



天翼云印刷文字识别

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	4
1.3 产品特性	6
1.4 应用场景	7
1.5 术语解释	7
2 计费说明	10
2.1 计费模式	11
2.2 产品续订	13
2.3 产品退订	15
3 快速入门	16
3.1 入门指引	16
3.2 服务创建	16
3.3 入门操作	18
3.4 入门实践	19
4 用户指南	20
4.1 产品购买	20
4.2 创建应用及开通应用	23
4.3 用户控制台	27
4.4 API调用指南	30

5 最佳实践	35
5.1 通用型OCR	35
5.2 身份证识别	38
6 API 参考	40
6.1 调用前必知	41
6.2 如何调用 API	43
6.3 API	58
6.4 更新历史	95
7 常见问题	96
7.1 计费类	96
7.2 购买类	99
7.3 操作类	100
7.4 使用限制	101

1 产品简介

1.1 产品定义

印刷文字识别(OCR, Optical Character Recognition)是一种技术,能够检测图片中的打印字符,并将其转换为可编辑的文本格式,最后以JSON格式返回识别结果。

OCR服务以开放API的方式提供给用户,用户可以使用编程语言如Python和Java调用OCR服务API,将图片转换为文字,进而实现自动化关键数据的采集,提升业务效率。

OCR服务面向企业用户,提供以下能力:

通用型的OCR:支持网络图片上文字的自动化识别,以实现图片文字的提取。

证件类的OCR:支持身份证、营业执照等证件图片上有效信息的自动识别和关键字段结构化提取。

行业类的OCR:支持多种行业特定图片的结构化信息提取和识别,如车牌、行驶证、驾驶证等,进一步帮助企业实现自动化效率提升。

1.2 产品优势

识别精度高

通过海量标注样本训练,配合国内领先的人工智能算法,针对文字识别的特点,能够快速、准确地处理输入的图像数据,提取出关键的文字信息,满足不同业务场景下的文字识别需求,识别精度高。

数据安全

天翼云印刷文字识别通过先进的算法和训练模型，大大减少了人工干预，能够自动处理和纠正一些常见的误识问题，同时降低隐私风险，保护数据资产。此外在文字识别过程中，采用了一系列加密和安全传输措施，确保用户数据在传输和存储过程中的安全。

快速高效

传统的文字识别方法往往依赖于人工操作，不仅效率低下，而且容易出错。天翼云印刷文字识别产品提供24小时全天候服务，其识别速度高于人工10倍以上。无论白天还是夜晚，都可以随时上传图像进行文字识别。这种不间断的服务模式确保了用户在任何时间都能获得及时的响应，大大提高了工作效率。

使用便捷

为了满足不同用户的需求，我们提供了标准API接口，使用户可以轻松地将印刷文字识别集成到现有的业务流程或系统中。这种便捷的集成方式降低了技术门槛，减少了开发成本，使产品能够快速适应各种应用场景。

使用场景丰富

印刷文字识别产品在各个行业和领域中都有广泛的应用。根据客户需求，精心设计了车牌识别、身份证识别、营业执照识别等使用场景，以满足不同领域的文字识别需求，提供高效、准确的文字识别解决方案。

服务质量稳定

依托天翼云平台，我们通过一系列的措施和技术手段，确保服务的稳定性和可用性，为用户提供可靠、高效的产品和服务。随着技术的不断进步和应用场景的不断拓展，我们将继续致力于提升产品的性能和服务品质。

部署方式灵活

可以提供公共云、私有云的部署方式，满足客户部署需求。

对于希望利用云服务优势、注重灵活性和扩展性的客户，我们提供公共云部署方式。通过将印刷文字识别产品部署在公有云平台上，客户可以快速接入服务。

对于注重数据安全和隐私保护的客户，我们提供私有云部署方式。通过将印刷文字识别产品部署在本地私有云平台上，客户可以在保证数据安全的同时，享有产品带来的便利。

1.3 产品特性

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。支持网络图片上文字信息的自动化识别，快速实现图片上文字的提取。

证件类OCR

支持身份证、营业执照的识别，可以对图片上有效信息进行自动检测和关键字段结构化提取。

- **身份证识别**

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

- **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。

- **行驶证识别**

识别行驶证主页图片的内容，并结构化输出号牌号码、车辆类型、所有人、住址、使用性质、品牌型号、车辆识别代号等信息。

- **驾驶证识别**

识别驾驶证主页图片的内容，并结构化输出驾驶证号、姓名、性别、国籍、住址、出生日期等信息。

行业类OCR

支持车牌等多种行业特定类型图片的结构化信息提取和识别，助力行业自动化效率提升

。

- **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。

1.4 应用场景

各类产品的应用场景

- **通用OCR识别应用场景**
网络图片文字的提取，客户可提取图片上的文字，进行内容监管。
- **车牌识别应用场景**
停车场及社区出入口，高速公路收费站，城市交通、公路卡口道路。
卡证识别应用场景
- **用户注册，交通出行，身份认证。**

各类场景中的可以实现的应用功能

- **办公场景**
网络图片的文字提取。
- **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。

1.5 术语解释

OCR

Optical Character Recognition, 光学字符识别, 一种通过扫描后将文本转换为可编辑数字文本的技术。通过使用图像处理和机器学习等技术, OCR可以将印刷文本转换为计算机可读的数字文本, 从而实现对文本的数字化处理和自动化识别。

字符集

Character set, 用于表示所有可识别字符的特定字符集合。每个字符集包含的字符个数不同, 有些字符集只包含基本的字母和数字, 而有些则包含大量的符号和特殊字符。

分段

Segmentation, 图像预处理的一个步骤, 将文本块从整个图像中分离出来。

二值化

Binarization, 将文本图像转换为只包含黑色和白色像素的图像。这种处理方法的优点是能够简化图像处理过程, 提高处理效率。

预处理

Preprocessing, 识别前对文本图像进行处理的过程, 包括去除噪声、增强对比度等。

切割

Snapping, 将图像中的字符切割成单独的字符, 方便后续的字符识别和处理。

识别率

Recognition rate, OCR系统正确识别字符的比率。识别率的计算公式为: (正确识别的字符数 / 总字符数) × 100%。识别率越高, 说明OCR系统的性能越好, 识别效果越准确。在实际应用中, 识别率是评价OCR系统性能的重要标准之一, 也是用户选择OCR系统的重要依据之一。

矩形化

Rectification, 对文本图像进行矫正, 使字符出现在一个水平或垂直的矩形内。通过矩形化处理, 文本图像中的字符可以被统一到一个标准的排列方式, 有助于提高OCR系统的识别准确率和处理速度。同时, 矩形化处理还可以消除由于拍摄角度、纸张弯曲等因素导致的字符倾斜、扭曲等问题, 使得后续的字符识别更加准确和可靠。

