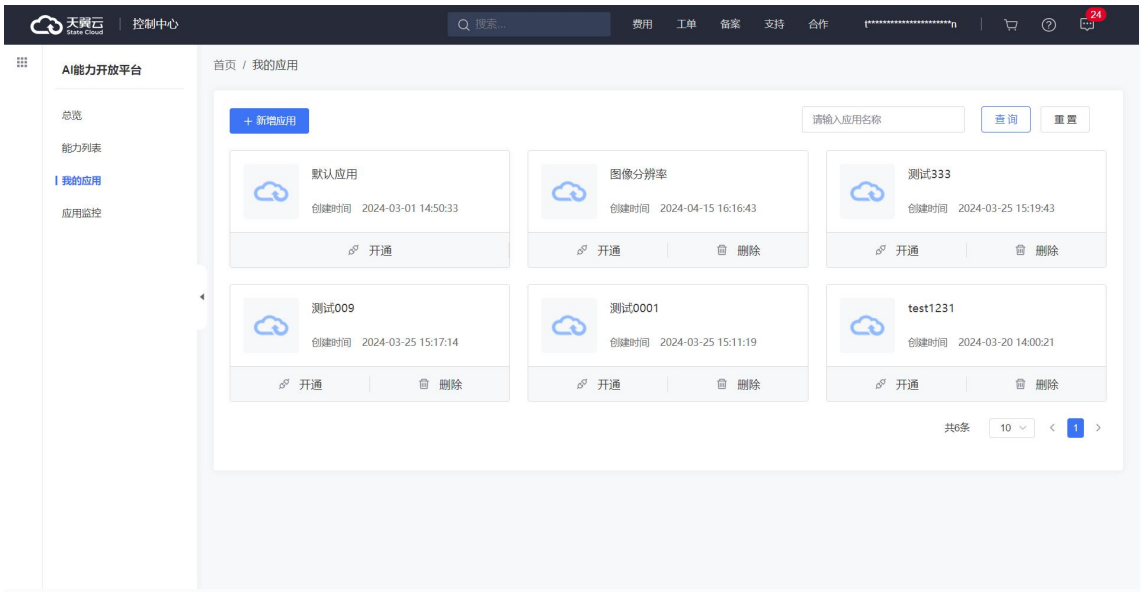
2.总览。总览页面可以查看到到已开通能力、能力调用排行榜以及平均响应时间等内容。



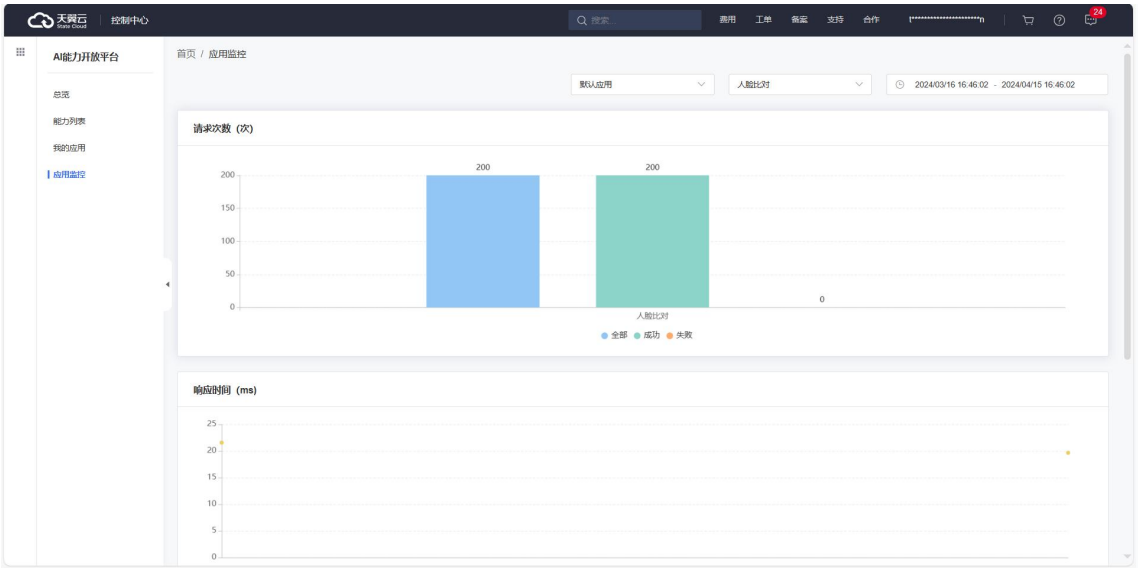
3.能力列表。点击左侧菜单栏【能力列表】，可以查看已有能力。



4.我的应用。点击左侧菜单栏【我的应用】，可以查看已经创建的应用。



5.应用监控。点击左侧菜单栏【应用监控】，可以查看所创建应用的请求次数、响应时间与请求流量。



3 用户指南

3.1 产品价格

人脸识别采用封顶资源包的方式订购:

