



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	4
1.1 产品定义	4
1.2 产品优势	7
1.3 功能特性	7
1.4 应用场景	9
1.5 术语解释	10
2 计费说明	12
2.1 计费模式	12
2.2 产品续订	15
2.3 产品退订	16
3 快速入门	18
3.1 入门指引	18
3.2 服务创建	18
3.3 入门操作	19
3.4 入门实践	21
4 用户指南	22
4.1 产品购买	22
4.2 创建应用及开通应用	25
4.3 用户控制台	30
4.4 API调用指南	32

5 最佳实践	37
5.1 人脸活体检测	37
5.2 人脸比对	40
6 API参考	43
6.1 调用前必知	43
6.2 如何调用API	45
6.3 API	60
7 常见问题	87
7.1 计费类	87
7.2 购买类	89
7.3 操作类	90
7.4 使用限制	91
7.5 安全相关	93

1 产品简介

1.1 产品定义

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（应用程序编程接口）的方式提供给用户，用户通过实时访问和调用API获取人脸处理结果，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。

当前人脸识别提供了以下子服务：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

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，给出人脸的数量、位置等信息，方便用户后续应用。



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。



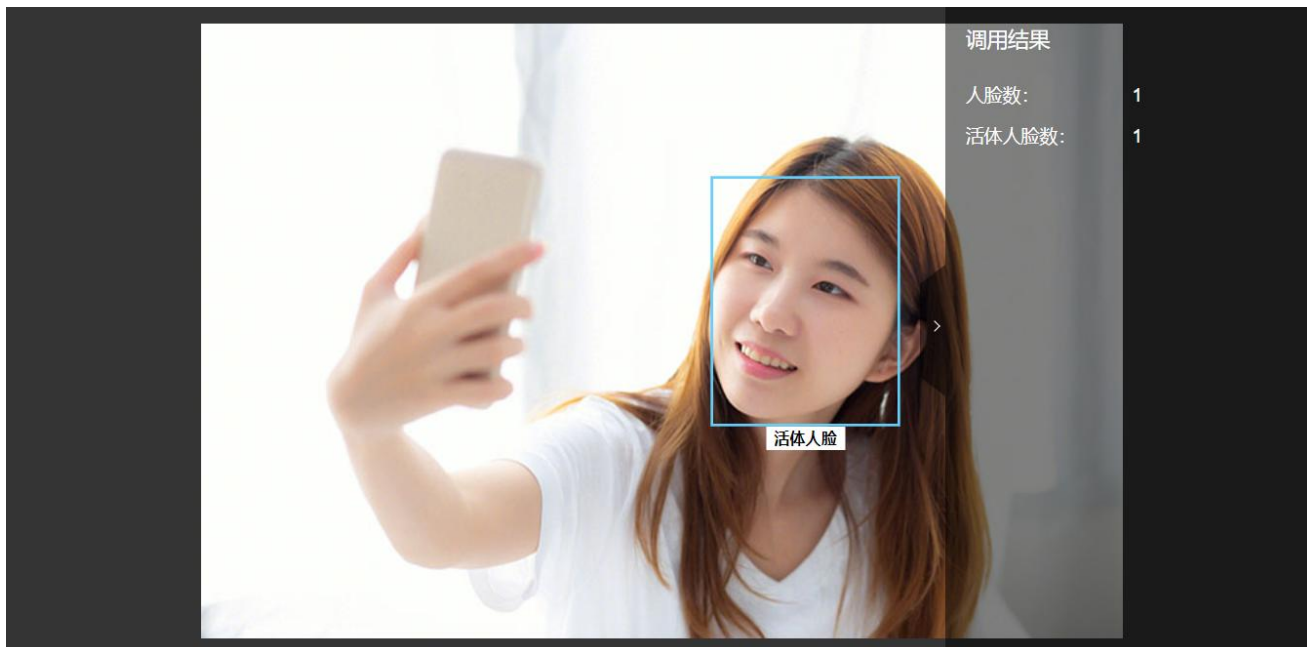
是否戴口罩识别

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。



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。



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。



1.2 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

1.3 功能特性

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。
- 具备出色的多人脸检测能力，系统在处理不同角度和光照条件下仍然表现出色。对于一定程度的遮挡，也能有效应对。
- 跨平台兼容性使其适用于各种硬件和操作系统，并通过可配置的安全选项确保数据的保密性。

人脸属性识别

- 在人脸检测基础上，分析检测后的人脸的数量、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。
- 产品设计考虑了多样的场景，包括不同角度、光照条件以及一定程度的遮挡，保证人脸属性识别的准确性。
- 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

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。
- 对于公共卫生各种场景下，如公共交通、医疗机构和商业场所，进行有效的口罩佩戴监测。
- 产品支持多人脸检测，具备角度和光照鲁棒性，并可处理一定程度的遮挡。

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，有效防止静态图像等非活体攻击。
- 通过分析眼部活动、面部表情和其他活体迹象，确保在验证过程中只有真实的活体能够通过。
- 适用于各种应用场景，包括金融、安防和身份验证领域，具备高实时性和可靠性，有助于提升安全性和防范欺诈。

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。
- 支持多种场景下的人脸比对，包括不同角度、光照条件和表情变化。

- 跨平台兼容性使其适用于各种硬件和操作系统，并通过可配置的安全选项确保数据的保密性。

1.4 应用场景

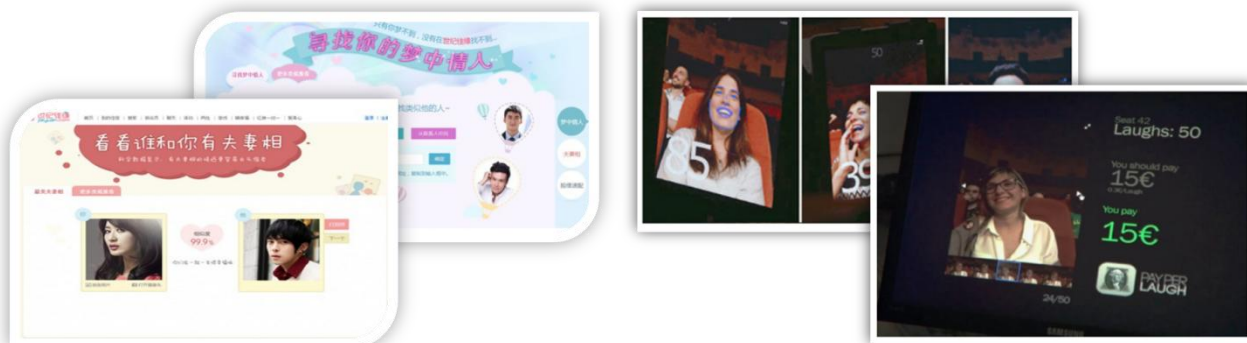
门禁人脸识别

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。人脸检测内置于智能门禁系统，能准确识别进来人员人脸信息，提供出行方便的服务，同时有效的杜绝外来无关人员进入，保证环境安全。



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。



商业银行

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。



1.5 术语解释

人脸检测

人脸检测算法的原理简单来说是一个“扫描”加“判定”的过程。在整个图像范围内扫描，再逐个判定候选区域是否是人脸的过程。

人脸属性识别

识别出人脸的性别、年龄、姿态、表情等属性值的一项技术，可以自动识别摄像头视野中人物的性别、年龄等特征并标注出来。

人脸活体

判断人脸图像是真人还是其他假体,确保在验证过程中只有真实的活体能够通过。

活体攻击

翻拍人脸图片，或使用印刷人脸照片检测，判断是否具备生物特征，提高验证的安全性。

人脸比对

人脸比对算法的输入是两个人脸特征，人脸特征由前面的人脸提取特征算法获得，输出两个特征之间的相似度，衡量两个人脸之间相似度。

人脸比对1:1，判定两个人脸图是否为同一人的算法。它的输入是两个人脸特征，通过人脸比对获得两个人脸特征的相似度，通过与预设的阈值比较来验证这两个人脸特征是否属于同一人。

人脸配准

定位出人脸五官关键点坐标。基于人脸检测的坐标框，按某种事先设定规则将人脸区域抠取出来，缩放到固定尺寸，然后进行关键点位置的计算。

提取特征

将一张人脸图像转化为可以表征人脸特点的特征，具体表现形式为一串固定长度的数值。首先将五官关键点坐标进行旋转、缩放等等操作来实现人脸对齐，然后再提取特征并计算出数值串。

人脸聚类

将一个集合内的人脸根据身份进行分组的算法。在没有进行人工身份标注前，只知道分到一个组的人脸是属于同一个身份，但不知道确切身份。

特征脸

从统计的观点，寻找人脸图像分布的基本元素，即人脸图像样本集协方差矩阵的特征向量，以此近似地表征人脸图像。这些特征向量称为特征脸。

弹性图匹配

一种基于几何特征和对灰度分布信息进行小波纹理分析相结合的识别算法。

神经网络方法

通过学习的过程获得对人脸规律和规则的隐性表达，将人脸作为一个一维向量输入。

2 计费说明

2.1 计费模式

计费说明如下：

人脸识别产品以资源包计费，部分产品已经商用。

- 人脸识别产品采用封顶资源包的计费方式，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，需要付费购买使用。
- 目前人脸识别的人脸活体检测、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。
- 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。
- 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），调用一次接口扣减一次。
- 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AI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。

调用量的抵扣顺序

- 调用量的抵扣顺序为：购买资源包后，将按照资源包下单顺序抵扣额度。

人脸检测计费详情

付费资源包价格表

产品名称	调用量规格	调用有效期	单价
人脸检测	100万次	1年	450元
人脸检测	1,000万次	1年	4,100元
人脸检测	5,000万次	1年	17,500元
人脸检测	10,000万次	1年	25,000元

人脸属性识别计费详情

付费资源包价格表

产品名称	调用量规格	调用有效期	单价
人脸属性识别	100万次	1年	450元
人脸属性识别	1,000万次	1年	4,100元
人脸属性识别	5,000万次	1年	17,500元
人脸属性识别	10,000万次	1年	25,000元

人脸比对计费详情

付费资源包价格表

产品名称	调用量规格	调用有效期	单价
------	-------	-------	----

产品名称	调用量规格	调用有效期	单价
人脸比对	100万次	1年	3,100元
人脸比对	1,000万次	1年	30,400元
人脸比对	5,000万次	1年	142,500元
人脸比对	10,000万次	1年	255,000元

是否戴口罩识别计费详情

付费资源包价格表

产品名称	调用量规格	调用有效期	单价
是否戴口罩识别	100万次	1年	0元
是否戴口罩识别	1,000万次	1年	0元
是否戴口罩识别	5,000万次	1年	0元
是否戴口罩识别	10,000万次	1年	0元

人脸活体检测计费详情

付费资源包价格表

产品名称	调用量规格	调用有效期	单价
人脸活体检测	100万次	1年	0元
人脸活体检测	1,000万次	1年	0元
人脸活体检测	5,000万次	1年	0元

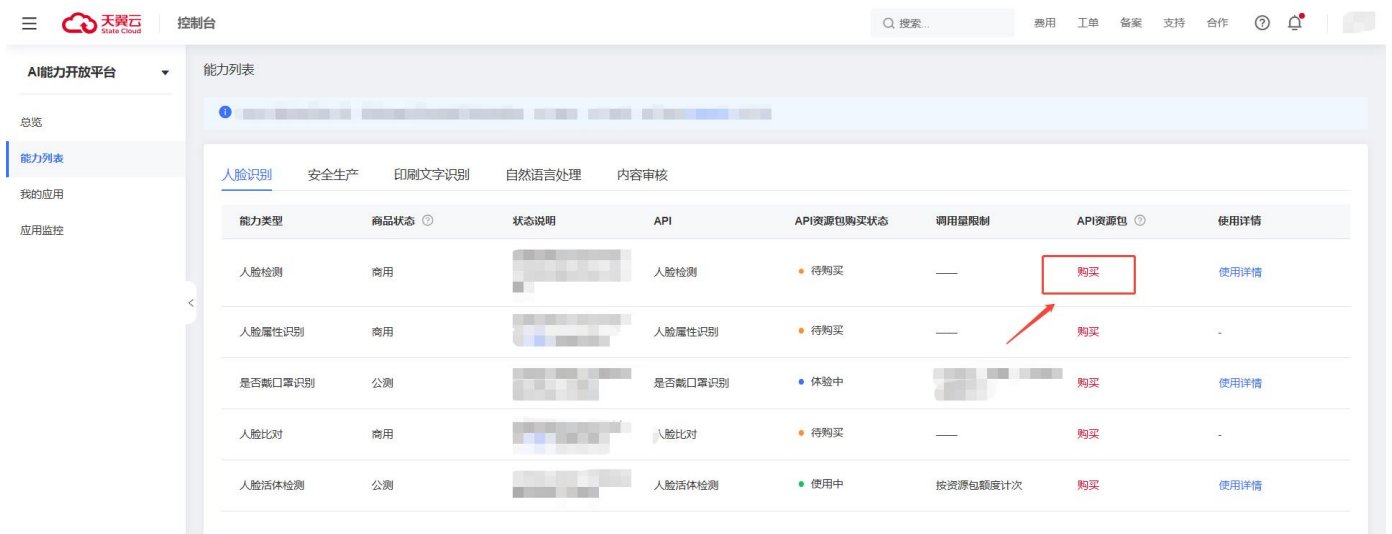
产品名称	调用量规格	调用有效期	单价
人脸活体检测	10,000万次	1年	0元