特征提取

Feature extraction, 指从原始图像中提取出一些具有代表性的特征, 这些特征可以用于描述图像的内容和结构。在文本图像识别中, 特征提取是关键步骤之一, 它可以帮助机器更好地理解 and 识别文本信息。

文本框检测

Text frame detection, 在图像中识别和定位文本框的位置和大小的过程, 可以帮助确定需要识别的文本区域, 使OCR只检测感兴趣区域, 从而减少OCR系统的计算量和误差。

字符识别

Character recognition, 使用OCR技术将扫描的文本图像转换为数字文本的过程。

逐字识别

Character by character recognition, OCR系统按照文本图像中字符的顺序, 逐个字符进行识别的过程。在逐字识别中, OCR系统通常会处理单个字符的图像, 因此可以更加细致地处理每个字符的细节和特征。它可以提高字符识别的准确率和可靠性, 并且方便进行校对和修正。

神经网络

Neural network, 一种模拟人脑神经元结构的计算模型, 由多个神经元组成, 通过训练和学习进行模式识别和预测等任务, 常用于OCR中的字符识别。

语言模型

Language Model，对于自然语言处理（NLP）任务，如OCR，区分不同单词的语法和上下文规则的统计模型。

改写

Rewriting，通过重新排列和连接字符来识别单个字符的技术，可以有效地识别出单个字符，并且在遇到不同的字符和字体时具有良好的适应性。

同形异义

Homophone，在OCR中，指具有相同发音但含义不同的单词。

最小编辑距离（Minimum edit distance）：在OCR中，用于计算已知单词和OCR结果之间的差异。

形态学处理

Morphological processing，一种基于数学形态学的图像处理方法，用于分析和处理图像中的形状和结构。在OCR图像处理中，形态学处理可以采用腐蚀、膨胀等方式去除噪声和细节，从而提高OCR识别的准确性和可靠性。

腐蚀操作可以将图像中的每个像素与其周围的像素进行比较，如果周围的像素值比当前像素值小，则将当前像素值降低。腐蚀操作可以用来消除小的噪声点和突出的细节，使图像更加平滑。膨胀操作是将图像中的每个像素与其周围的像素进行比较，如果周围的像素值比当前像素值大，则将当前像素值提高。膨胀操作可以用来扩大图像中的亮区域，使得一些小的连通区域合并成一个大的连通区域。

字符序列

Character sequence，在OCR中，将多个字符组合在一起，以建立单词和句子的概念。

2 计费说明

2.1 计费模式

计费说明如下：

- 印刷文字识别产品采用封顶资源包的计费方式，一些在售卖的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。
- 目前印刷文字识别的身份证识别、通用型OCR、车牌识别、行驶证识别、驾驶证识别这五个能力处于商用状态，需要付费购买才能使用。
- 目前印刷文字识别的营业执照识别暂时为限时免费状态，开通即可使用。
- 为满足客户不同业务使用量需求，每类API设置四档套餐，如：用户评估认为每年应用身份证识别的API调用大概为50万次，则可以购买五个10万次/年的调用资源包，若大概为100万次，则可以购买一个100万次/年的调用资源包。
- 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，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：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。
- 如果当前抵扣额度为付费资源包，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。

通用型OCR计费详情

免费试用包额度

产品名称	免费调用额度	调用有效期
通用型OCR	10,000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

付费资源包价格表

产品名称	调用量规格	调用有效期	单价
通用型OCR	10万次	1年	450元
通用型OCR	100万次	1年	3,200元
通用型OCR	500万次	1年	11,500元
通用型OCR	1,000万次	1年	22,000元

身份证识别计费详情

免费试用包额度

产品名称	免费调用额度	调用有效期
身份证识别	10,000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

付费资源包价格表

产品名称	调用量规格	调用有效期	单价
身份证识别	10万次	1年	1,350元
身份证识别	100万次	1年	9,600元
身份证识别	500万次	1年	39,000元
身份证识别	1,000万次	1年	78,000元

车牌识别计费详情

免费试用包额度

产品名称	免费调用额度	调用有效期
车牌识别	10,000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

付费资源包价格表

产品名称	调用量规格	调用有效期	单价
车牌识别	10万次	1年	1,350元
车牌识别	100万次	1年	9,600元
车牌识别	500万次	1年	39,000元
车牌识别	1,000万次	1年	78,000元

行驶证识别计费详情

免费试用包额度

产品名称	免费试用额度	调用有效期
行驶证识别	10,000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

付费资源包价格表

产品名称	调用量规格	调用有效期	单价
行驶证识别	1万次	1年	300元
行驶证识别	10万次	1年	2800元
行驶证识别	100万次	1年	15000元
行驶证识别	500万次	1年	45000元
行驶证识别	1000万次	1年	80000元

驾驶证识别计费详情

免费试用包额度

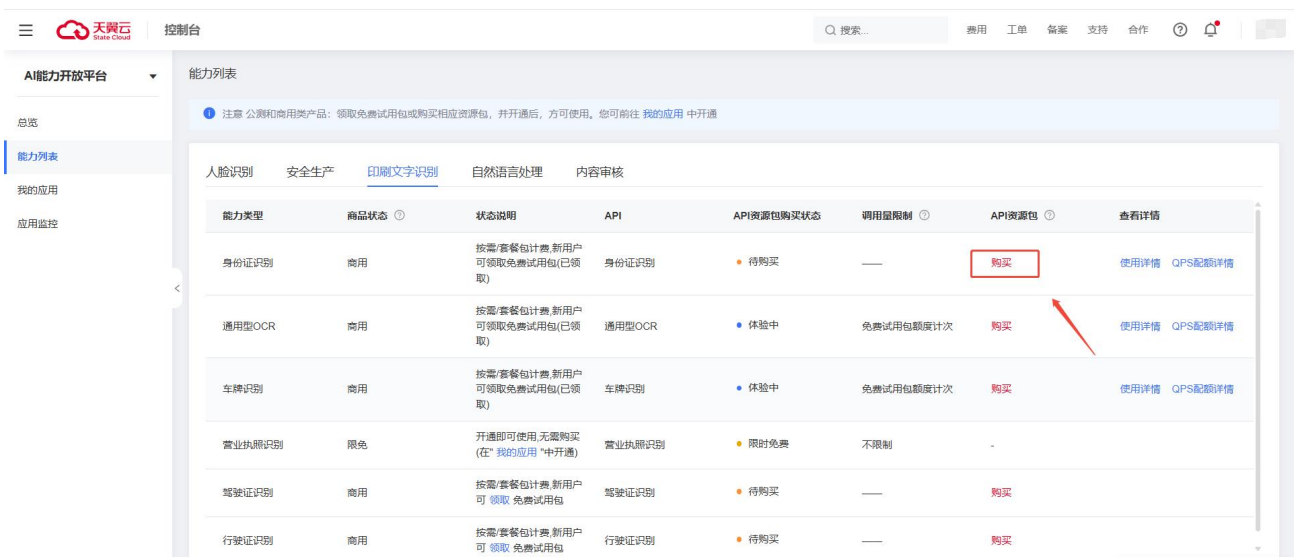
产品名称	免费试用额度	调用有效期
驾驶证识别	10,000次	以单个API为统计维度，有效期从领取之日起一年内有效，过期作废