API名称	100万次	1000万次	5000万次	1亿次
人脸检测	450元	4100元	17500元	25000元
人脸属性识别	450元	4100元	17500元	25000元
人脸比对	3100元	30400元	142500元	255000元

订购说明:

1.人脸识别产品采用封顶资源包的计费方式，商用的API服务会免费赠送调用额度供测试使用，若有已购买且在有效期内的商品资源包，则可进行抵扣。

目前人脸识别的人脸检测、人脸属性识别、人脸比对这三个能力处于商用状态，需要付费购买或者申请体验免费额度方能使用。

目前人脸识别人脸活体检测、是否戴口罩识别这两个能力处于公测阶段，公测期间0元购买不收费。

2.为满足客户不同业务使用量需求，每类API设置四档套餐，如：用户评估认为应用每年人脸属性识别的API调用大概为500万次，则可以购买五个100万次/年的调用资源包，若大概为1000万次，则可以购买一个1000万次/年的调用资源包。

3.人脸检测和人脸属性识别按照所检测的图像张数进行计数，检测一张扣减一次;

人脸比对按照API调用次数进行收费（用户在比对时会指定两张照片，其中一张可能是系统中已经存的，一张是用户新上传的，也有可能两张需比对的图像都是新上传的），调用一次接口（无论是一张还是两张）扣减一次。

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，用户无法再调用产品API。对于有效期内未使用的资源包剩余调用次数，且购买时间超过7天，本产品不支持退订。

备注：产品价格以价格发文为准。

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 API 参考

4.1API 概览

4.1.1 概述

本说明提供了人脸识别产品 API 的描述、语法、参数说明及示例等内容。

4.1.2API 概览

功能名称	描述
人脸检测	用于检测输入图像中的人脸，输出人脸位置坐标
人脸属性识别	用于检测输入图像中的人脸年龄、性别等属性
人脸比对	用于检测输入的两张图像中的人脸相似度
人脸活体检测	用于检测输入图像中的人脸是否为活体
是否戴口罩识别	用于检测输入图像中的人脸是否戴口罩

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 yyyymmddTHHMMSSZ，例如 20211221T163014Z
host	终端节点域名
appkey	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

- 待签字符串：使用规范请求和其他信息创建待签字符串。

- 计算密钥：使用 HEADER、ctyun-eop-sk、ctyun-eop-ak 来创建 Hmac 算法的密钥。
- 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。
- 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，E OP 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 yyyymmddTHHMMSSZ;言简意 赅一些，就是年月日 T 时分秒 Z

- 首先用申请获得的 ctyun-eop-sk 作为密钥，eop-date 作为数据，算出 ktime
- ktime 作为密钥，你申请来的 ctyun-eop-ak 数据，算出 kAk;
- kAk 作为密钥，eop-date 的年月日值作为数据；算出 kdate

eop-date	yyyymmddTHHMMSSZ（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“构造动态秘钥”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 成出 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）

4.1.4Python 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'
    #  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(r'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)
```

4.3 API

4.3.1 人脸检测

1、接口描述

用于检测输入图像中的人脸，输出人脸位置坐标

2、请求方法

POST

3、接口要求

- 图片大小限制：图片单张大小小于 2MB
- 图片格式限制：图片格式支持jpg/jpeg/png/bmp/gif 格式

4、请求 URL

/v1/aiop/api/2f6hqix09mv4/face/PERSON/person/detectFaceFromBase64

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	AI 应用 appkey	"562b89493b1a40e1b97ea05e50d d8170"	

请求体 body 参数

参数	是否必填	参数类型	说明	示例	下级对象
imageContent	是	string	传入图片的base64 编码，图 片使用常规的 base64 编码方 式，编码后，不包含前缀，剔 除前缀例如 “”	-	

6、请求代码示例