注意:表格价格仅供参考，具体购买价格以价格发文为准。

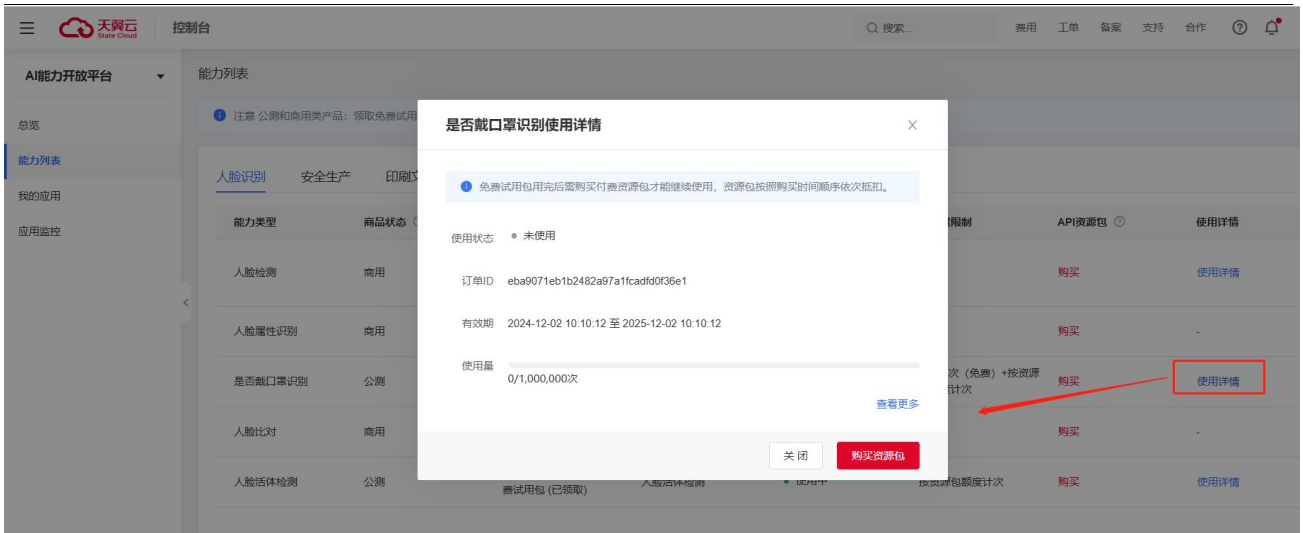
2.2 产品续订

续订和续费说明：

- 续订说明：当已订购的资源包订单即将到期或即将用完时，可订购新的资源包。
- 续费模式：手动续费，产品价格与订购价格保持一致。
- 使用详情：在已购买的产品使用详情中，可再次购买次数包，也可查看使用状态、有效期、使用量。



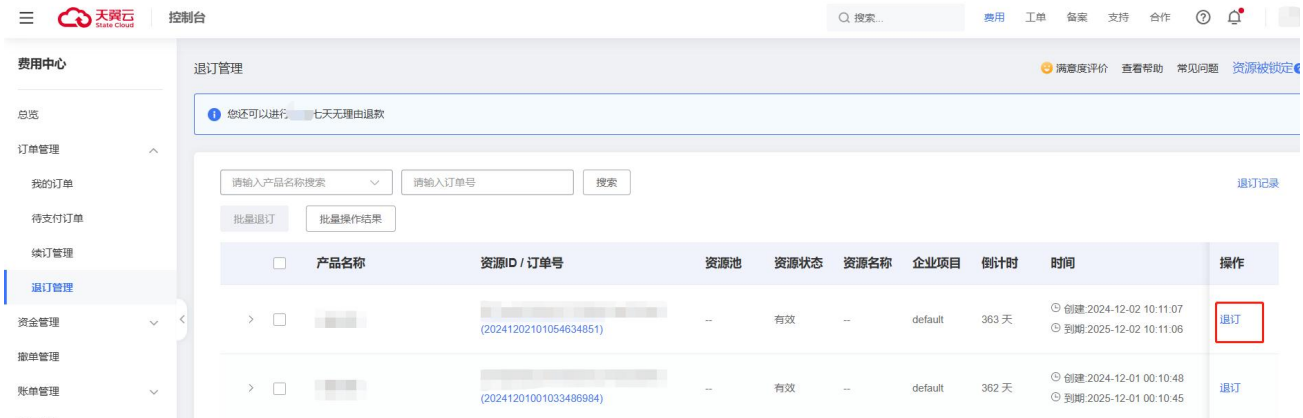
能力类型	商品状态	状态说明	API	API资源包购买状态	调用量限制	API资源包	使用详情
人脸检测	商用	[模糊]	人脸检测	待购买	—	购买	使用详情
人脸属性识别	商用	[模糊]	人脸属性识别	待购买	—	购买	-
是否戴口罩识别	公测	[模糊]	是否戴口罩识别	体验中	[模糊]	购买	使用详情
人脸比对	商用	[模糊]	人脸比对	待购买	—	购买	-
人脸活体检测	公测	[模糊]	人脸活体检测	使用中	按资源包额度计次	购买	使用详情



2.3 产品退订

退订说明：人脸识别产品支持7天未使用的情况下申请退订。

- 退订地址：控制台—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。



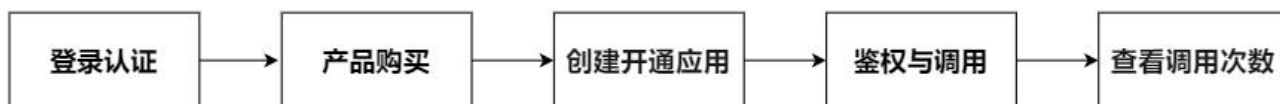
3 快速入门

3.1 入门指引

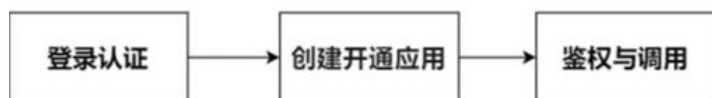
人脸识别服务，是基于人的脸部特征信息，利用计算机对人脸图像进行处理、分析和理解，进行身份识别的一种智能服务。

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



3.2 服务创建

步骤一：成为天翼云用户

注册天翼云账号

打开天翼云官网，点击右上角【免费注册】按照操作提示完成账号注册。

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。

- 进入账号中心页面，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。
- 如果您是企业用户，推荐进行企业认证，以便获取更多便利。更多实名认证操作信息，请参见实名认证。

步骤二：产品购买

产品购买详细步骤请查看产品购买。



步骤三：创建应用及开通应用

创建应用及开通应用操作步骤请查看创建应用及开通应用。



3.3 入门操作

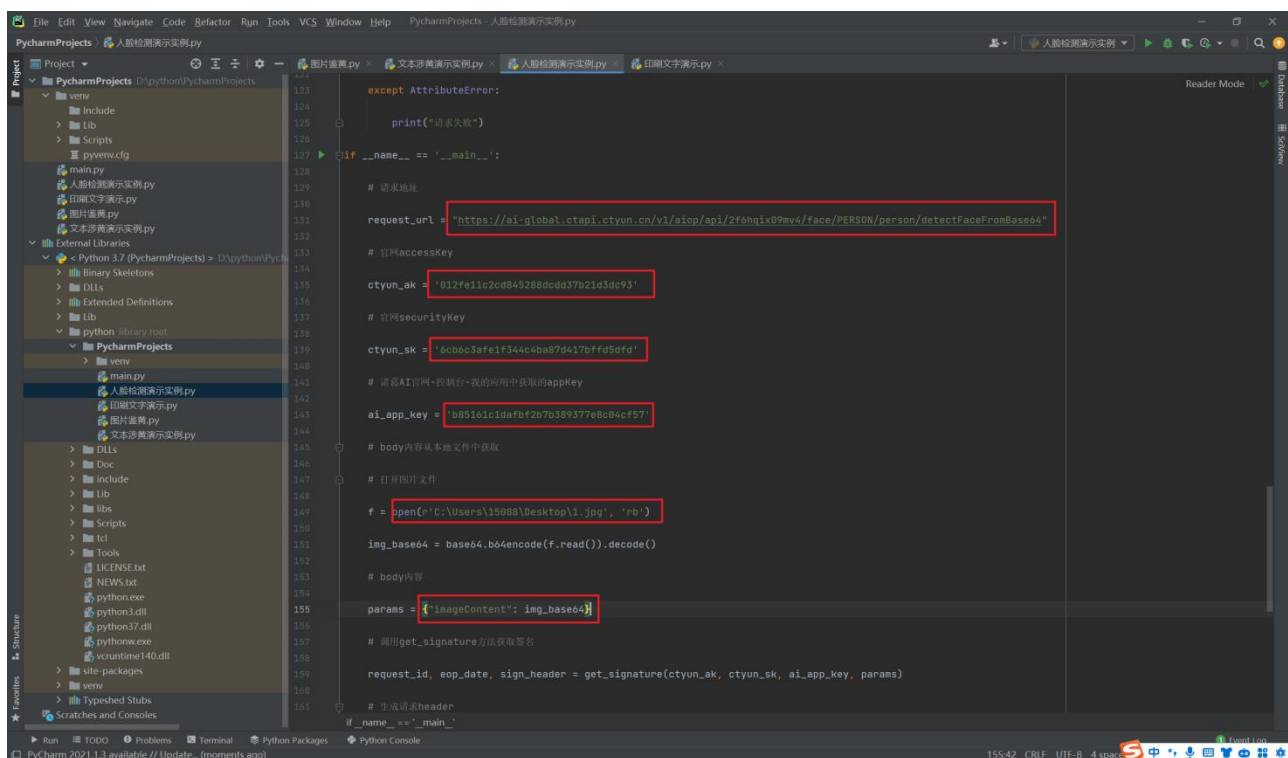
步骤一：鉴权及调用API

以人脸检测为例，更多详情请查看认证鉴权。

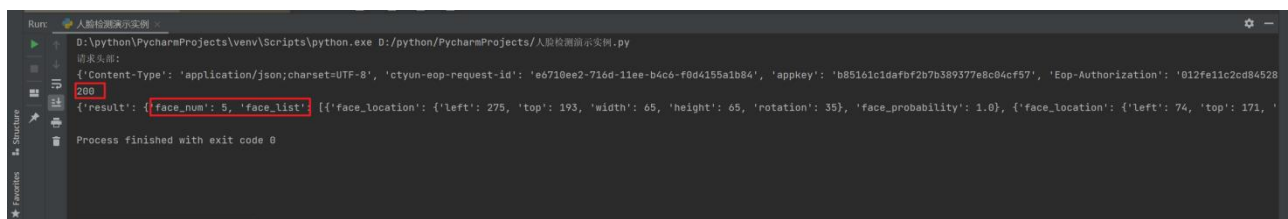
1. 选择合适自己的开发工具，在这里用Python为例，点击Python3调用示例，选择复制全部代码；



2. 粘贴刚复制的代码块，接下来替换URL、AccessKey和SecurityKey、AppKey、入参类型；



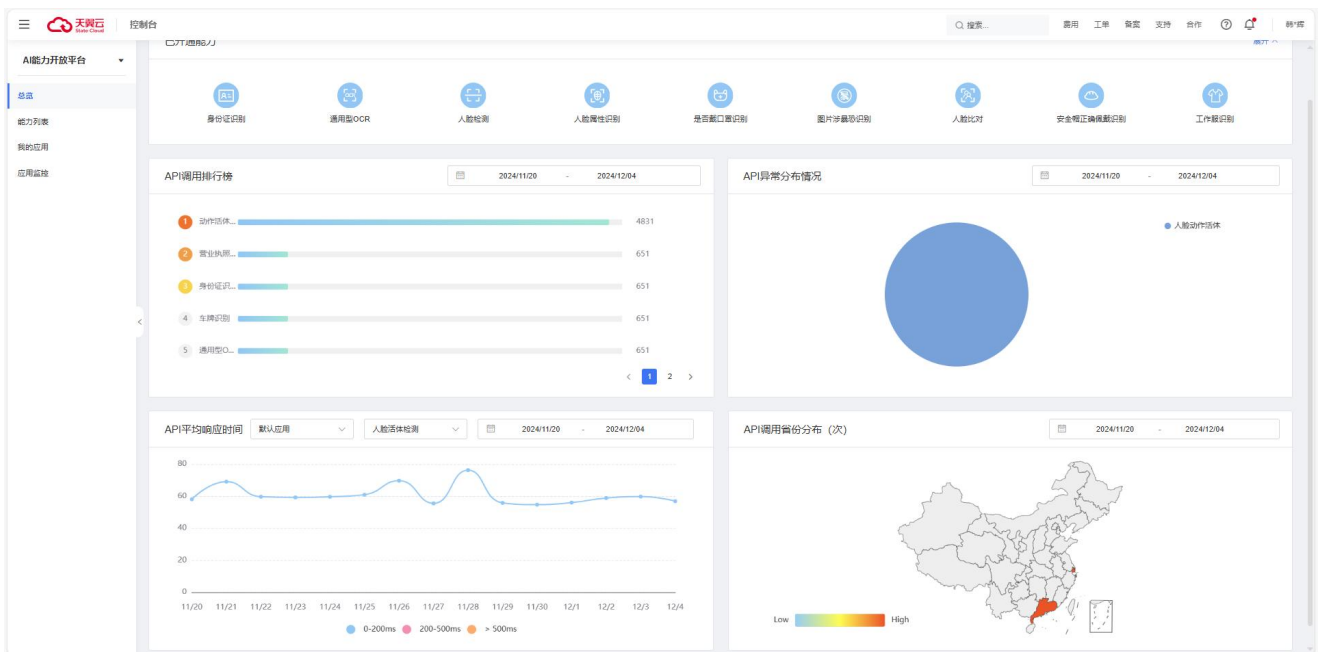
3. 当状态码为200，则表示调用成功，下方显示成功数量、失败数量、审核结果等信息。



步骤二：查看调用次数

打开控制台，总览页面可以查看到已开通能力、能力调用排行榜以及平均响应时间等内容，更多操作台详情请查看用户控制台。

- 已开通能力：已经加入到应用并已开通成功的能力。
- 能力调用排行榜：规定时间内，已调用的原子能力按照调用次数进行排序。
- 能力异常分布情况：规定时间内，原子能力异常分布的圆形统计图。
- 平均响应时间：选取应用分组原子能力后在规定时间内，查看其平均响应时间。
- 能力调用省份分布：规定时间内，按照省份分布，鼠标悬浮可显示次数。