付费资源包价格表

产品名称	调用量规格	调用有效期	单价
驾驶证识别	1万次	1年	300元
驾驶证识别	10万次	1年	2800元
驾驶证识别	100万次	1年	15000元
驾驶证识别	500万次	1年	45000元
驾驶证识别	1000万次	1年	80000元

2.2 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。



控制台

AI能力开放平台

能力列表

总览

我的应用

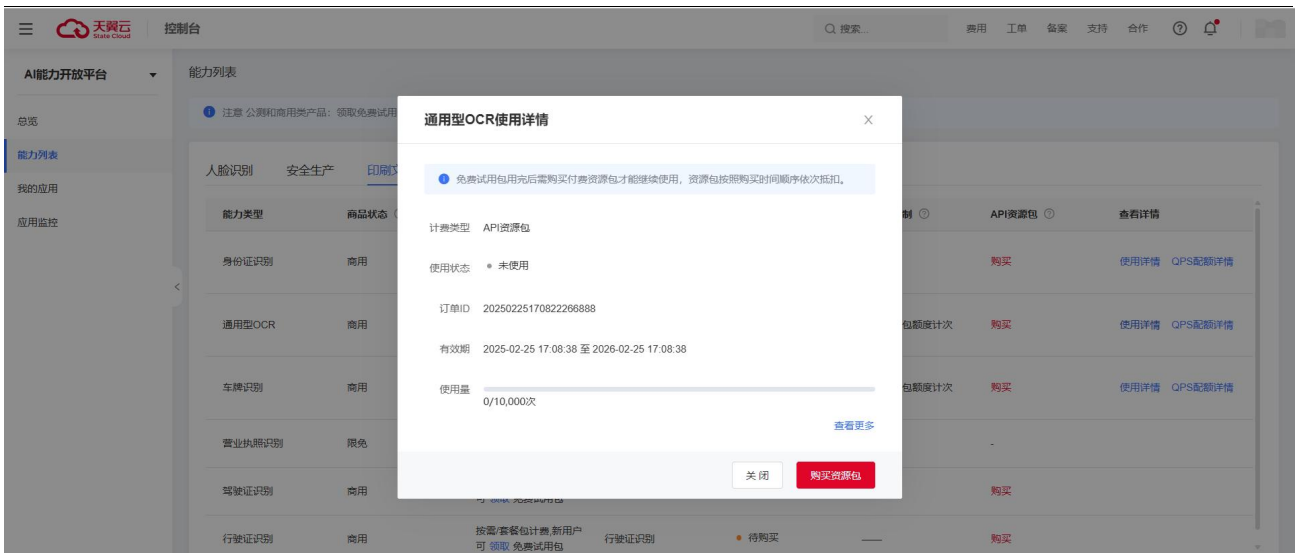
应用监控

能力列表

人脸识别 安全生产 印刷文字识别 自然语言处理 内容审核

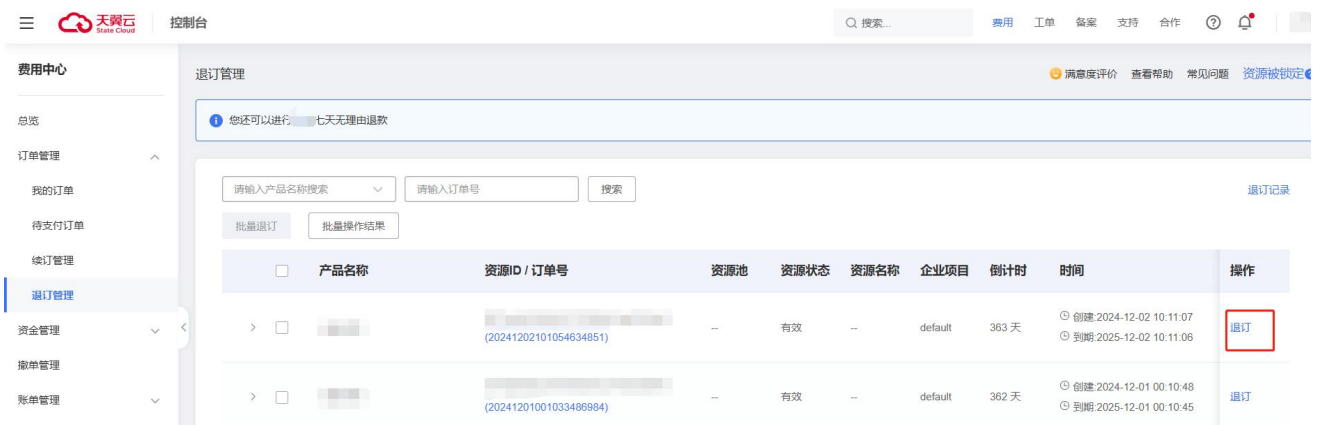
注意 公测和商用类产品：领取免费试用包或购买相应资源包，并开通后，方可使用。您可前往 我的应用 中开通

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	查看详情
身份证识别	商用	按需/套餐包计费, 新用户可领取免费试用包(已领取)	身份证识别	待购买	—	购买	使用详情 QPS配额详情
通用型OCR	商用	按需/套餐包计费, 新用户可领取免费试用包(已领取)	通用型OCR	体验中	免费试用包额度计次	购买	使用详情 QPS配额详情
车牌识别	商用	按需/套餐包计费, 新用户可领取免费试用包(已领取)	车牌识别	体验中	免费试用包额度计次	购买	使用详情 QPS配额详情
营业执照识别	限免	开通即可使用, 无需购买(在“我的应用”中开通)	营业执照识别	限时免费	不限制	-	
驾驶证识别	商用	按需/套餐包计费, 新用户可领取免费试用包	驾驶证识别	待购买	—	购买	
行驶证识别	商用	按需/套餐包计费, 新用户可领取免费试用包	行驶证识别	待购买	—	购买	