```
Curl -X POST "https://ai-global.ctapi.ctyun.cn/v1/aiop/api/2f6hqix09mv4/face/PERSON/person/detectFaceFromBase64"
-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 '{"imageContent":"AAAAAAAAA...."}'
```

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	string	返回状态，返回 0 表示成功，返回错误代码参考下面的错误	"0"	

参数	是否必填	参数类型	说明	示例	下级对象
			代码列表。		
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	返回的人脸检测结果对象		result

表 result

参数	是否必填	参数类型	说明	示例	下级对象
face_num	是	int	图片中人脸的数量	-	
face_list	是	list	每个人脸的详细信息	-	
face_list[].face_loaction	是	object	人脸所处位置	-	

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	string	错误码，放置API对应的错误码	"4101"	
message	是	string	请求失败时返回值固定为"error"	"error"	
details	是	string	返回对应的错误信息	"请求内容错误"	

8、返回值示例

请求成功返回值示例

```
{
    "code": 0,
    "message": "success",
    "result":{
        "face_num" : 1,
        "face_list" : [{
            "face_location" : {
                "top" : 36,
                "left" : 48,
                "width" : 58,
                "height" : 72
            }
        }]
    }
}
```

请求失败返回值示例

```
{
    "code": "4101",
```

```
    "message": "error",
    "details": "请求内容错误"
}
```

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
4101	请求内容错误	传入内容为空，或者传入的参数名错误
4102	请求参数格式错误	参数格式不满足要求，如请求参数字段类型错误等
4103	图片大小超过2M	图片大小超过2M
4104	图片解码失败	图片为空，base64 编码内容有误，或图片格式不支持
4105	未检测到人脸	上传图片中不包含人脸

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 人脸属性识别

1、接口描述

用于检测输入图像中的人脸年龄、性别等属性

2、请求方法

POST

3、接口要求

- 图片大小限制：图片单张大小小于 2MB
- 图片格式限制：图片格式支持jpg/jpeg/png/bmp/gif 格式

4、请求 URL

/v1/aiop/api/2f6hw5o5t7gg/face/PERSON/person/detectAgeGenderFromBase64

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	AI 应用 appkey	"562b89493b1a40e1b97ea05e50dd8 170"	

请求体 body 参数

参数	是否必填	参数类型	说明	示例	下级对象
imageContent	是	string	传入图片的base64编码，图片使用常规的base64编码方式，编码后，不包含前缀，剔除前缀例如“”	-	

6、请求代码示例

```
Curl -X POST "https://ai-
global.ctapi.ctyun.cn/v1/aiop/api/2f6hw5o5t7gg/face/PERSON/person/detectAgeGenderFromBase64"
-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 '{"imageContent":"AAAAAAAAA...."}'
```

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	string	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	返回的人脸属性识别结果对象		result

表 result

参数	是否必填	参数类型	说明	示例	下级对象
face_num	是	int	图片中人脸的数量	-	

参数	是否必填	参数类型	说明	示例	下级对象
face_list	是	list	每个人脸的详细信息	-	
face_list[].face_loaction	是	object	人脸所处位置	-	
face_list[].gender	是	string	人脸的性别属性	-	
face_list[].age	是	string	人脸的年龄属性	-	

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	string	错误码，放置API对应的错误码	"4101"	
message	是	string	请求失败时返回值固定为"error"	"error"	
details	是	string	返回对应的错误信息	"请求内容错误"	

8、返回值示例

请求成功返回值示例

```
{
```

```
    "code": 0,
    "message": "success",
    "result":{
        "face_num" : 1,
        "face_list" : [{
            "face_location" : {
                "top" : 36,
                "left" : 48,
                "width" : 58,
                "height" : 72
            },
            "gender" : "Male",
            "age" : "23"
        }]
    }
}
```


请求失败返回值示例