3.4 入门实践

产品实践名称	实践描述	章节目录
人脸活体检测	通过人脸活体检测业务实例，详细介绍业务接入的基本流程和操作方案建议，消除客户业务操作实践困惑。	参考下面最佳实践章节
人脸比对	通过人脸比对业务实例，详细介绍业务接入的基本流程和操作方案建议，消除客户业务操作实践困惑。	参考下面最佳实践章节

4 用户指南

4.1 产品购买

购买操作步骤

步骤一：成为天翼云用户

注册天翼云账号

打开天翼云官网，点击右上角【免费注册】按照操作提示完成账号注册。

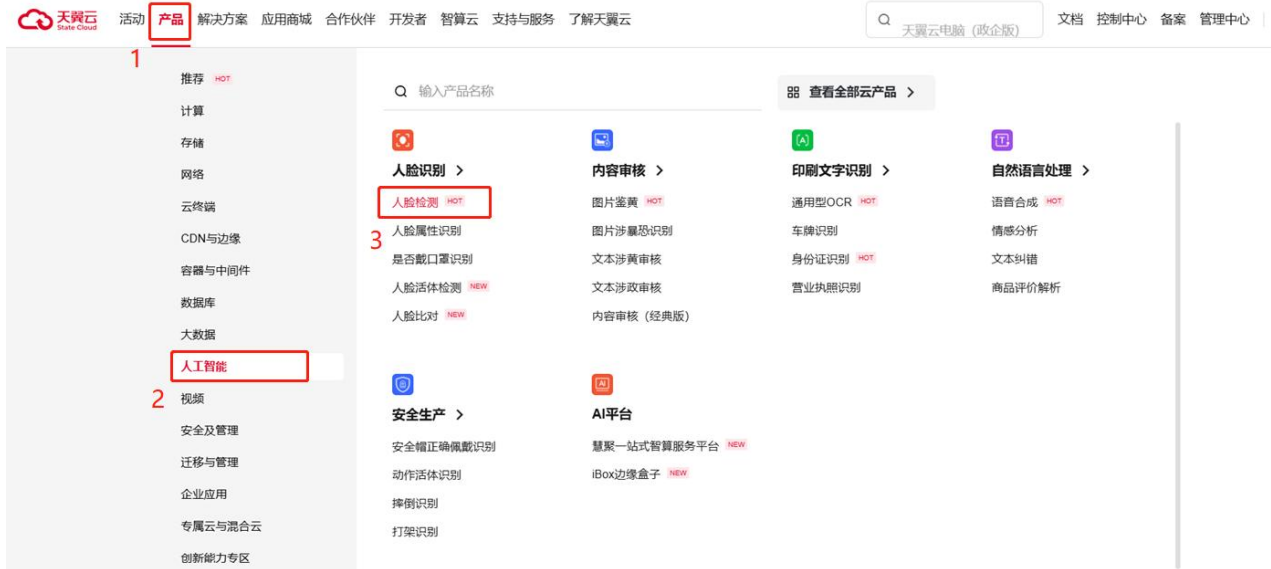
天翼云账号实名认证

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。

- 进入账号中心页面，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。
- 如果您是企业用户，推荐进行企业认证，以便获取更多便利。更多实名认证操作信息，请参见实名认证。

步骤二：选择服务

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】。



步骤三：选择资源包

1. 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】。

三
天翼云 State Cloud
控制台

Q 搜索...
费用
工单
备案
支持
合作
?
🔔

< 能力列表

人脸检测

选择配置

规格	调用有效期	单价	购买数量
100万次	1年	¥450.00	- 1 +
1,000万次	1年	¥4,100.00	- 0 +
5,000万次	1年	¥17,500.00	- 0 +
10,000万次	1年	¥25,000.00	- 0 +

配置费用
¥450.00

参考价格，具体扣费请以账单为准。[了解计费详情](#)

取消
下一步：确认配置

2. 点击勾选同意相关协议，点击【立即购买】。

三
天翼云 State Cloud
控制台

Q 搜索...
费用
工单
备案
支持
合作
?
🔔

< 能力列表

产品名称	计费方式	规格	调用有效期	购买数量	单价
人脸检测	包周期	100万次	1年	1	¥450.00

协议

☒ 我已阅读并同意相关协议。《天翼云人脸识别服务协议》

配置费用
¥450.00

参考价格，具体扣费请以账单为准。[了解计费详情](#)

上一步
立即购买

步骤四：支付订单

点击【立即支付】，跳转到支付成功界面即购买成功。

费用中心

订单管理

资金管理

账单管理

产品管理

资源管理

成本管理

合同管理

卡券管理

订单支付详情

订单号: 20241204171325744978 订单类型: 订购 创建时间: 2024-12-04 17:13:26 更新时间: 2024-12-04 17:13:26

产品	配置	订购数量	所属资源池	周期	金额 (元)
人脸识别	人脸识别_100万次	1	-	1次	450.00元

费用合计: 450.00元

支付方式: 预付费

账户余额(元)
(账户余额 ¥ 0.00)

代金券(元)
(代金券余额 ¥ 0.00)

优惠券
无可用优惠券

订单费用 + 450.00元

账户余额 - 0.00元

代金券 - 0.00元

优惠券 - 0.00元

需支付 450.00元

立即支付

1. 支付订单请务必确认所有商品已进行实名认证, 如无, 请点击 [立即实名认证](#)

2. 云市场、甄选商城及甄选服务订单不能使用优惠券/代金券支付

3. 订单不支持同时使用代金券和优惠券

4. 批量支付时订单为同一客户账号才可同时使用优惠券, 多个账号无法使用优惠券

5. 批量支付只可使用一张优惠券, 请您选择符合优惠券支付

费用中心

订单管理

资金管理

账单管理

产品管理

资源管理

成本管理

合同管理

卡券管理

我的订单/支付结果

支付成功

查看订单列表

订单号: 20240424102323434062 订单类型: 订购 创建时间: 2024-04-24 10:23:24 更新时间: 2024-04-24 10:23:58

产品	配置	订购数量	所属资源池	周期	金额 (元)
人脸识别	人脸识别_100万次	1	-	1次	450.00元

费用合计: 450.00元

支付方式: 后付费

订单费用: + 450.00元

4.2 创建应用及开通应用

本文以人脸检测为例作为说明。

操作步骤如下：

成为天翼云用户

注册天翼云账号

打开天翼云官网，点击右上角【免费注册】按照操作提示完成账号注册。

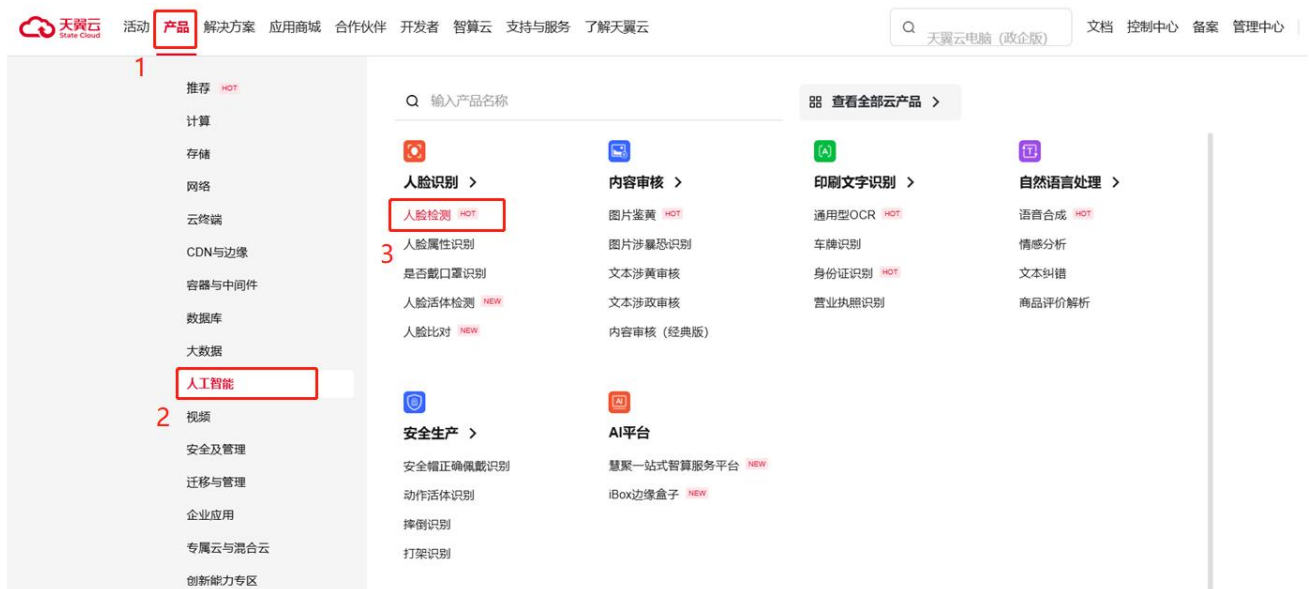
天翼云账号实名认证

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。

- 进入账号中心页面，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。
- 如果您是企业用户，推荐进行企业认证，以便获取更多便利。更多实名认证操作信息，请参见实名认证。

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）。

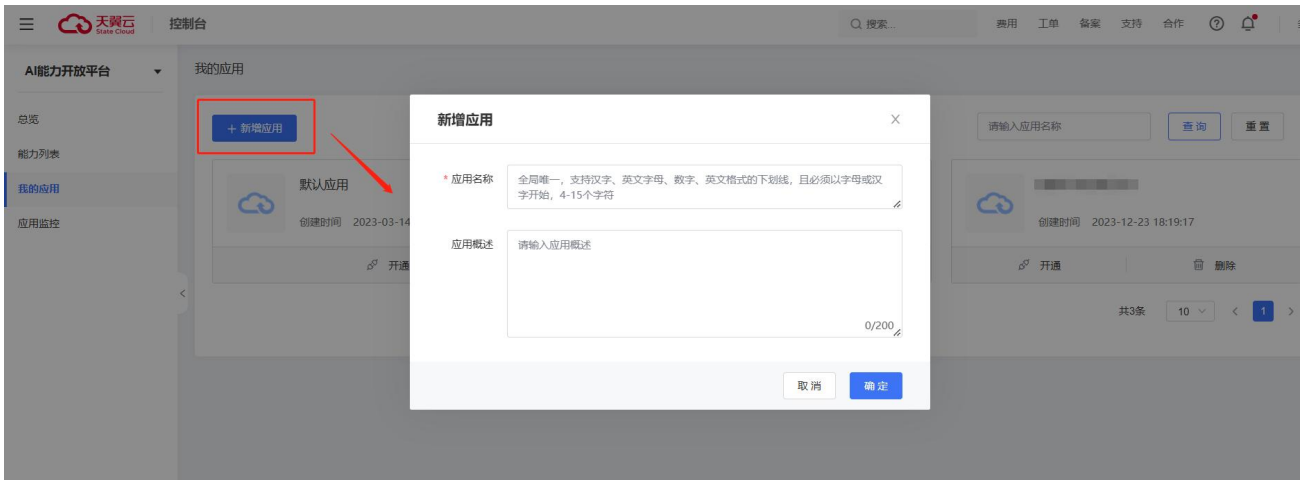


2. 跳转到服务详情页后，点击【立即开通】；

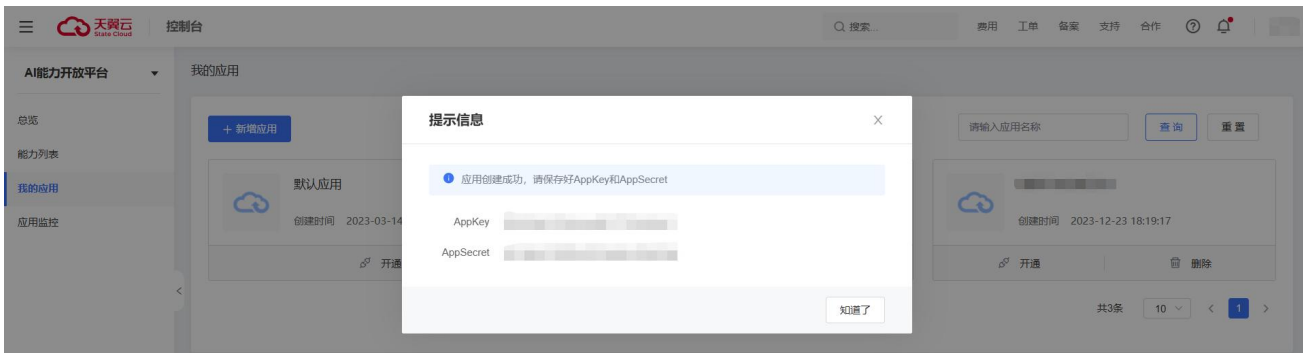


创建新应用

1. 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。

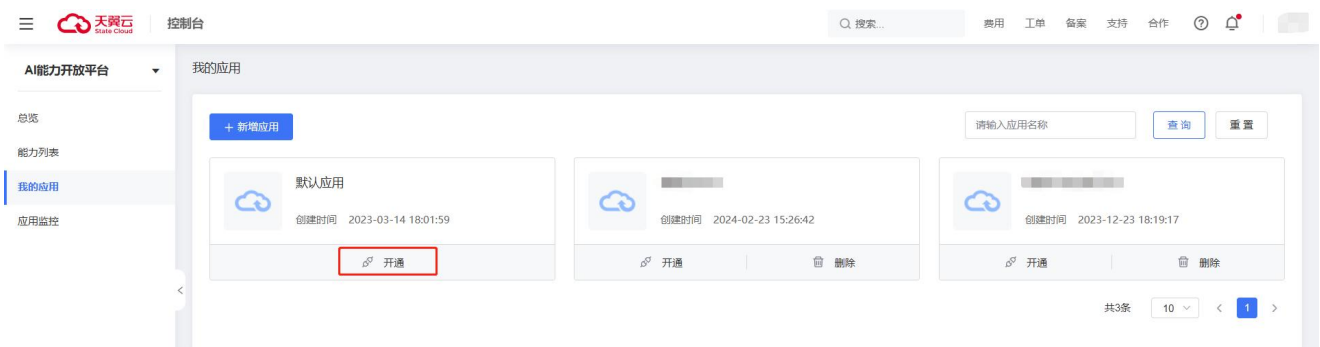


2. 再次弹出小弹窗，提示应用创建成功，请保存好AppKey和AppSecret，点击【知道了】即新应用创建成功。



加入应用

1. 根据需要选择应用，点击【开通】，跳转到应用详情页。



2. 点击【去开通】，添加能力。

天翼云 控制台

我的应用 / 应用详情

< 应用详情

基础信息 [修改](#)