2.3 产品退订

- 退订说明：印刷文字识别支持7天未使用的情况下申请退订。
- 退订地址：我的一费用中心—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。

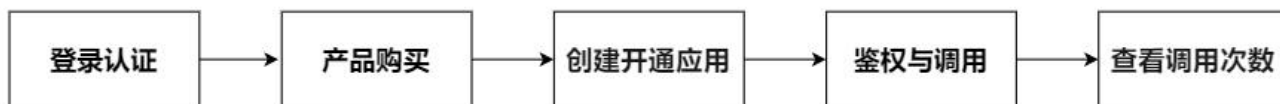


3 快速入门

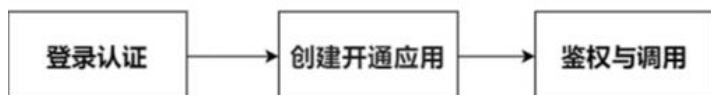
3.1 入门指引

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，提供通用型OCR、车牌识别、身份证识别、营业执照识别四类，满足客户的图片识别需求。

商用产品步骤流程



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

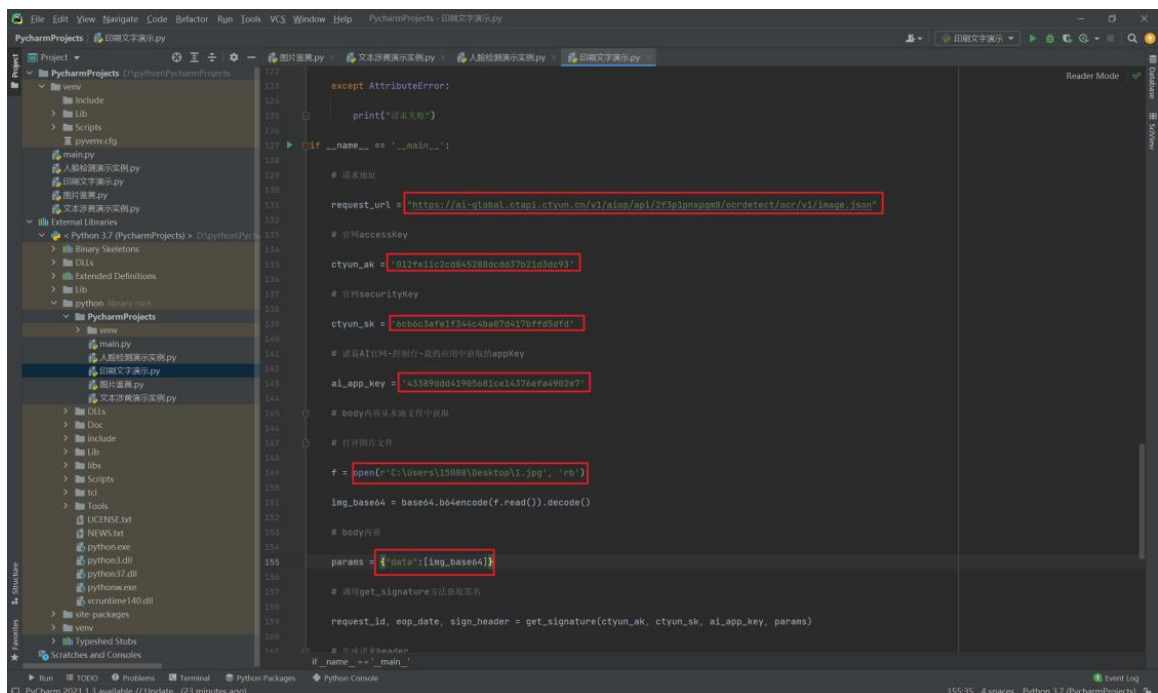
步骤一：鉴权及调用API

以身份证识别为例，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。

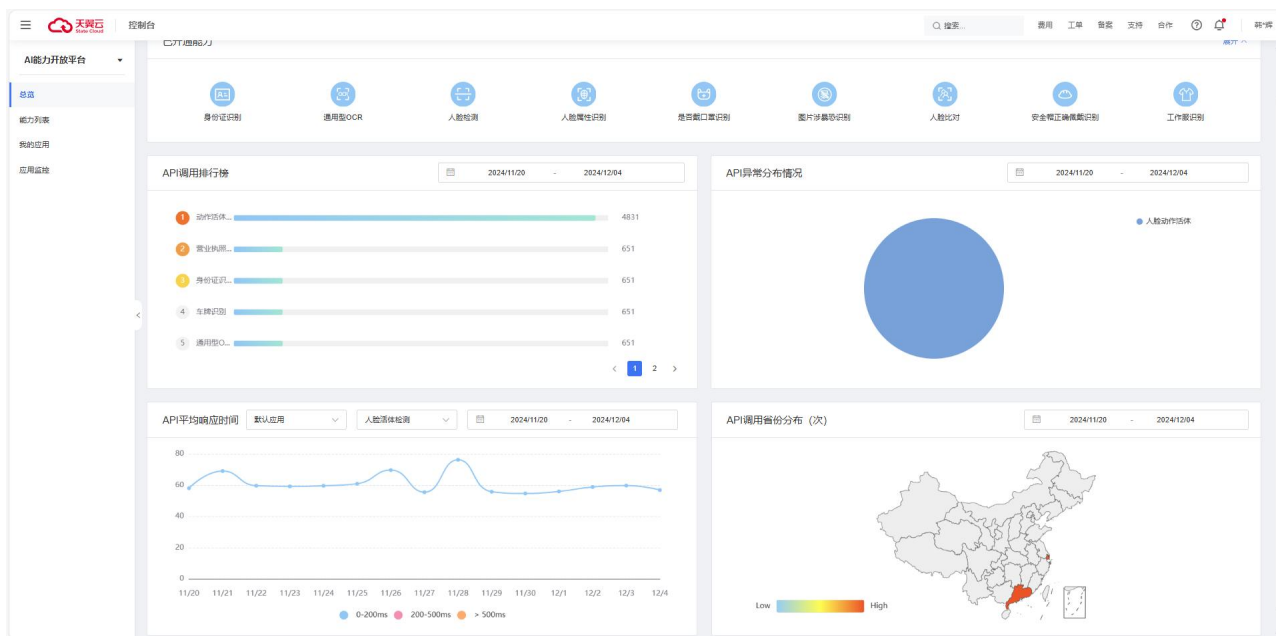
```

D:\python\PycharmProjects\venv\Scripts\python.exe D:/python/PycharmProjects/调用次数统计.py
请求头:
{'Content-Type': 'application/json;charset=UTF-8', 'ctyun-eop-request-id': 'f9644826-7170-11ee-a021-f0d4155a1b84', 'apikey': '43389ddd419b5681ce14376efa4902e7', 'Eop-Authorization': '012fe11c2cd84528'}
200
{'code': 0, 'message': {'success': 1, 'fail': 0}, 'result': [{'text_line': 0, 'detail': []}], 'statusCode': 0, 'returnObj': [{'text_line': 0, 'detail': []}]}
Process finished with exit code 0
  