```
{
    "code": "4101",
    "message": "error",
    "details": "请求内容错误"
}
```


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
4101	请求内容错误	传入内容为空，或者传入的参数名错误
4102	请求参数格式错误	参数格式不满足要求，如请求参数字段类型错误等

错误码	错误信息	错误描述
4103	图片大小超过2M	图片大小超过2M
4104	图片解码失败	图片为空，base64编码内容有误，或图片格式不支持
4105	未检测到人脸	上传图片中不包含人脸

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 人脸比对

1、接口描述

用于检测输入的两张图像中的人脸相似度

2、请求方法

POST

3、接口要求

- 图片大小限制：图片单张大小小于 2MB
- 图片格式限制：图片格式支持jpg/jpeg/png/bmp/gif 格式

4、请求 URL

/v1/aiop/api/2f7awxekgvls/face/compare/PERSON/person/compareFromBase64

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	AI 应用 appkey	"562b89493b1a40e1b97ea05e50d d8170"	

请求体 body 参数

参数	是否必填	参数类型	说明	示例	下级对象
img1Base64	是	string	人脸比对的第一张图片，图片使用常规的 base64 编码方式，编码后，不包含前缀，剔除前缀例 如””	-	
img2Base64	是	string	人脸比对的第二张图片，图片使用常规的 base64 编码方式，编码后，不包含前缀，剔除前缀例 如””	-	

6、请求代码示例

```
Curl -X POST "https://ai-
global.ctapi.ctyun.cn/v1/aiop/api/2f7awxekgvls/face/compare/PERSON/person/compareFromBase64"
-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 '{"img1Base64":"AAAAAAAAA....","img2Base64":"BBBBBBBBBB...."}'
```

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	string	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	float	两张人脸的相似度	0.99999	result

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	string	错误码，放置API对应的错误码	"4101"	
message	是	string	请求失败时返回值固定为"error"	"error"	
details	是	string	返回对应的错误信息	"请求内容错误"	

8、返回值示例

请求成功返回值示例

```
{
    "code": 0,
    "message": "success",
    "result": 0.99999
}
```

请求失败返回值示例

```
{
    "code": "4101",
    "message": "error",
    "details": "请求内容错误"
}
```

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。

错误码	错误信息	错误描述
4101	请求内容错误	传入内容为空，或者传入的参数名错误
4102	请求参数格式错误	参数格式不满足要求，如请求参数字段类型错误等
4103	图片一大小超过2M	图片一大小超过2M
	图片二大小超过2M	图片二大小超过2M
4104	图片一解码失败	图片为空，base64 编码内容有误，或图片格式不支持
	图片二解码失败	图片为空，base64 编码内容有误，或图片格式不支持
4105	图片一未检测到人脸	上传图片一中不包含人脸
4106	图片二未检测到人脸	上传图片二中不包含人脸

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 人脸活体检测

1、接口描述

用于检测输入图像中的人脸是否为活体

2、请求方法

POST

3、接口要求

- 图片大小限制：图片单张大小小于 2MB
- 图片格式限制：图片格式支持jpg/jpeg/png/bmp/gif 格式

4、请求 URL

/v1/aiop/api/2hfksnibjaos/face-fas-action/person/detectFasFromBase64

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	AI 应用	"562b89493b1a40e1b97ea05e50dd8170"	

参数	是否必填	参数类型	说明	示例	下级对象
			appkey		

请求体 body 参数

参数	是否必填	参数类型	说明	示例	下级对象
imageContent	是	string	传入图片的 base64 编码，图 片使用常规的 base64 编码方 式，编码后，不包含前缀，剔 除前缀例如 “”	-	

6、请求代码示例

```
Curl -X POST "https://ai-global.ctapi.ctyun.cn/v1/aiop/api/2hfksnibjaos/face-fas-action/person/detectFasFromBase64"

-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 '{"imageContent":"AAAAAAAAA...."}'
```

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	string	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	返回的人脸活体检测结果对象		result

表 result

参数	是否必填	参数类型	说明	示例	下级对象
face_num	是	int	图片中人脸的数量	-	

参数	是否必填	参数类型	说明	示例	下级对象
face_list	是	list	每个人脸的详细信息	-	
face_list[].face_loaction	是	object	人脸所处位置	-	
face_list[].FaceAntiSpoofing	是	string	活体人脸/非活体人脸/人脸清晰度差，无法判断	-	

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	string	错误码，放置API对应的错误码	"4101"	
message	是	string	请求失败时返回值固定为"error"	"error"	
details	是	string	返回对应的错误信息	"请求内容错误"	

8、返回值示例

请求成功返回值示例

```
{
    "code": 0,
    "message": "success",
    "result":{
        "face_num" : 1,
        "face_list" : [{
            "face_location" : {
                "top" : 36,
                "left" : 48,
                "width" : 58,
                "height" : 72
                },
            "FaceAntiSpoofing" : "活体人脸/非活体人脸/人脸清晰度较差，无法判断",
        }]
    }
}
```

请求失败返回值示例