创建应用测试

描述 -

AppKey 05f45f4bc333e3c4ba017703af1b9c6 AppSecret ***** [重置](#)

< 已开通能力 [开通](#)

暂无已开通能力 [去开通](#)

天翼云 控制台

我的应用 / 应用详情

< 添加能力

已选择能力 人脸检测 x

热门能力

新上线能力

AI技术能力

人脸识别

安全生产

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](#)

查看应用

点击确认开通，即反馈【能力已开通】，并返回应用详情，在能力管理中展示了已开通的能力。

三 天翼云 控制台

Q 搜索... 费用 工单 备案 支持 合作

A能力开放平台

我的应用 / 应用详情

应用详情

基础信息

创建应用测试

描述 -

AppKey [redacted] AppSecret ***** 重置

已开通能力 开通

人脸检测 待购买

获取EncKey

说明 使用人脸识别及印刷文字识别相关接口时，可通过获取的EncKey对数据进行加密和解密。

EncKey ***** 获取

业务信息及管理

应用分类	-	应用平台	-
应用说明	-		
白名单	-	黑名单	-

获取应用的AppKey、AppSecret

点击应用，在应用详情页可以查看对应的AppKey、AppSecret。

三 天翼云 控制台

Q 搜索... 费用 工单 备案 支持 合作

A能力开放平台

我的应用 / 应用详情

应用详情

基础信息

创建应用测试

描述 -

AppKey [redacted] AppSecret ***** 重置

已开通能力 开通

人脸检测 待购买

获取EncKey

说明 使用人脸识别及印刷文字识别相关接口时，可通过获取的EncKey对数据进行加密和解密。

EncKey ***** 获取

业务信息及管理

应用分类	-	应用平台	-
应用说明	-		
白名单	-	黑名单	-

4.3 用户控制台

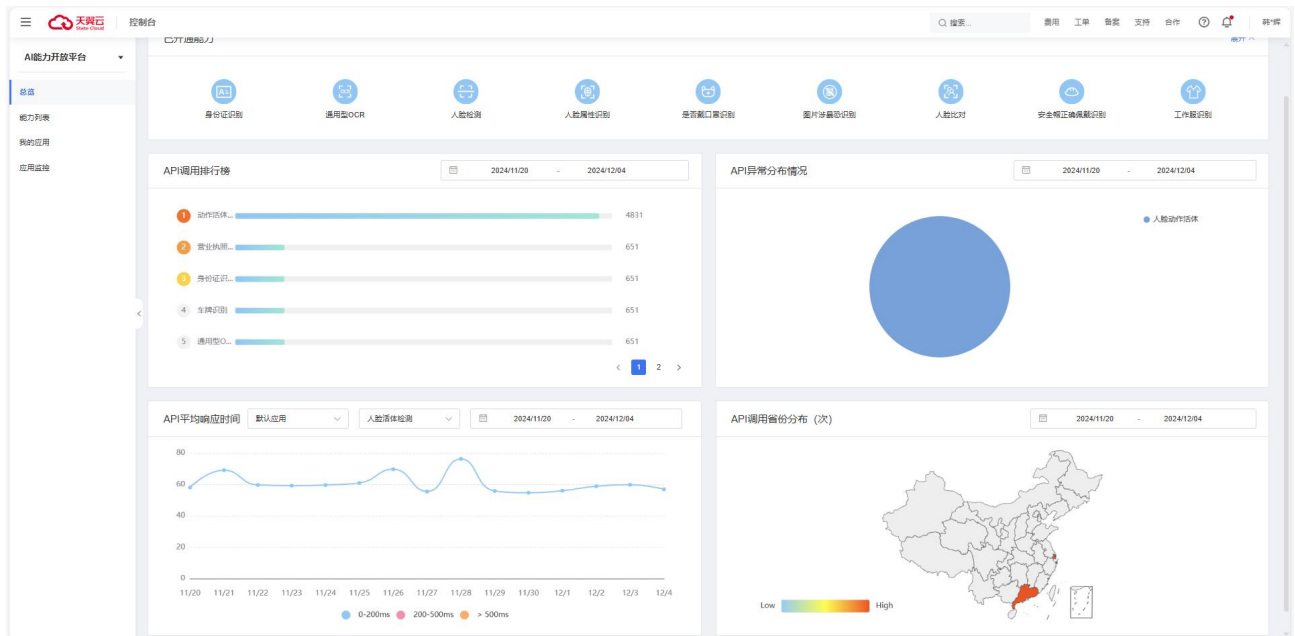
控制台

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。



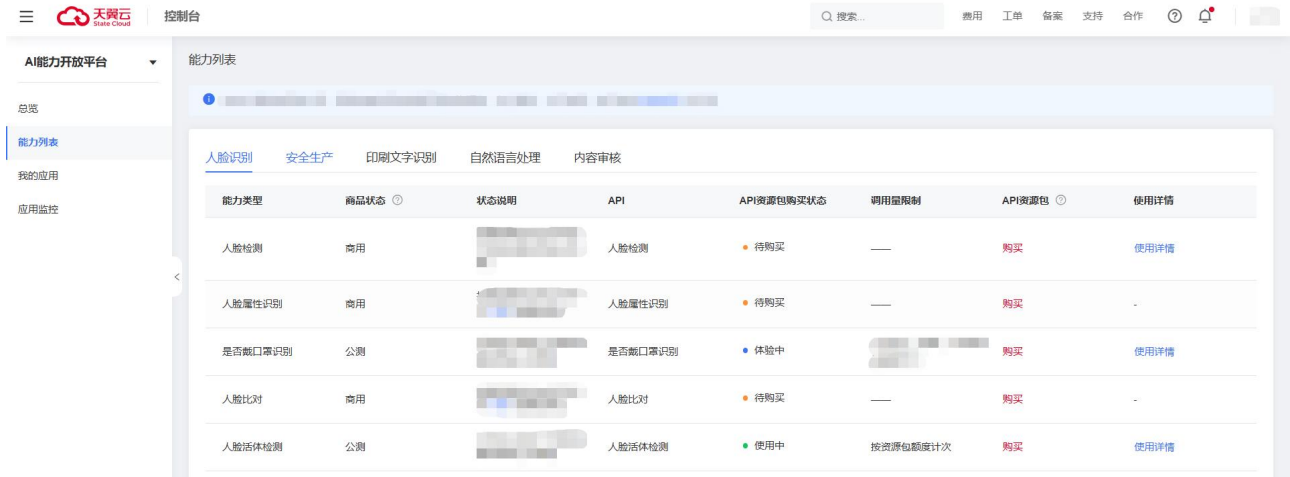
总览

点击左侧菜单栏【总览】，可以查看已开通能力、API调用排行榜以及平均响应时间等内容。



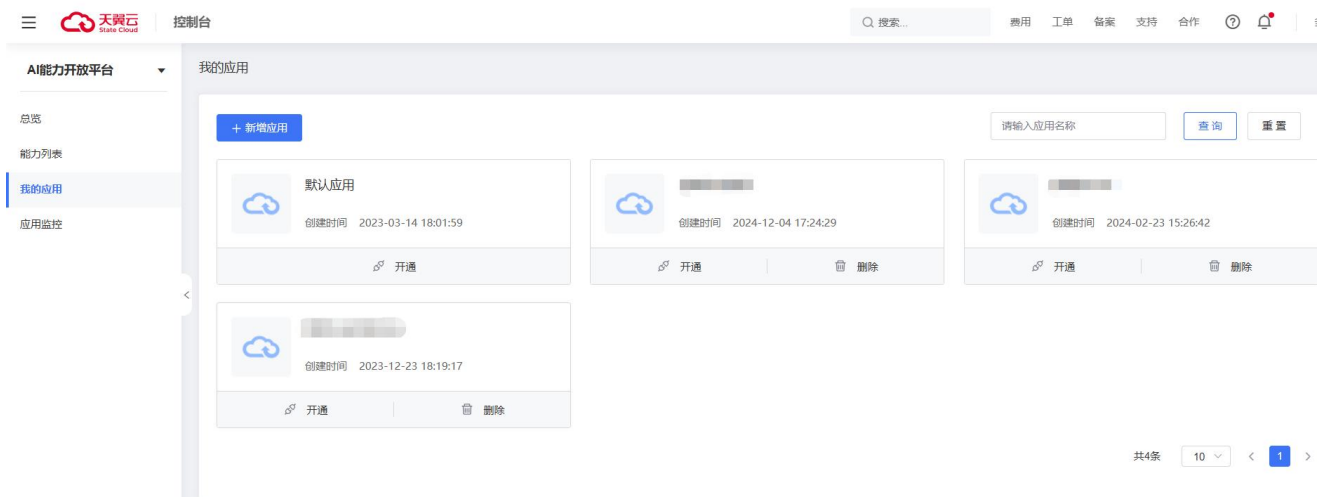
能力列表

点击左侧菜单栏【能力列表】，可以查看已有能力。



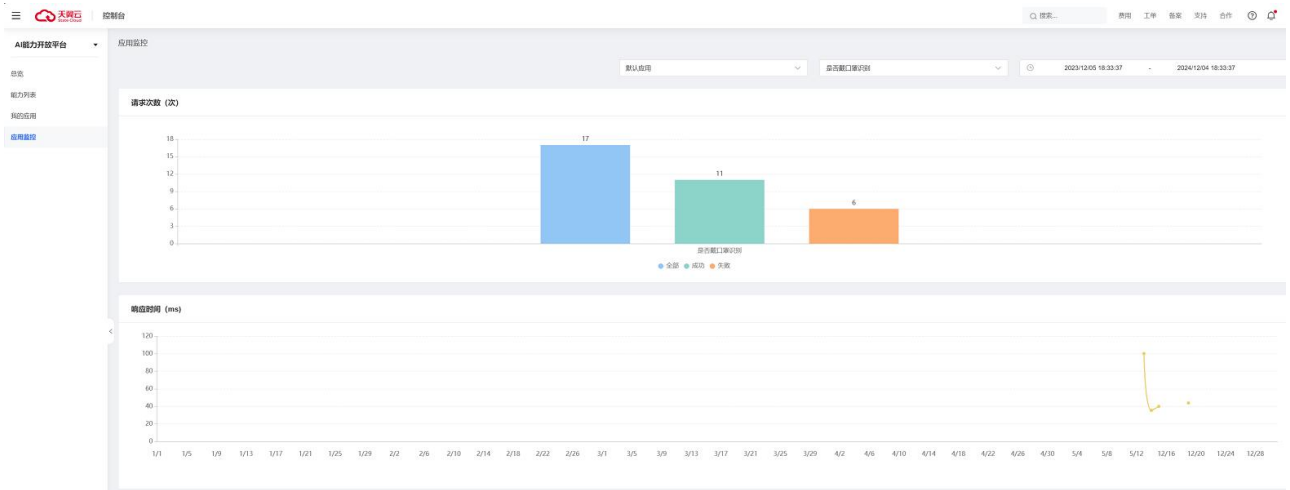
我的应用

点击左侧菜单栏【我的应用】，可以查看已经开通的应用。



应用监控

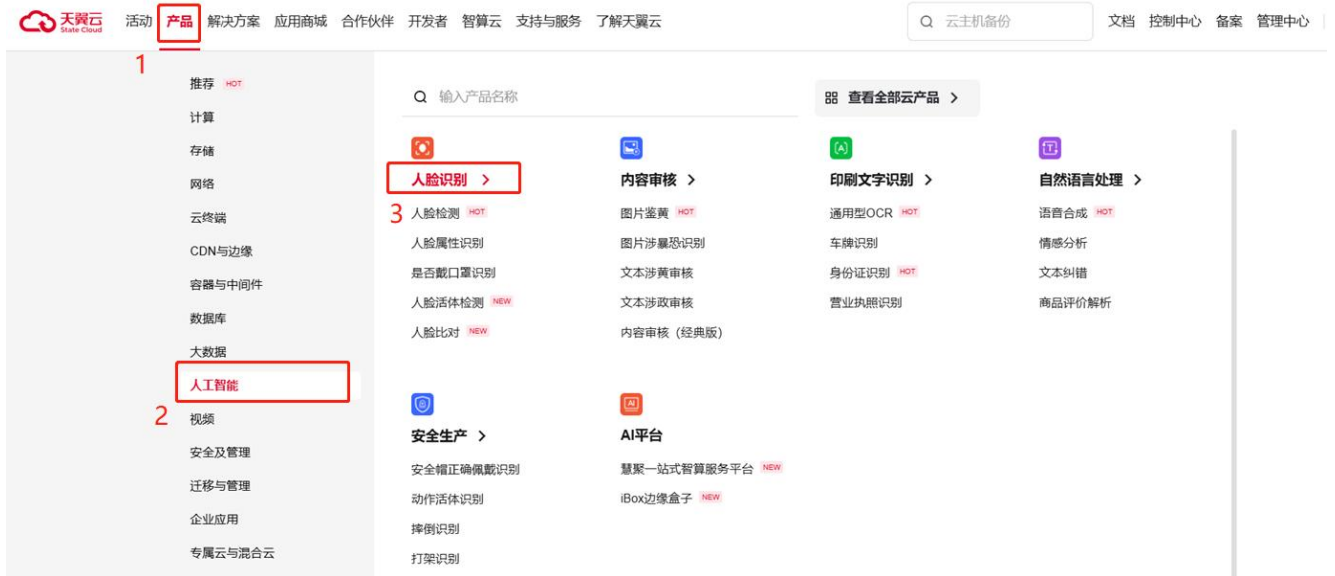
点击左侧菜单栏【应用监控】，可以查看所创建应用的请求次数、响应时间与请求流量。



4.4 API调用指南

1.选择产品聚合页

点击【产品-人工智能】，选择【人脸识别】，打开对应的产品聚合页。



2.打开产品文档

点击【产品文档】，跳转到对应的文档中心，文档中心中【API参考】章节介绍了API调用的相关说明。

人脸识别

人脸识别（Face Recognition）服务提供人脸检测、人脸活体检测、人脸属性识别、是否戴口罩识别、人脸比对等功能。为开发者和企业提供高性能的在线API服务和解决方案，针对图片或视频进行分析，可应用于人脸识别和认证、安防考勤、人脸搜索等各种场景。