```

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，包含商用购买、公测产品。
- 能力调用排行榜：时间周期内，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：时间周期内，原子能力异常的分布情况，以饼状图展示。
- 平均响应时间：选取应用分组原子能力和时间周期后，查看其平均响应时间。
- 能力调用省份分布：时间周期内，各省份对能力的调用次数分布情况。鼠标悬停在某一省份时，可查看该省份详细数据信息。



3.4 入门实践

实践产品名称	实践描述	章节目录
--------	------	------

通用型OCR	通过通用型OCR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。	参考文档最佳实践章节
身份证识别	通过身份证识别业务实例，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。	

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。

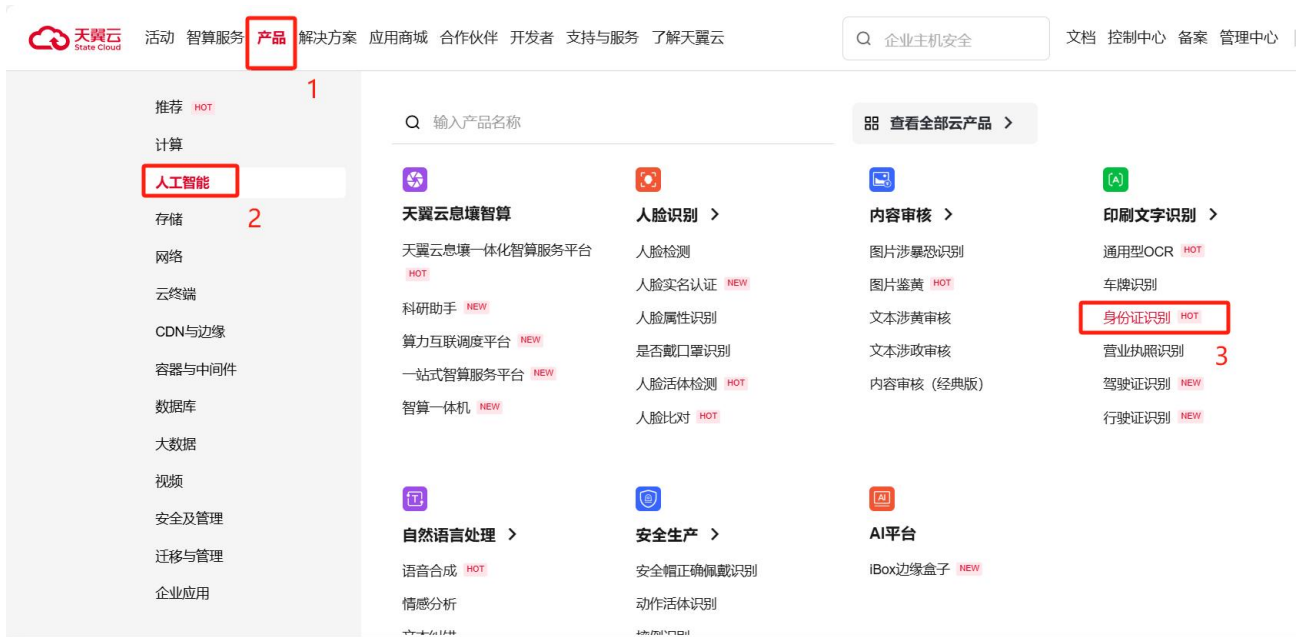
天翼云账号实名认证

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】，根据需求选择印刷文字识别下的对应产品（以身份证识别为例）。



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

身份识别

选择配置

规格	调用有效期	单价	购买数量
10万次	1年	¥1,350.00	- 1 +
100万次	1年	¥9,600.00	- 0 +
500万次	1年	¥39,000.00	- 0 +
1,000万次	1年	¥78,000.00	- 0 +

配置费用 **¥ 1,350.00**

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
身份识别	包周期	10万次	1年	1	¥1,350.00

协议

☒ 我已阅读并同意相关协议 [《天翼云印刷文字识别服务协议》](#)

配置费用 **¥ 1,350.00**

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

费用合计: 1,350.00元

支付方式: ☒ 预付费

账户余额(元)
(账户余额 ¥ 0.00) 无可用账户余额

代金券(元)
(代金券余额 ¥ 0.00) 无可用代金券

优惠券 无可用优惠券

订单费用 + 1,350.00元

账户余额 - 0.00元

代金券 - 0.00元

优惠券 - 0.00元

需支付 1,350.00元

立即支付

我的订单/支付结果

支付成功 返回订单列表

订单号: 20240424101243208983 订单类型: 订购 创建时间: 2024-04-24 10:12:44 更新时间: 2024-04-24 10:13:43

产品	配置	订购数量	所属资源池	周期	金额(元)
产品1	开通中				1,350.00元
印刷文字识别(文字识别)	身份证识别_10万次	1	-	1次	1,350.00元