```
{
    "code": "4101",
    "message": "error",
    "details": "请求内容错误"
}
```

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
4101	请求内容错误	传入内容为空，或者传入的参数名错误
4102	请求参数格式错误	参数格式不满足要求，如请求参数字段类型错误等
4103	图片大小超过2M	图片大小超过2M
4104	图片解码失败	图片为空，base64 编码内容有误，或图片格式不支持
4105	未检测到人脸	上传图片中不包含人脸
5000	服务内部错误	接口服务出现未知错误

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。

4.3.5 是否戴口罩识别

1、接口描述

用于检测输入图像中的人脸是否戴口罩

2、请求方法

POST

3、接口要求

- 图片大小限制：图片单张大小小于2MB
- 图片格式限制：图片格式支持jpg/jpeg/png/bmp/gif格式

4、请求 URL

/v1/aiop/api/2f6hycj3a9z4/face/PERSON/person/detectMaskFromBase64

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	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
imageContent	是	string	传入图片的base64 编码，图 片使用常规的 base64 编码方 式，编码后，不包含前缀，剔 除前缀例如 “”	-	

6、请求代码示例

```
Curl -X POST "https://ai-global.ctapi.ctyun.cn/v1/aiop/api/2f6hycj3a9z4/face/PERSON/person/detectMaskFromBase64"
-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 '{"imageContent":"AAAAAAAAA...."}'
```

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	string	返回状态，返回 0	"0"	

参数	是否必填	参数类型	说明	示例	下级对象
			表示成功，返回错误代码参考下面的错误代码列表。		
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	返回的是否戴口罩识别结果对象		result

表 result

参数	是否必填	参数类型	说明	示例	下级对象
face_num	是	int	图片中人脸的数量	-	
face_list	是	list	每个人脸的详细信息	-	
face_list[].face_loaction	是	map	人脸所处位置	-	

face_list[].Mask	是	bool	是否佩戴口罩	-	

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	string	错误码，放置API对应的错误码	"4101"	
message	是	string	请求失败时返回值固定为"error"	"error"	
details	是	string	返回对应的错误信息	"请求内容错误"	

8、返回值示例

请求成功返回值示例

```
{
    "code": 0,
    "message": "success",
    "result":{
    "face_num" : 1,
    "face_list" : [{
        "face_location" : {
            "top" : 36,
            "left" : 48,
            "width" : 58,
            "height" : 72
        },
```