[立即开通](#) [管理控制台](#) [产品价格](#) [产品文档](#)

算力套餐
让算力触手可及

云上钜惠
领取5万元大礼包更实惠

云主机特惠
新老同享云主机3折起

动态 更新Java调用示例
动态 支持用户查看AK, SK
动态 查看人脸识别的认证鉴权

3.查看API的请求地址

API的请求地址格式为：{终端节点地址}+{对应接口URI}。

3.1终端节点地址

https://ai-global.ctapi.ctyun.cn。



天翼云 活动 产品 解决方案 应用商城 合作伙伴 开发者 智算云 支持与服务 了解天翼云

Q 弹性云主机 文档 控制中心 备案 管理中心

查看所有产品

人脸识别

筛选文档目录

产品退订

快速入门

用户指南

最佳实践

API参考

调用前必知

如何调用API

终端节点

构造请求

帮助中心 > 人脸识别 > API参考 > 如何调用API > 终端节点

终端节点

最近更新时间: 2023-12-29 01:07:55

主要作用：用户信息的发送和接收，信令信息的控制处理、安全保护等作用。

- https://ai-global.ctapi.ctyun.cn。

< 上一篇：如何调用API

下一篇：构造请求 >

相关文档

- 服务条款
- 人脸活体检测
- 购买类
- 术语解释
- 功能特性
- 应用场景

以上内容是否对您有帮助？ ☐ 有 ☐ 没有

3.2对应接口URI

以人脸检测为例，选择【API参考-API-人脸检测】，右侧即可查看URI。

4.查看接口文档

以人脸检测为例：点击【API参考】-【API】，选择【人脸检测】，右侧即可查看、请求方法、接口要求、请求URL、请求参数、返回值说明、状态码等信息。

5.构造请求

点击【API参考-如何调用API】，选择【构造请求】。



6. 认证鉴权

在帮助中心菜单中点击【认证鉴权】，可根据右侧步骤完成认证鉴权。

具体步骤：信息的获取——基本签名流程——创建待签名字符串——构造动态密钥——签名应用及示例。



7. 调试接口并查看状态码

调试后，如果API返回状态码为200，表示请求成功，API调试结束。

查看所有产品

人脸识别

筛选文档目录

用户指南

最佳实践

API 参考

调用前必知

如何调用API

API

人脸检测

人脸属性识别

人脸比对

人脸活体检测

是否戴口罩识别

```
{
  "statusCode": "4101",
  "message": "error",
  "details": "请求内容错误",
  "error": "AI_OP_4101"
}
```

状态码

http状态码	描述
200	表示请求成功

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
AI_OP_4101	请求内容错误	传入内容为空，或者传入的参数名错误
AI_OP_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
AI_OP_4103	图片大小超过2M	图片大小超过2M
AI_OP_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

本目录

- 接口描述
- 请求方法
- 接口要求
- URI
- 请求参数
 - 1. 请求头header参数
 - 2. 请求体body参数
- 请求代码示例
- 返回值说明
 - 1. 请求成功返回响应参数
 - 2. 请求失败返回响应参数
- 返回值示例
 - 1. 请求成功返回响应示例
 - 2. 请求失败返回响应示例
- 状态码
- 错误码说明
 - base64 编码规则: { 常规的 safe base64 方式



5 最佳实践

5.1 人脸活体检测

背景信息

对于首次使用人脸活体检测业务的用户，如希望快速的解决业务需求问题，可参照本实践案例，通过应用场景、产品功能、前提条件、前期准备、实践步骤等说明，实现业务快速接入。

应用场景

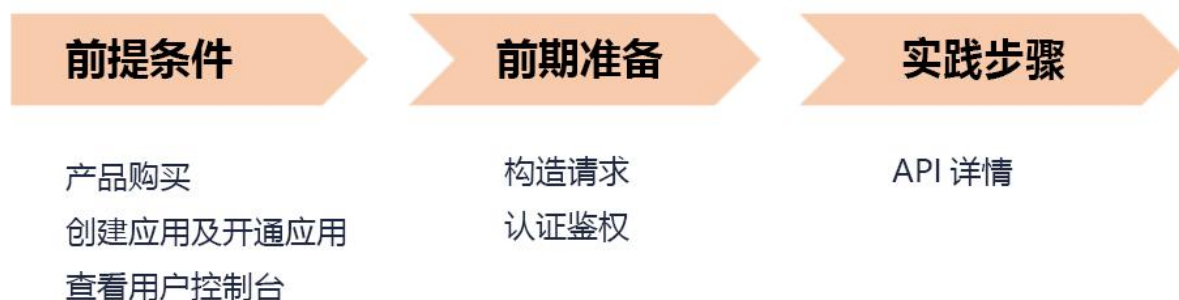
金融服务：在银行、金融机构和支付应用中，活体检测可以用于确保用户在开设账户、进行交易或验证身份时是真实的人，从而增强账户安全性。

智能门禁：在办公大楼、住宅小区等场所的门禁系统中，活体检测可以防止未经授权的人员通过使用照片或假面具进入。

产品功能

基于复杂的神经网络模型深度学习方法，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，通过使用人脸活体检测技术，可以有效地防止恶意用户使用非真实的人脸进行欺骗和冒充。

实践流程



前提条件

- 购买人脸活体检测相关产品，订购流程详情请查看产品购买；
- 创建人脸活体检测应用及开通人脸检测应用，开通服务流程详情请查看创建应用及开通应用；
- 查看购买人脸活体检测产品情况，产品整体情况请查看用户控制台。

前期准备

构造请求

- 请求地址：{终端节点地址}+{对应接口URL}，详情查看构造请求；
- 终端节点地址：https://ai-global.ctapi.ctyun.cn；
- 对应接口URI：
/v1/aiop/api/2f6hqix09mv4/face/PERSON/person/detectFaceFromBase64。

认证鉴权

认证鉴权详细版本，查看认证鉴权章节。

1.信息的获取

- 登录云网门户，在“控制台”->“账号中心”->“安全设置”->“用户AccessKey”点击“查看”获取。

2.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=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))；
- 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；

- 如果你加入一个ccad的header，同时这个header也需要进行签名，则待签名的header组合：ccad:123\nctyun-eop-request-id:123456789\neop-date:20210531T100101Z\nhost:1.1.1.1:9080\n。

4.构造动态密钥

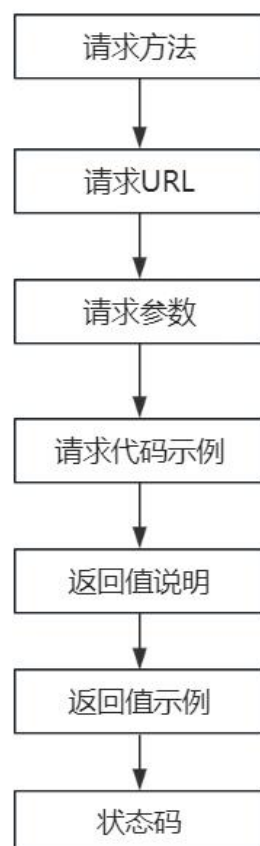
- 发起请求时，需要构造一个eop-date的时间，这个时间的格式是yyyymmddTHHMMSSZ，言简意赅一些，就是年月日T时分秒Z；
- 先是用申请的ctyun-eop-sk作为密钥，eop-date作为数据，算出ktime；
- 用ktime作为密钥，申请的ctyun-eop-ak作为数据，算出kAk；
- 用kAk作为密钥，eop-date的年月日值作为数据，算出kdate。

5.签名应用

- 由“构造动态密钥”和“创建待签名字符串”分别得出来的待签名字符串string_signature、kdate生成出Signature；
- 得到Eop-Authorization，然后将数据整合成header放在http_client内，发出即可。

实践步骤

实践步骤详情查看API人脸活体检测。



5.2 人脸比对

背景信息

对于首次使用人脸比对业务的用户，如希望快速的解决业务需求问题，可参照本实践案例，通过应用场景、产品功能、前提条件、前期准备、实践步骤等说明，实现业务快速接入。

应用场景

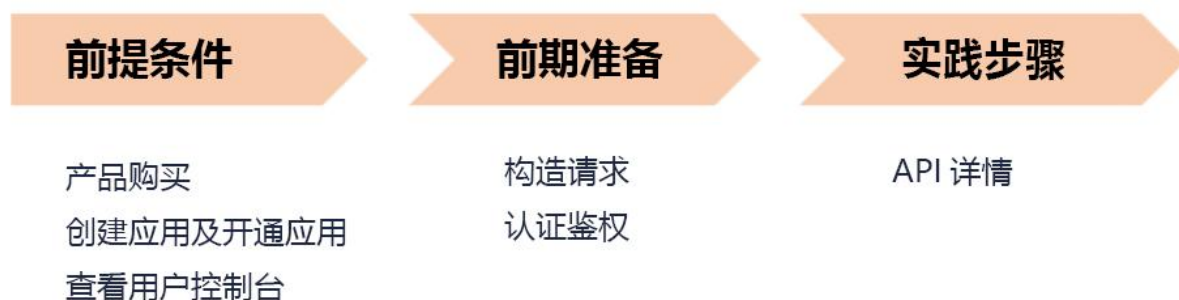
身份验证：在银行、金融机构、医疗机构等场所，人脸比对可以用于验证用户的身份，确保他们具有合法的访问权限。

考勤管理：在企业、学校等地方，人脸比对可以用于员工或学生的考勤管理，准确记录出勤情况。

产品功能

高精度1：1人脸匹配，轻松应对身份验证、门禁系统的人脸比对等业务场景。通过高维度特征抽取，计算相似度，可以根据设定阈值自主准确判断是否为同一人。

实践流程



前提条件

- 购买人脸比对相关产品，订购流程详情请查看产品购买；
- 创建人脸比对应用及开通人脸比对应用，开通服务流程详情请查看创建应用及开通应用；
- 查看购买人脸比对产品情况，产品整体情况请查看用户控制台。

前期准备

构造请求

- 请求地址: {终端节点地址}+{对应接口URL}, 详情查看构造请求;
- 终端节点地址: <https://ai-global.ctapi.ctyun.cn>;
- 对应接口URL:
`/v1/aiop/api/2f7awxekgvls/face/compare/PERSON/person/compareFromBase64`.

认证鉴权

认证鉴权详细版本, 查看认证鉴权章节。

1.信息的获取

- 登录云网门户, 在“控制台”->“账号中心”->“安全设置”->“用户AccessKey”点击“查看”获取。

2.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=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));
- 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;
- 如果你加入一个ccad的header, 同时这个header也需要进行签名, 则待签名的header组合: ccad: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;
- 先是用申请的ctyun-eop-sk作为密钥, eop-date作为数据, 算出ktime;

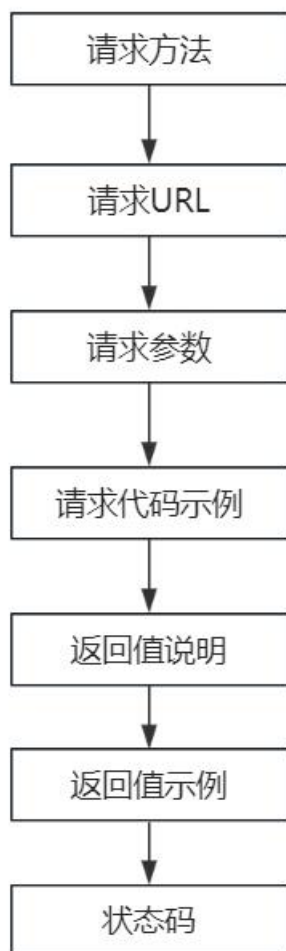
- 用ktime作为密钥，申请的ctyun-eop-ak作为数据，算出kAk；
- 用kAk作为密钥，eop-date的年月日值作为数据，算出kdate。

5.签名应用

- 由“构造动态密钥”和“创建待签名字符串”分别得出来的待签名字符串string_sigture、kdate生成出Signature；
- 得到Eop-Authorization，然后将数据整合成header放在http_client内，发出即可。

实践步骤

实践步骤详情查看API人脸比对。



6.1 调用前必知

6.1.1概述

人脸识别是天翼云自研AI平台提供的产品之一，通过自主研发人脸识别算法模型，提供人脸系列API服务，包括人脸检测定位，人脸属性识别，人脸比对等能力。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，为开发者和企业提供高性能的在线API服务和解决方案，针对图片进行分析，可应用于人脸识别和认证、安防考勤等各种场景。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。

6.1.2API概览

功能名称	描述
人脸检测	对自然场景下的人脸进行快速检测，返回检测到的人脸区域坐标，可实时检测多张人脸
人脸属性识别	针对检测到的人脸，准确识别多种人脸属性信息，包括年龄、性别等，支持多人脸属性实时高效检测，可用于安防监控等业务场景
人脸比对	对检测到的人脸进行高维度的特征抽取，计算相似度，业务侧可以根据设定阈值判断是否为同一个人。可进行1：1高精度人脸匹配，应用于人脸比对等业务场景
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

功能名称	描述
	活体判断，有效防止照片等非活体攻击
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，有效应对大流量人员场景

6.1.3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

6.2 如何调用API

6.2.1终端节点

- 主要作用：用户信息的发送和接收，信令信息的控制处理、安全保护等作用。
- <https://ai-global.ctapi.ctyun.cn>。

6.2.2构造请求

请求地址

- {终端节点地址}+{对应接口URL}。
- 终端节点地址：<https://ai-global.ctapi.ctyun.cn>。
- 对应接口URL：每个API对应一个接口URL，请参考各个接口URL。