费用合计: 1,350.00元

支付方式: 后付费

订单费用: + 1,350.00元

4.2 创建应用及开通应用

本文以身份证识别为示例作为说明。

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。

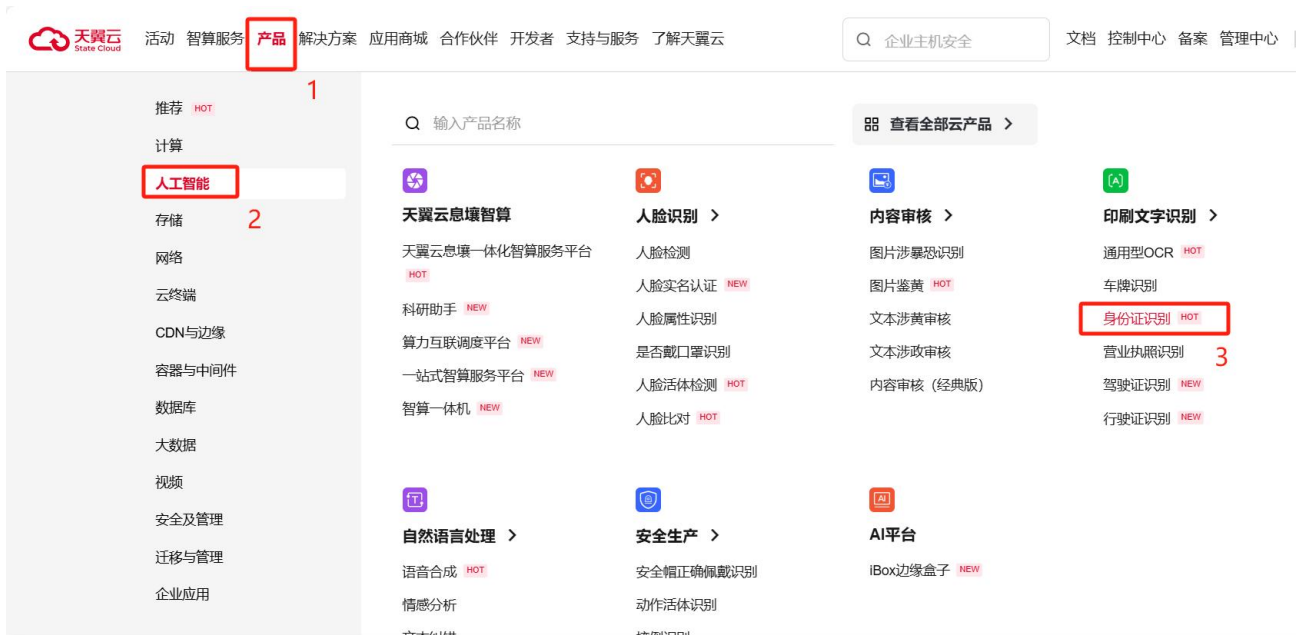
天翼云账号实名认证

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。
- 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】，根据需要选择印刷文字识别下的对应产品（以身份证识别为例）。