```
        "Mask" : false
    }]
    }
}
```

请求失败返回值示例

```
{
    "code": "4101",
    "message": "error",
    "details": "请求内容错误"
}
```

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
4101	请求内容错误	传入内容为空，或者传入的参数名错误
4102	请求参数格式错误	参数格式不满足要求，如请求参数字段类型错误等
4103	图片大小超过2M	图片大小超过2M
4104	图片解码失败	图片为空，base64 编码内容有误，或图片格式不支持
4105	未检测到人脸	上传图片中不包含人脸

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。

4.4 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

人脸识别支持哪些计费方式?

计费方式：目前我们只提供封顶资源包的计费方式，资源包有效期一年，资源包价格表如下：

API名称	100万次	1000万次	5000万次	1亿次
人脸检测	450元	4100元	17500元	25000元
人脸属性识别	450元	4100元	17500元	25000元
人脸比对	3100元	30400元	142500元	255000元

注：表格价格仅供参考，具体购买价格以价格发文为准

调用量的扣费顺序是?

调用量的扣减顺序为：免费试用包-付费资源包，购买资源包后，将按照资源包下单顺序抵扣额度。
如果当前扣减额度为付费资源包，而后又领取了免费试用包，则当前付费资源包使用完毕后，优先抵扣免费试用包，然后按照资源包下单顺序抵扣。
在购买之前，可先领取免费试用包，待体验过后再自行购买。

人脸识别资源包可以转移给到别人的账号么?

该产品目前不支持转移，建议购买前评估真实调用量，资源包可以叠加购买。
若后期支持转移功能，我们会发公告通知，请您留心注意。

5.2 购买类

人脸识别是否支持续订?

资源包是一次性产品，不支持标准续订操作，可通过连续购买资源包实现续订相同的功能。
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，按照次数包下单顺序抵扣额度。

资源包买错了可以退款吗?

退订说明：人脸识别产品服务开通后7天内如无调用接口支持退订，退订后即可关闭。

- 退订地址：我的—费用中心—订单管理—退订管理。
- 另外，您可通过天翼云官网工单或者客服电话【400-810-9889】沟通申请退款，款项会原路退回。

人脸识别服务是否支持代金券付款?

支持。操作方式：支持代金券付款，在卡券管理选择相应的代金券进行使用。

操作地址：我的—费用中心—卡券管理。

人脸识别服务是否支持试用?

在天翼云官网订购，支持有限次数的资源包试用。根据能力不同，试用次数也有所不同。可以通过天翼云官网工单或者客服电话【400-810-9889】沟通试用。

人脸识别服务也可以领取产品免费额度进行试用，支持领取免费额度的部分产品如下图所示。

API名称	免费额度	具体说明
人脸检测	10000次	以单个API为统计维度，有效期从领取之日起一年内有效，过期作废
人脸属性识别	10000次	以单个API为统计维度，有效期从领取之日起一年内有效，过期作废
人脸比对	10000次	以单个API为统计维度，有效期从领取之日起一年内有效，过期作废

5.3 操作类

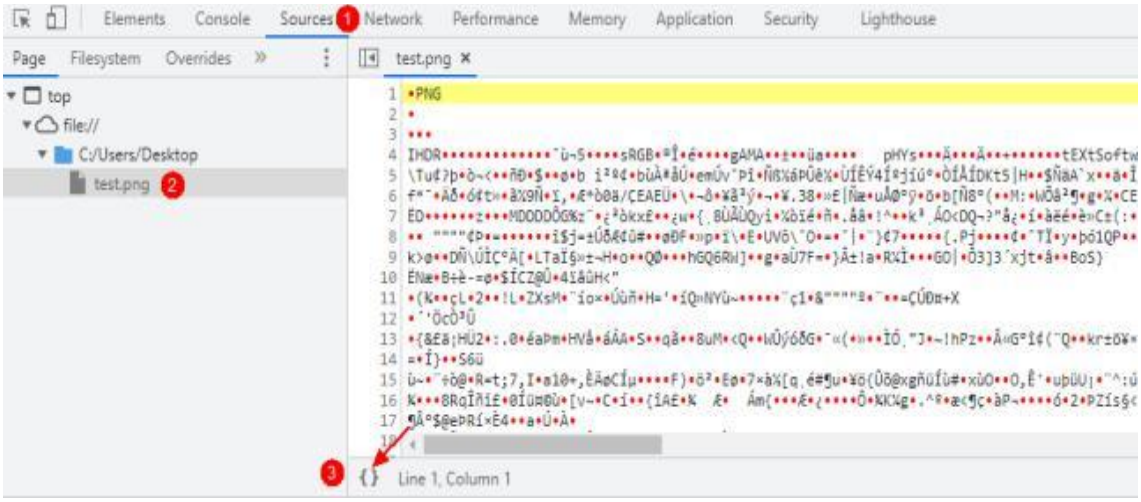
人脸识别在什么时候进入可使用状态?

A：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。

2.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。



3.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。

人脸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。

人脸识别的并发是多少?

默认5个并发。默认支持5个QPS，建议您在程序中可以进行一定的请求限制，避免收到大量限流报错。

如果因业务需要QPS超过5个，请提前线下咨询沟通再购买下单。

注意：如果您的程序在失败时有重试机制，当您扩大并发量后接口返回错误码时，请不要重试，否则可能加重限流报错情况。

5.4 使用限制

人脸识别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jpeg、png、bmp、jif。

人脸识别是 HTTP GET 请求还是 HTTP POST 请求?
人脸识别产品以API的方式提供服务，支持HTTP POST请求。

HTTP POST提供了加密传输、身份验证和授权以及请求参数验证和过滤等机制，从而可以更好的保障用户的数据安全。

人脸比对的 API 怎么用？上传几张图片?
人脸对比的 API 主要是对比两张图片中的人脸是否为同一人，因此上传至少上传 2 张人脸图片（即两张 图片为一组，可以为多组，进行批量人脸的对比）。

人脸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。

HTTP请求的详细信息可以查看[构造请求]。

请求地址：产品文档—API参考—如何调用API—构造请求。

人脸检测的 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。

API接口的详细信息可以查看产品API文档。

请求地址：产品文档—API参考—API—对应产品（如人脸检测）。