通信协议

- 接口通过 HTTPS 进行通信，保护用户数据的机密性和完整性，确保网络通信的安全性。

请求方法

- 支持的 HTTP 请求方法：POST。
- POST 请求支持的 Content-Type 类型：application/json。

请求头及说明

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
decrypted	接口加密功能开关，需要启用接口加密时，设置decrypted=true

6.2.3认证鉴权

信息的获取

云网平台获取

登录云网门户，在“控制台”->“账号中心”->“安全设置”->“用户AccessKey”点击“查看”获取。

用户AccessKey

支持新建、查询用户AccessKey，用于第三方控制管理



拖动滑块完成拼图



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。

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\n
eop-date:20210531T100101Z\n
host: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也要进行签名，则待签名的header组合：
ccad:123\n
ctyun-eop-request-id:123456789\n
eop-date:20210531T100101Z\n
host:1.1.1.1:9080\n

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。

- 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	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)

签名应用及示例

由“构造动态密钥”和“创建待签名字符串”分别得出来的待签名字符串string_signature、kdate生成出Signature。

Signature	待签名字符串string_signature、kdate；再根据 hmacsha256(kdate, string_signature)得出的结果，再将结果进行base64编码得出Signature
Eop-Authorization	{ctyun-eop-ak} headers=你构造待签名字符串时的header排序 Signature={Signature} (ctyun-eop-ak后及Signature前都有一个空格))header排序以分号”；” 拼接例子：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

注意：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。

6.2.4接口加密

1 如何启用加密功能

客户端发起请求时添加请求Header头，decrypted=true 即可启用接口加密功能。

2 加密算法说明

该功能涉及到三种国密算法，分别为非对称加密SM2、对称加密SM4及特征码算法Hmac-SM3。

2.1 非对称加密SM2

类似RSA的非对称加密，用来加密客户生成的对称加密SM4密钥。SM2公钥在AI能力开放平台(<https://ai.ctyun.cn/console>)中我的应用中应用详情下获取，控制台显示内容为经过Base64加密后的公钥字符串。算法标准约定：SM2算法采用BC包(BouncyCastle)，sm2p256v1标准，格式为PKCS8，拼接方式C1C3C2，非压缩格式。

2.2 对称加密SM4

客户端自行生成128或者256位SM4密钥，采用BC包(BouncyCastle),格式为SM4/ECB/PKCS5Padding。

2.3 特征码算法Hmac-SM3

Hmac和SM3算法配合生成特征码串，防止接口被拦截篡改，生成特征码时的密钥需使用。encryptedHashKey字段传递到服务端。

2.4 客户端加密步骤

- 1、从控制台获取公钥串，使用Base64解密获取公钥字节数组。
- 2、生成SM4对称密钥字节数组，使用第1步的公钥加密对称密钥，并使用Base64加密结果，填入ciphertextBlob字段。
- 3、使用第2步生成的对称密钥加密接口原版Body体JSON串，并使用Base64加密结果，填入encryptedBody字段。

- 4、生成Hmac-sm3哈希密钥，使用公钥进行加密，并使用Base64加密结果，填 encryptedHashKey 字段。
- 5、使用Hmac-sm3及第4步哈希密钥，提取 ciphertextBlob 字段特征码并使用Base64加密结果，填入 ciphertextBlobHash 字段。
- 6、使用Hmac-sm3及第4步哈希密钥，提取 encryptedBody 字段特征码并使用Base64加密结果，填入 encryptedBodyHash 字段。

2.5 请求体结构

入参结构体如下，下列五个参数都需要经过Base64加密处理：

```
{  
  "ciphertextBlob": "使用SM2公钥加密后的客户生成SM4对称密钥",  
  "encryptedBody": "使用对称密钥SM4加密后的接口原版Body体JSON串，编码UTF-8",  
  "encryptedHashKey": "使用公钥加密后的进行hmac-sm3哈希时使用的密钥",  
  "ciphertextBlobHash": "ciphertextBlob字段的hmac-sm3结果",  
  "encryptedBodyHash": "encryptedBody字段的hmac-sm3结果"  
}
```

2.6 返回体结构

若接口返回非0错误，则不进行加密，结构同未使用加密功能时一致，客户端可根据返回体是否包含statusCode进行区分。

若接口请求成功且返回statusCode为0，则会进行加密处理，返回值如下，下列两个参数都经过Base64加密处理：

```
{  
  "encryptedResultHash": "encryptedResult字段的hmac-sm3结果，哈希密钥与客户端一致",  
  "encryptedResult": "SM4对称密钥加密后的接口响应结果"}  
}
```

客户端获取返回值后可选择性校验encryptedResultHash确保请求结果未被篡改。

获取encryptedResult字段后，使用请求时的sm4对称密钥进行解密，即可获得到未启用加密时接口的正常返回体内容（JSON串），编码UTF8。

2.7 加解密相关错误码说明

encryptedResult字段通过SM4对称密钥解密后可获取到不启用加密功能格式的返回值，相比未使用加密功能的请求方式，会有以下几种特殊错误码：

错误码	错误信息	错误描述
-----	------	------

错误码	错误信息	错误描述
AI_OP_40017	加密参数不符合要求	入参格式不符合加密功能要求
AI_OP_40018	ciphertextBlob哈希值不匹配 /encryptedBody哈希值不匹配	相关字段哈希值不匹配，存在被篡改可能
AI_OP_40019	SM2解密失败	SM2非对称解密出现异常
AI_OP_40020	SM4加密失败/SM4解密失败	SM4对称密钥加解密出现异常

3 参考代码(JAVA, jdk1.8及以上)

3.1 maven工程引入以下依赖

```

<!-- BC包，若jdk为1.8以下可替换对应版本及artifactId-->
<dependency>
<groupId>org.bouncycastle</groupId>
<artifactId>bcprov-jdk18on</artifactId>
<version>1.78.1</version>
</dependency>

<!--Base64功能包，推荐使用此包内Base64类进行加解密，生成加密结果不能包含换行-->
<dependency>
<groupId>commons-codec</groupId>
<artifactId>commons-codec</artifactId>
<version>1.17.1</version>
</dependency>

```

3.2 SM2工具类

```

import org.bouncycastle.jce.provider.BouncyCastleProvider;import javax.crypto.Cipher;import java.security.*;import
java.security.spec.ECGenParameterSpec;import java.security.spec.PKCS8EncodedKeySpec;import
java.security.spec.X509EncodedKeySpec;public class SM2Util {static {

Security.addProvider(new BouncyCastleProvider());

} // 生成SM2密钥对public static KeyPair generateKeyPair() {try{KeyPairGenerator keyPairGenerator =
KeyPairGenerator.getInstance("EC", "BC");

keyPairGenerator.initialize(new ECGenParameterSpec("sm2p256v1"));return keyPairGenerator.generateKeyPair();

} catch (Exception e) { //TODO 做异常处理

}

} // 加密

```

```
public static byte[] encrypt(byte[] publicKey, byte[] data) {
    try{
        PublicKey pubKey = KeyFactory.getInstance("EC", "BC")
            .generatePublic(new X509EncodedKeySpec(publicKey));

        Cipher cipher = Cipher.getInstance("SM2", "BC");
        cipher.init(Cipher.ENCRYPT_MODE, pubKey);

        return cipher.doFinal(data);
    } catch (Exception e) {
        //TODO 做异常处理
    }
}

// 解密
public static byte[] decrypt(byte[] privateKey, byte[] encryptedData) {
    try{
        PrivateKey priKey = KeyFactory.getInstance("EC", "BC")
            .generatePrivate(new PKCS8EncodedKeySpec(privateKey));

        Cipher cipher = Cipher.getInstance("SM2", "BC");
        cipher.init(Cipher.DECRYPT_MODE, priKey);

        return cipher.doFinal(encryptedData);
    } catch (Exception e) {
        //TODO 做异常处理
    }
}

public static void main(String[] args) {
    KeyPair keyPair = generateKeyPair();
    byte[] a = keyPair.getPublic().getEncoded();
    byte[] b = keyPair.getPrivate().getEncoded();
}
}
```

3.3 SM4工具类

```
import org.bouncycastle.jce.provider.BouncyCastleProvider;

import javax.crypto.Cipher;import javax.crypto.KeyGenerator;import javax.crypto.SecretKey;import
javax.crypto.spec.SecretKeySpec;import java.security.Security;
```

```
public class SM4Util {

    static {
        Security.addProvider(new BouncyCastleProvider());
    }

    // 生成SM4密钥
    public static byte[] generateKey() {
        try{
            KeyGenerator keyGenerator = KeyGenerator.getInstance("SM4", "BC");
            keyGenerator.init(128); // 可选 128 或 256
            SecretKey secretKey = keyGenerator.generateKey();
            return secretKey.getEncoded();
        } catch (Exception e) {
            //TODO 处理异常
        }
    }

    // 加密
    public static byte[] encrypt(byte[] keyBytes, byte[] data) {
        try{
            SecretKeySpec keySpec = new SecretKeySpec(keyBytes, "SM4");
            Cipher cipher = Cipher.getInstance("SM4/ECB/PKCS5Padding", "BC");
            cipher.init(Cipher.ENCRYPT_MODE, keySpec);
            return cipher.doFinal(data);
        } catch (Exception e) {
            //TODO 处理异常
        }
    }

    // 解密
    public static byte[] decrypt(byte[] keyBytes, byte[] encryptedData) {
        try{
            SecretKeySpec keySpec = new SecretKeySpec(keyBytes, "SM4");
            Cipher cipher = Cipher.getInstance("SM4/ECB/PKCS5Padding", "BC");
            cipher.init(Cipher.DECRYPT_MODE, keySpec);
            return cipher.doFinal(encryptedData);
        } catch (Exception e) {
            //TODO 处理异常
        }
    }
}
```

```
    }  
  
    }  
}
```

3.4 Hmac-sm3工具类

```
import org.apache.commons.codec.binary.Base64;import org.bouncycastle.crypto.digests.SM3Digest;import  
org.bouncycastle.crypto.macs.HMac;import org.bouncycastle.crypto.params.KeyParameter;  
  
import java.security.SecureRandom;  
  
public class HmacSM3Util {  
  
    /**  
     * 计算 HMAC-SM3  
     *  
     * @param key 密钥  
     * @param data 数据  
     * @return HMAC-SM3 值  
     */  
    public static byte[] hmacSM3(byte[] key, byte[] data) {  
        HMac hmac = new HMac(new SM3Digest());  
        hmac.init(new KeyParameter(key));  
  
        hmac.update(data, 0, data.length);  
  
        byte[] result = new byte[hmac.getMacSize()];  
        hmac.doFinal(result, 0);  
  
        return result;  
    }  
  
    /**  
     * 生成指定长度的随机密钥  
     *  
     * @param length 密钥长度（字节）  
     * @return 随机生成的密钥  
     */  
    public static byte[] generateRandomKey(int length) {  
        SecureRandom random = new SecureRandom();  
  
        byte[] key = new byte[length];  
        random.nextBytes(key);  
  
        return key;  
    }  
}
```

```
}

// 示例用法

public static void main(String[] args) {

    // 生成随机密钥

    byte[] key = generateRandomKey(16); // 16字节长度的随机密钥

    // 待计算的数据

    byte[] data = "Hello, HMAC-SM3!".getBytes();

    // 计算 HMAC-SM3 值

    byte[] hmacValue = hmacSM3(key, data);

    // 打印结果

    System.out.println("HMAC-SM3 值: " + Base64.encodeBase64String(hmacValue));

}

}
```

3.5 生成请求体

```
//未启用加密功能时的请求体json串, 查询对应接口文档确定格式

String body = "{\"ImageData\":\"imagedatabase64xxxx\"}";

//控制台-应用详情下获取的公钥

String encKey = "xxxxxxx";

Map<String,String> requestBody = new HashMap<>();

//公钥转换为字节数组备用

byte[] publicKey = Base64.decodeBase64(encKey);

//生成ciphertextBlob字段

byte[] sm4Key = SM4Util.generateKey();

requestBody.put("ciphertextBlob", Base64.encodeBase64String(SM2Util.encrypt(publicKey, sm4Key)));

//生成encryptedBody字段

requestBody.put("encryptedBody", Base64.encodeBase64String(SM4Util.encrypt(sm4Key, body.getBytes())));

//生成Hmac-sm3密钥

byte[] hmacSm3Key = HmacSM3Util.generateRandomKey(16);

//生成encryptedHashKey字段

requestBody.put("encryptedHashKey", Base64.encodeBase64String(SM2Util.encrypt(publicKey, hmacSm3Key)));

//生成ciphertextBlobHash字段

requestBody.put("ciphertextBlobHash", Base64.encodeBase64String(HmacSM3Util.hmacSM3(hmacSm3Key,
requestBody.get("ciphertextBlob").getBytes())));

//生成encryptedBodyHash字段
```

```
requestBody.put("encryptedBodyHash", Base64.encodeBase64String(HmacSM3Util.hmacSM3(hmacSm3Key,
requestBody.get("encryptedBody").getBytes())));
```

//TODO requestBody即为加密请求体，遵循鉴权逻辑调用接口即可，鉴权部分逻辑请参考鉴权相关说明文档

6.2.5 Python3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%S", 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'