2. 跳转到服务详情页后，点击【立即开通】；

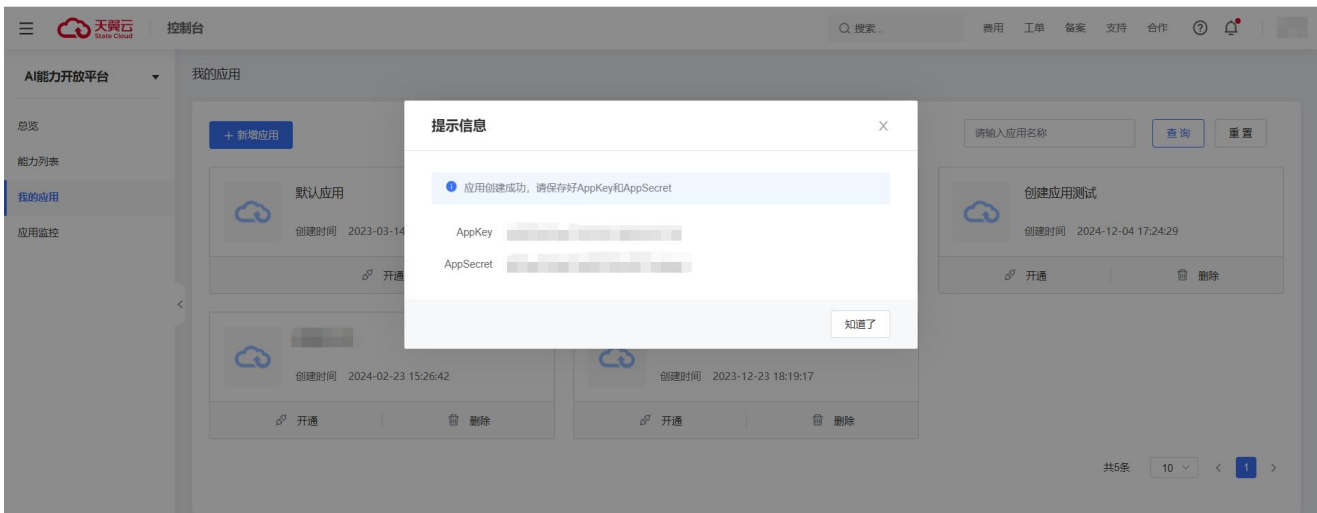


创建新应用

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。

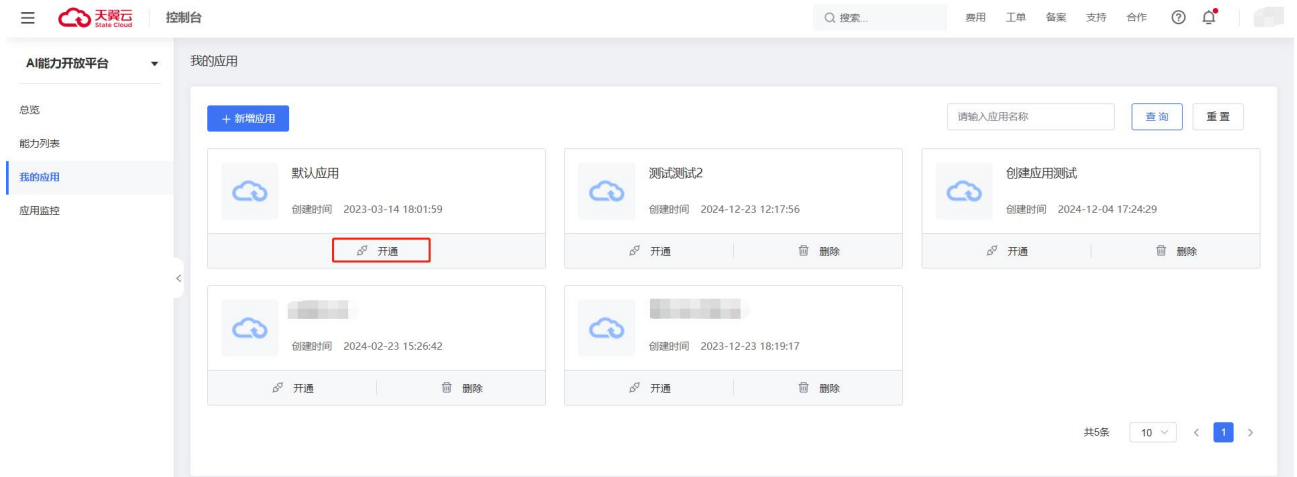


2.再次弹出小弹窗，提示应用创建成功，请保存好AppKey和AppSecret，点击【知道了】即新应用创建成功。

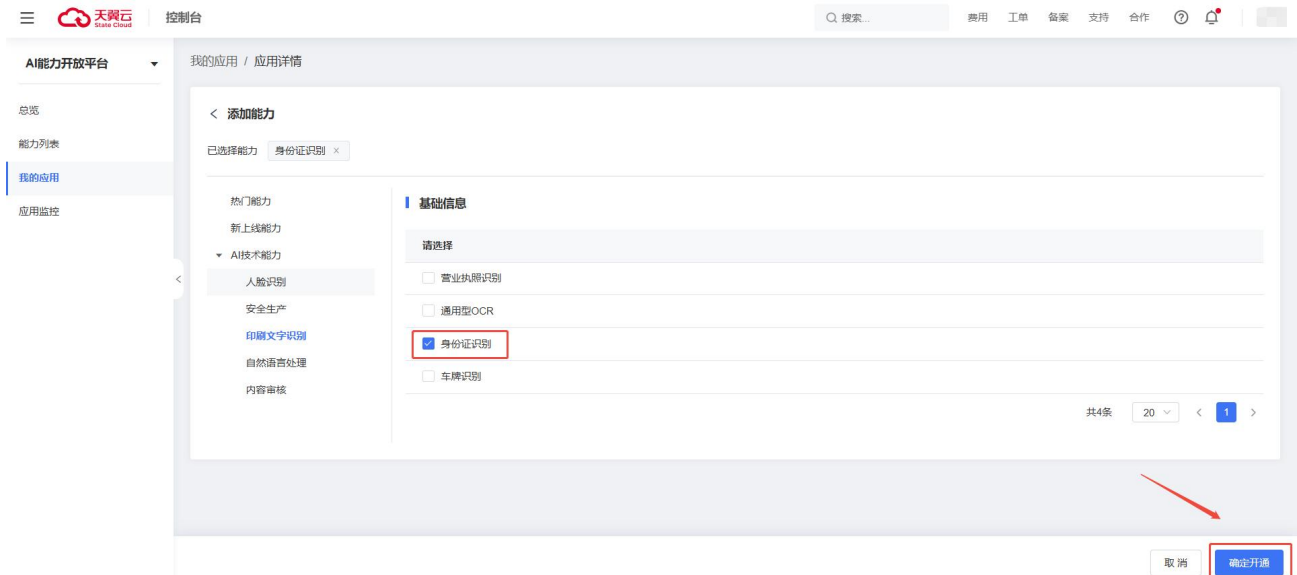
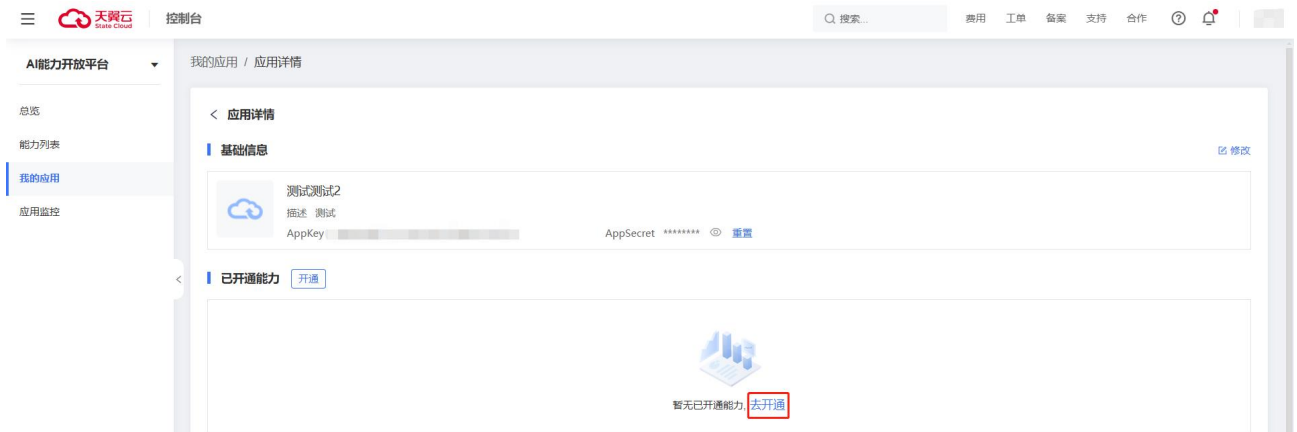