    # 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)
```

6.2.6Java调用示例

附件下载链接: <https://www.ctyun.cn/document/10264960/10029522>

6.2.7PHP调用示例

附件下载链接: <https://www.ctyun.cn/document/10264960/10583402>

6.2.7C#调用示例

附件下载链接: <https://www.ctyun.cn/document/10264960/10612501>

6.3 API

6.3.1人脸检测

接口描述

用于检测输入图像中的人脸，输出人脸位置坐标。

请求方法

POST

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格式。

URI

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

请求参数

1.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	平台应用appkey	562b89493b1a40e1b97ea05e50dd8170	
ctyun-eop-request-id	是	String	用户请求 id，由用户构造，用户可以通过 uuid 等	33dfa732-b27b-464f-b15a-21ed6845afd5	

参数	是否必填	参数类型	说明	示例	下级对象
			方法自行生成唯一字符串，用于日志请求追踪。		
eop-date	是	String	请求时间，由用户构造，形如 yyyyymmddTHHMMSSZ。	20211221T163014Z	
host	是	String	终端节点域名，固定字段	ai-global.ctapi.ctyun.cn	
Eop- Authorizat ion	是	String	由天翼云官网 accessKey 和 securityKey 经签名后生成，参与签名生成的字段包括天翼云官网 accessKey 、securityKey 、平台应用的appkey（非必须），用户请求id（非必须），请求时间，终端节点域名（非必须）以及请求体内容。		

2.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	-	

参数	是否必填	参数类型	说明	示例	下级对象
			“data:image/jpeg;base64, ”		

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...."}'
```

返回值说明

1.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
statusCode	是	String	接口状态码，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statusCode 为 0，返回 success；如果statusCode 非 0，返回 error	success	
returnObj	是	Object	返回的人脸检测结果对象		returnObj

2.表returnObj

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	1	
face_list	是	List	每个人脸的详细信息	-	
face_list[].face_location	是	Object	人脸所处位置	-	

3.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
statusCode	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	
error	是	String	返回对应的错误码	AI_OP_4101	

返回值示例

1.请求成功返回值示例

```
{
  "statusCode": "0",
  "message": "success",
  "returnObj": {
    "face_num": 1,
```

```
"face_list" : [{  
  "face_location" : {  
    "top" : 36,  
    "left" : 48,  
    "width" : 58,  
    "height" : 72  
  }  
}]  
}}
```

2.请求失败返回值示例

```
{  
  "statusCode": "4101",  
  "message": "error",  
  "details": "请求内容错误",  
  "error": "AI_OP_4101"}  
}
```

状态码

http状态码	描述
200	表示请求成功

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
AI_OP_4101	请求内容错误	传入内容为空，或者传入的参数名错误
AI_OP_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
AI_OP_4103	图片大小超过2M	图片大小超过2M
AI_OP_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
AI_OP_4105	未检测到人脸	上传图片中不包含人脸

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().toString()` 函数进行编码。

6.3.2人脸属性识别

接口描述

用于检测输入图像中的人脸年龄、性别等属性。

请求方法

POST

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格式。

URI

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

请求参数

1.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	平台应用appkey	562b89493b1a40e1b97ea05e50dd8170	
ctyun-eop-request-id	是	String	用户请求 id, 由用户构造, 用户可以通过 uuid	33dfa732-b27b-464f-b15a-21ed6845afd5	

参数	是否必填	参数类型	说明	示例	下级对象
			等方法自行生成唯一字符串，用于日志请求追踪。		
eop-date	是	String	请求时间，由用户构造，形如 yyyymmddTHHMMSSZ。	20211221T163014Z	
host	是	String	终端节点域名，固定字段	ai-global.ctapi.ctyun.cn	
Eop-Authorization	是	String	由天翼云官网 accessKey 和 securityKey 经签名后生成，参与签名生成的字段包括天翼云官网 accessKey 、 securityKey、平台应用的appkey（非必须），用户请求id（非必须），请求时间，终端节点域名（非必须）以及请求体内容。		

2.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 “data:image/jpeg;base64,”	-	

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...."}'
```

返回值说明

1.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
statusCode	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	

参数	是否必填	参数类型	说明	示例	下级对象
returnObj	是	Object	返回的人脸属性识别结果对象		returnObj

2.表returnObj

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_location	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	-	

3.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
statusCode	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	
error	是	String	返回对应的错误码	AI_OP_4101	

返回值示例

1.请求成功返回值示例

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

2. 请求失败返回值示例

```
{
  "statusCode": "4101",
  "message": "error",
  "details": "请求内容错误"
  "error": "AI_OP_4101"
}
```

状态码

http状态码	描述
200	表示请求成功

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
AI_OP_4101	请求内容错误	传入内容为空，或者传入的参数名错误
AI_OP_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
AI_OP_4103	图片大小超过2M	图片大小超过2M
AI_OP_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
AI_OP_4105	未检测到人脸	上传图片中不包含人脸

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。

6.3.3人脸比对

接口描述

用于检测输入的两张图像中的人脸相似度。

请求方法

POST

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格式。

URI

`/v1/aiop/api/2f7awxekgvl/face/compare/PERSON/person/compareFromBase64`

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
----	------	------	----	----	------

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	平台应用appkey	562b89493b1a40e1b97ea05e50dd8170	
ctyun-eop-request-id	是	String	用户请求 id，由用户构造，用户可以通过 uuid 等方法自行生成唯一字符串，用于日志请求追踪	33dfa732-b27b-464f-b15a-21ed6845afd5	
eop-date	是	String	请求时间，由用户构造，形如 yyyymmddTHHMMSSZ。	20211221T163014Z	
host	是	String	终端节点域名，固定字段	ai-global.ctapi.ctyun.cn	
version	是	String	服务版本号，固定值v2（目前产品有v1和v2两个版本，v1版本不需要传入此字段信息，建议使用v2版本）	v2	

参数	是否必填	参数类型	说明	示例	下级对象
Eop- Authorization	是	String	由天翼云官网 accessKey 和 securityKey 经签名后 生成，参与签名生成的 字段包括天翼云官网 accessKey 、 securityKey、平台应 用的appkey（非必须） ，用户请求id（非必须） ），请求时间，终端节 点域名（非必须）以及 请求体内容。		

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” data:image/jpeg;base64,”	-	
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” data:image/jpeg;base64,”	-	

请求代码示例

```
Curl -X POST "https://ai-global.ctapi.ctyun.cn/v1/aiop/api/2f7awxekgvl5/face/compare/PERSON/person/compareFromBase64"
-H "Content-Type: application/json"
-H "ctyun-eop-request-id:33dfa732-b27b-464f-b15a-21ed6845afd5"
-H "appkey:XXX"
-H "version:v2"
-H "Eop-Authorization:XXX"
-H "eop-date:20211109T104641Z"
-H "host:ai-global.ctapi.ctyun.cn"
--data '{"img1Base64":"AAAAAAAAA...", "img2Base64":"BBBBBBBBBBB..."}'
```

返回值说明

1.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
statusCode	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	
returnObj	是	Float	两张人脸的相似度	0.99999	

2.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
statusCode	是	String	错误码，放置API对应的错误码	500001	
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	

参数	是否必填	参数类型	说明	示例	下级对象
error	是	String	返回对应的错误码	AI_OP_500001	

返回值示例

1.请求成功返回值示例

```
{
  "statusCode": "0",
  "message": "success",
  "returnObj": 0.99999}
```

2.请求失败返回值示例

```
{
  "statusCode": "500001",
  "message": "error",
  "details": "请求内容错误"
  "error": "AI_OP_500001"}
```

状态码

http状态码	描述
200	表示请求成功

错误码说明

v2版本错误码：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
AI_OP_400001	请求路径错误	请求路径错误
AI_OP_400002	请求方法错误	请求方法错误，请使用POST请求

错误码	错误信息	错误描述
AI_OP_400003	请求体内容为空	请求体请求数据为空，没有包含内容
AI_OP_400004	请求体非json格式	请求体内容需要符合json要求
AI_OP_400005	请求体类型错误	请求体需为字典，不能为其他类型
AI_OP_400006	必传的参数未传	必传的参数（img1Base64或img2Base64）未传
AI_OP_400008	请求体的参数字段类型错误	img1Base64、img2Base64字段应该是string类型
AI_OP_400009	请求体的参数字段值为空	img1Base64、img2Base64字段值为空字符
AI_OP_400011	base64数据处理异常	img1Base64 或 img2Base64 字段的 base64 字符串转换字节码异常
AI_OP_400012	文件格式不合法	仅支持 jpeg/png/jpg/bmp格式
AI_OP_400013	文件大小不符合要求	该文件大小不符合要求，图片一/图片二要求小于2M
AI_OP_410001	图片一/图片二解码错误	字节码解码为图片错误
AI_OP_500001	服务接口异常，请联系管理员	需要联系管理员处理
AI_OP_601002	图片一/图片二未检测到人脸	上传图片一/图片二不包含人脸

v1版本错误码：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
-----	------	------

错误码	错误信息	错误描述
AI_OP_4101	请求内容错误	传入内容为空，或者传入的参数名错误
AI_OP_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
AI_OP_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
AI_OP_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
AI_OP_4105	图片一未检测到人脸	上传图片一中不包含人脸
AI_OP_4106	图片二未检测到人脸	上传图片二中不包含人脸

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。

6.3.4人脸活体检测

接口描述

用于检测输入图像中的人脸是否为活体。

请求方法

POST

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格式。

URI

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

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	平台应用appkey	562b89493b1a40e1b97ea05e50dd8170	
ctyun-eop-request-id	是	String	用户请求 id，由用户构造，用户可以通过 uuid 等方法自行生成唯一字符串，用于日志请求追踪	33dfa732-b27b-464f-b15a-21ed6845afd5	
eop-date	是	String	请求时间，由用户构造，形如 yyyymmddTHHMMSSZ。	20211221T163014Z	
host	是	String	终端节点域名，固定字段	ai-global.ctapi.ctyun.cn	
Eop-Authorization	是	String	由天翼云官网 accessKey 和 securityKey 经签名后生成，参与签名生成的		

参数	是否必填	参数类型	说明	示例	下级对象
			字段包括天翼云官网 accessKey 、 securityKey、平台应用的appkey（非必须）， 用户请求id（非必须）， 请求时间，终端节点域名（非必须）以及请求体内容。		

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 “data:image/jpeg;base64,”	-	

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...."}'
```

返回值说明

1.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
statusCode	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	
returnObj	是	Object	返回的人脸活体检测结果对象		returnObj

2.表returnObj

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_location	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	-	

3.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
statusCode	是	String	错误码，放置API对应的错误码	4101	

参数	是否必填	参数类型	说明	示例	下级对象
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	
error	是	String	返回对应的错误码	AI_OP_4101	

返回值示例

1.请求成功返回值示例

```
{
  "statusCode": "0",
  "message": "success",
  "returnObj": {
    "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",
    }]
  }
}
```

2.请求失败返回值示例

```
{
  "statusCode": "4101",
  "message": "error",
  "details": "请求内容错误"
  "error": "AI_OP_4101"}
}
```

状态码

http状态码	描述
200	表示请求成功

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
AI_OP_4101	请求内容错误	传入内容为空，或者传入的参数名错误
AI_OP_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
AI_OP_4103	图片大小超过2M	图片大小超过2M
AI_OP_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
AI_OP_4105	未检测到人脸	上传图片中不包含人脸
AI_OP_5000	服务内部错误	接口服务出现未知错误

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().toString()` 函数进行编码。

6.3.5是否戴口罩识别

接口描述

用于检测输入图像中的人脸是否戴口罩。

请求方法

POST

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格式。

URI

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

请求参数

1.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	平台应用appkey	562b89493b1a40e1b97ea05e50dd8170	
ctyun-eop-request-id	是	String	用户请求 id，由用户构造，用户可以通过 uuid 等方法自行生成唯一字符串，用于日志请求追踪	33dfa732-b27b-464f-b15a-21ed6845afd5	
eop-date	是	String	请求时间，由用户构造，形如 yyyymmddTHHMMSSZ。	20211221T163014Z	
host	是	String	终端节点域名，固定字段	ai-global.ctapi.ctyun.cn	
Eop-Authorization	是	String	由天翼云官网 accessKey 和 securityKey 经签名后生成，参与签名生成的字段包括天翼云官网 accessKey 、		

参数	是否必填	参数类型	说明	示例	下级对象
			securityKey、平台应用的appkey（非必须），用户请求id（非必须），请求时间，终端节点域名（非必须）以及请求体内容。		

2.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“data:image/jpeg;base64,”	-	

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...."}'
```

返回值说明

1.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

参数	是否必填	参数类型	说明	示例	下级对象
statusCode	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	
returnObj	是	Object	返回的是否戴口罩识别结果对象		returnObj

2.表returnObj

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_location	是	Map	人脸所处位置	-	
face_list[].Mask	是	Bool	是否佩戴口罩	-	

3.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
statusCode	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	
error	是	String	返回对应的错误码	AI_OP_4101	

返回值示例

1.请求成功返回值示例

```
{
  "statusCode": "0",
  "message": "success",
  "returnObj":{
    "face_num" : 1,
    "face_list" : [{
      "face_location" : {
        "top" : 36,
        "left" : 48,
        "width" : 58,
        "height" : 72
      },
      "Mask" : false
    }]
  }
}
```

2.请求失败返回值示例

```
{
  "statusCode": "4101",
  "message": "error",
  "details": "请求内容错误"
  "error": "AI_OP_4101"}
}
```

状态码

http状态码	描述
200	表示请求成功

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
AI_OP_4101	请求内容错误	传入内容为空，或者传入的参数名错误
AI_OP_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
AI_OP_4103	图片大小超过2M	图片大小超过2M
AI_OP_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
AI_OP_4105	未检测到人脸	上传图片中不包含人脸

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。

7 常见问题

7.1 计费类

调用量在哪里查看

调用明细您可在控制台内查看，进入控制台选择【能力列表】里面对应能力的条目，点击【使用详情】按钮即可查看。

- 查看路径：控制台—能力列表—使用详情。
- 查看步骤：
 1. 在产品详情页，点击【管理控制台】，进入控制台。
 2. 点击左侧【能力列表】，选择希望查看的原子能力。
 3. 点击【使用详情】按钮即可查看已调用量、剩余可调用量、有效期。

调用量的抵扣顺序是？

- 调用量的扣减顺序为：购买资源包后，将按照资源包下单顺序抵扣额度。

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。
- 相关人脸识别API资源包价格表如下，点击立即购买即可跳转到购买窗口。

API名称	100万次	1,000万次	5,000万次	10,000万次
人脸检测	450元	4,100元	17,500元	25,000元
人脸属性识别	450元	4,100元	17,500元	25,000元
人脸比对	3,100元	30,400元	142,500元	255,000元
是否戴口罩识别（公测）	0元	0元	0元	0元
人脸活体检测（公测）	0元	0元	0元	0元

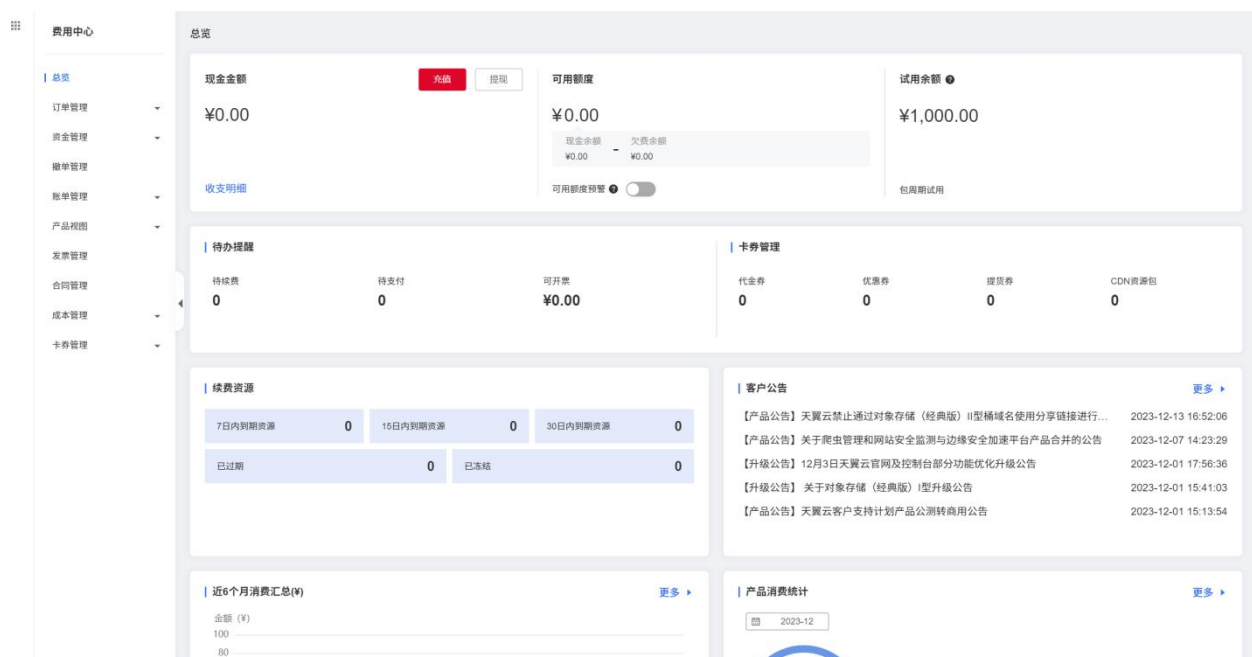
注意：表格价格仅供参考，具体购买价格以价格发文为准

什么情况下会扣资源包次数，识别报错会扣费吗？

- 资源包扣费规则按照：成功调用算入计费次数，若因网络错误未成功调用则不计算次数，其他错误则计算调用次数。
- 计算规则：单张图片算作一次调用。
- 若您的图片上存在多张图片，可能会导致对应接口识别报错，建议将所需图片自行拆解成单张图片进行调用识别。

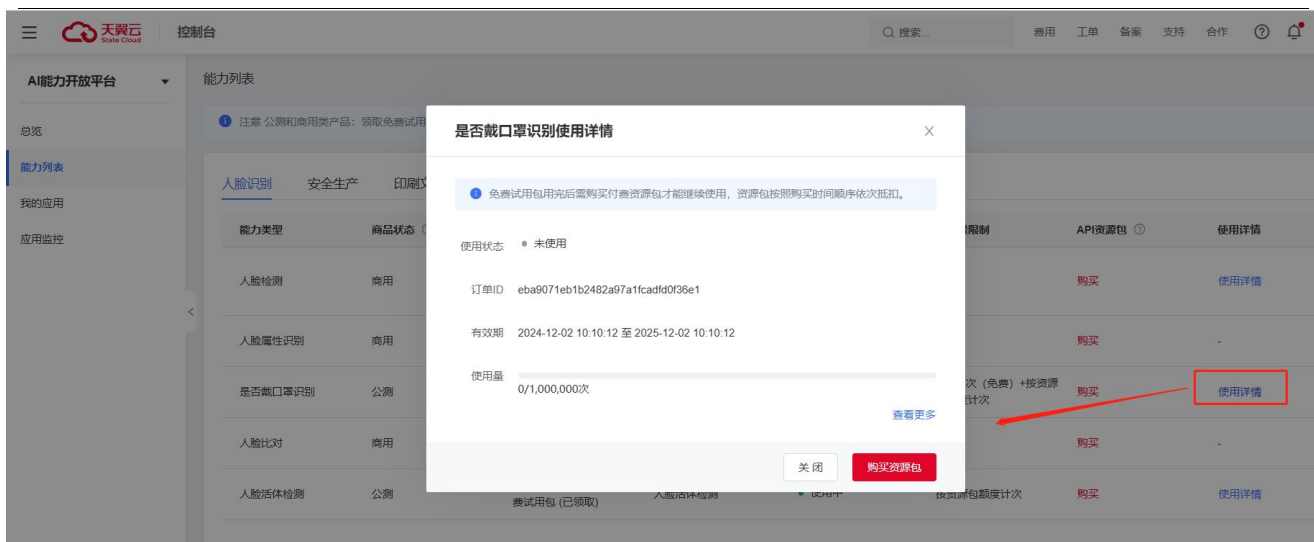
子账号产生的调用在哪里查看账单？

子账号所产生的费用均会归类为主账号名下，具体账单信息可登录天翼云官网费用中心查看。



人脸识别服务剩余次数如何查询？

- 如您已购买人脸识别资源包，资源包剩余次数可至控制台内查看，进入控制台选择对应能力的条目，点击【使用详情】按钮查询。
- 限免产品调用量不受限制，暂不支持查询剩余次数。
- 如您的资源包已消耗完毕，会显示“已用完”；如资源包已超过使用日期，会显示“已过期”；如资源包已退订，会显示“已退订”。



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。

人脸识别服务开通后能否关闭？不使用是否会产生费用？

- 人脸识别服务开通后无法关闭，如您后续无使用需求，不调用已开通的接口即无相关费用产生。
- 同时，请您保管好AppKey和AppSecret信息，保证后期无人调用。

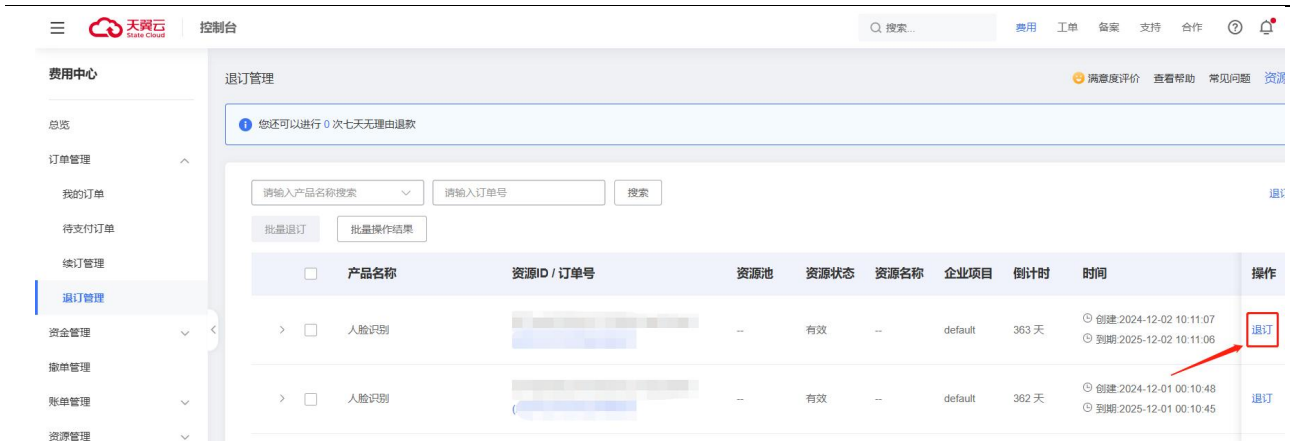
7.2 购买类

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，按照资源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：控制台—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。



资源包使用到一半可以退吗？

- 资源包一经售卖且产生正常调用抵扣，不允许进行退款。
- 资源包属于预付费产品，建议您在购买资源包的时候按实际调用情况预估调用量以免造成资源浪费。

人脸识别服务是否支持代金券付款？

- 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？

- 商用产品暂不支持试用，公测产品可0元订购使用。

7.3 操作类

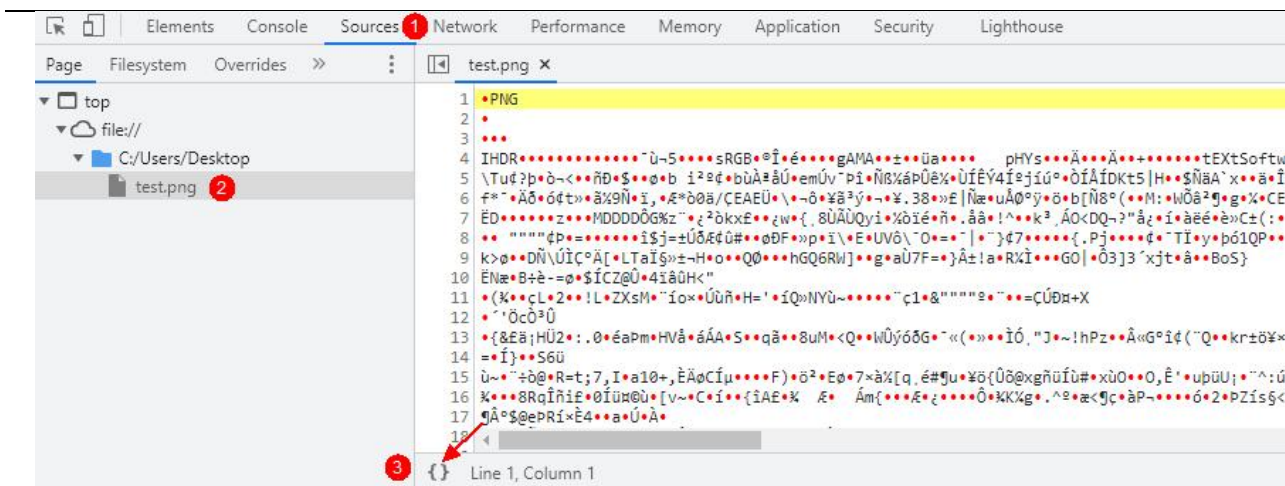
人脸识别在什么时候进入可使用状态？

- 限免类产品：将该产品加入到应用后，即可进入使用状态。
- 公测及商用类产品：将该产品加入到应用，并购买相应资源包，方可进入使用状态。

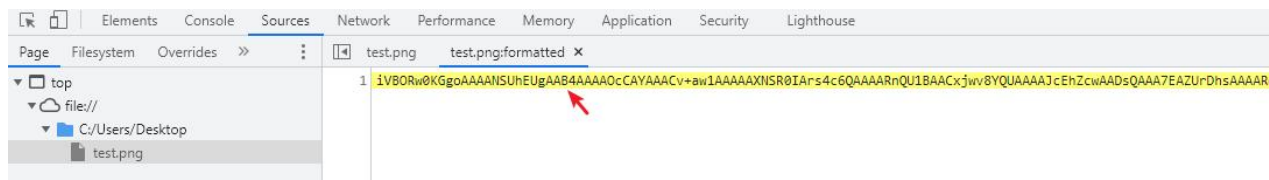
如何获取图片base64编码？

以Chrome浏览器为例作以下说明：

- 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？

- 人脸识别服务目前只支持调用一次接口识别一张图片（人脸比对调用一次识别一组两张图片）。
- 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。

7.4 使用限制

人脸识别对于上传的图片是否有要求？

- 图片单张大小不超过 2MB。
- 图片格式支持jpg、jpeg、png、bmp。

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，因此需上传2张人脸图片。

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）。

人脸识别能否提供100%识别准确率？

- 人脸识别准确率与上传的图片质量相关，同时也存在一定概率的误差，无法做到100%识别准确率。
- 如您对当前使用的人脸识别产品服务有识别准确率相关问题，您可通过天翼云官网工单或者拨打客服电话【400-810-9889】联系我们。

QPS不够用怎么办？

- 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。

人脸识别接口支持哪些格式？

- 人脸识别接口均支持图片格式数据，包括：jpg、jpeg、png、bmp。
- 每个API支持格式可能存在差异，详细请看API文档。
- 请求地址：产品文档—API参考—API，选择具体的产品查看格式要求。



人脸识别服务是否支持私有化部署？

- 人脸识别支持私有化部署。
- 您可通过天翼云官网工单或者客服电话【400-810-9889转1】沟通私有化部署相关合作。

7.5 安全相关

是否可以设置IP白名单呢？

- 人脸识别是API服务，暂不支持白名单设置，您可以在自己的服务器上调用我们的服务。
- 调用过程中如遇到问题，可通过天翼云官网工单或者客服电话【400-810-9889】沟通。

人脸识别传输的数据是否经过加密呢？

- 天翼云人脸识别采用天翼云官网标准EOP网关，在整个数据传输过程中，从源端到目的端，都采取了一系列的安全措施，包括身份验证、加密技术和授权机制、数据完整性保护机制和数据恢复机制等，这些措施可以确保数据传输的安全性和可靠性，提高系统的安全性。
- 若您的数据比较敏感、安全保密性要求高的话，可考虑使用私有化部署。

使用人脸识别服务，图片数据是否会存储？

- 天翼云一直注重保护客户隐私，图片数据不会存储。
- 同时因相关规定限制，人脸识别公有云服务数据不落盘，用户的原始图片和识别数据均不作保留，使用人脸识别服务后，图片数据就会立即释放，不会存储。