加入应用

1.根据需要选择应用，点击【开通】，跳转到应用详情页。

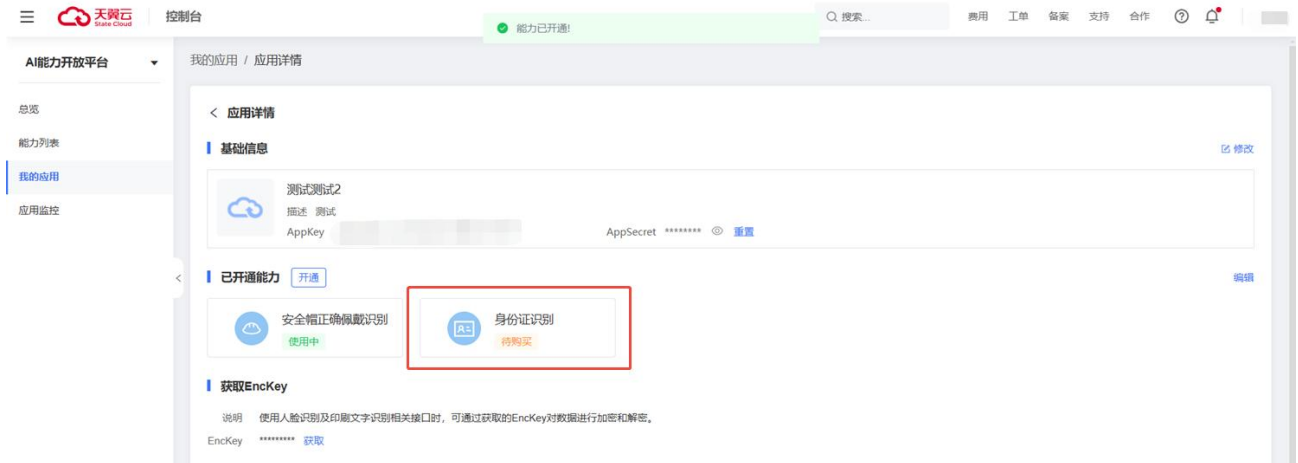


2. 点击【去开通】，添加能力。



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。



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。



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。

身份证识别

识别正反面身份证图片的内容，并结构化输出姓名、性别、民族、出生日期、住址、身份证号、签发机关、有效期限字段。

立即购买

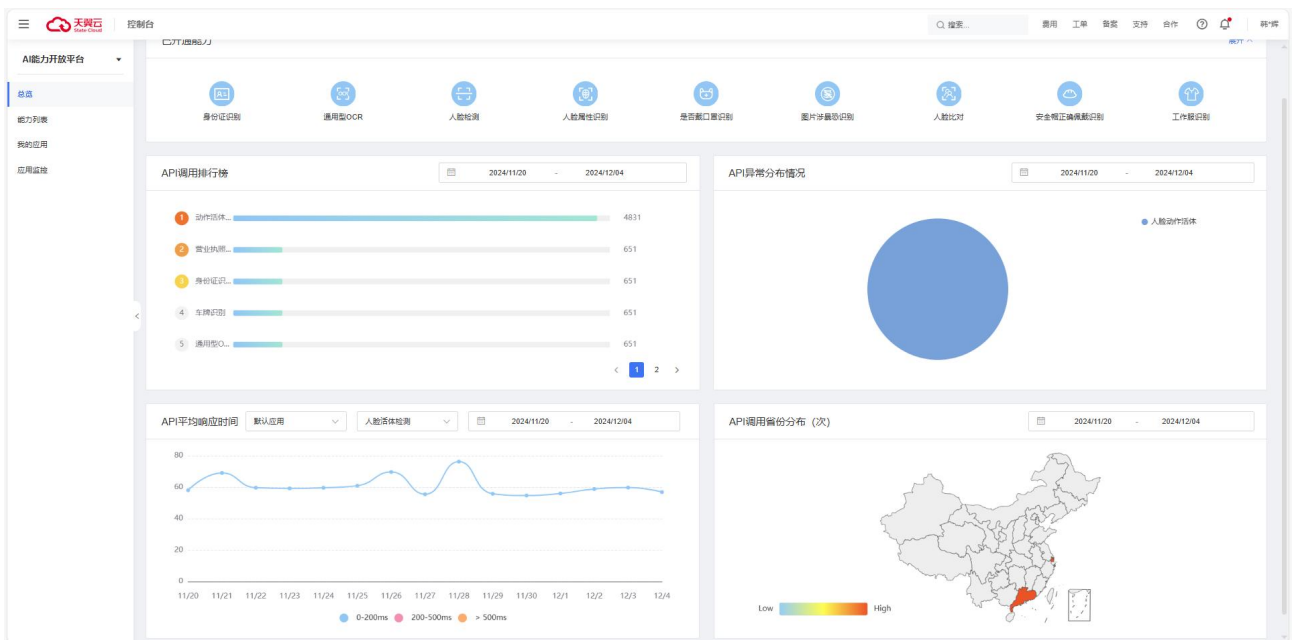
立即开通

管理控制台 >

产品文档 >

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。

控制台

AI能力开放平台

能力列表

注意 公测和商用类产品：领取免费试用包或购买相应资源包，并开通后，方可使用。您可前往 [我的应用](#) 中开通

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	查看详情
身份证识别	商用	按需/套餐包计费,新用户可 领取 免费试用包	身份证识别	待购买	—	购买	
通用型OCR	商用	按需/套餐包计费,新用户可 领取 免费试用包	通用型OCR	待购买	—	购买	
车牌识别	商用	按需/套餐包计费,新用户可 领取 免费试用包	车牌识别	待购买	—	购买	
营业执照识别	限免	开通即可使用,无需购买(在“ 我的应用 ”中开通)	营业执照识别	限时免费	不限制	-	
驾驶证识别	商用	按需/套餐包计费,新用户可 领取 免费试用包	驾驶证识别	待购买	—	购买	
行驶证识别	商用	按需/套餐包计费,新用户可 领取 免费试用包	行驶证识别	待购买	—	购买	

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。

控制台

AI能力开放平台

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 2023-03-14 18:01:59

[开关](#)



测试测试2

创建时间 2024-12-23 12:17:56

[开关](#) [管理](#) [删除](#)



创建应用测试

创建时间 2024-12-04 17:24:29

[开关](#) [管理](#) [删除](#)



应用1

创建时间 2024-02-23 15:26:42

[开关](#) [管理](#) [删除](#)



应用2

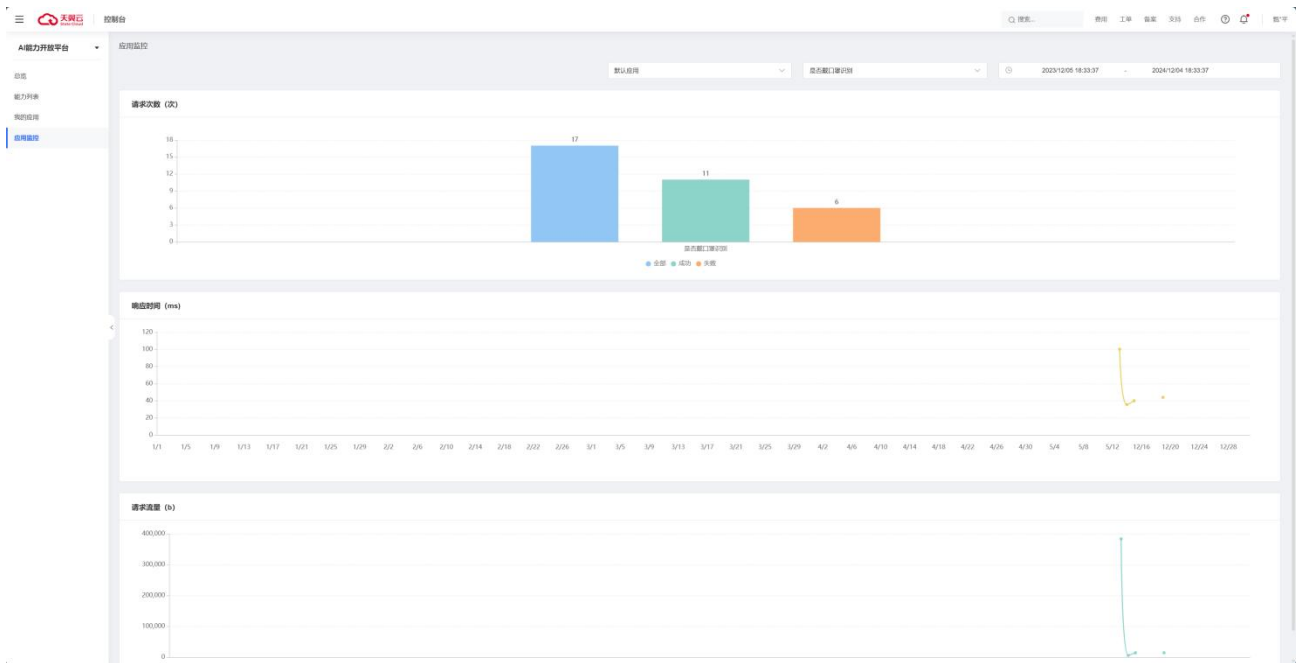
创建时间 2023-12-23 18:19:17

[开关](#) [管理](#) [删除](#)

共5条 [10](#) [1](#)

应用监控

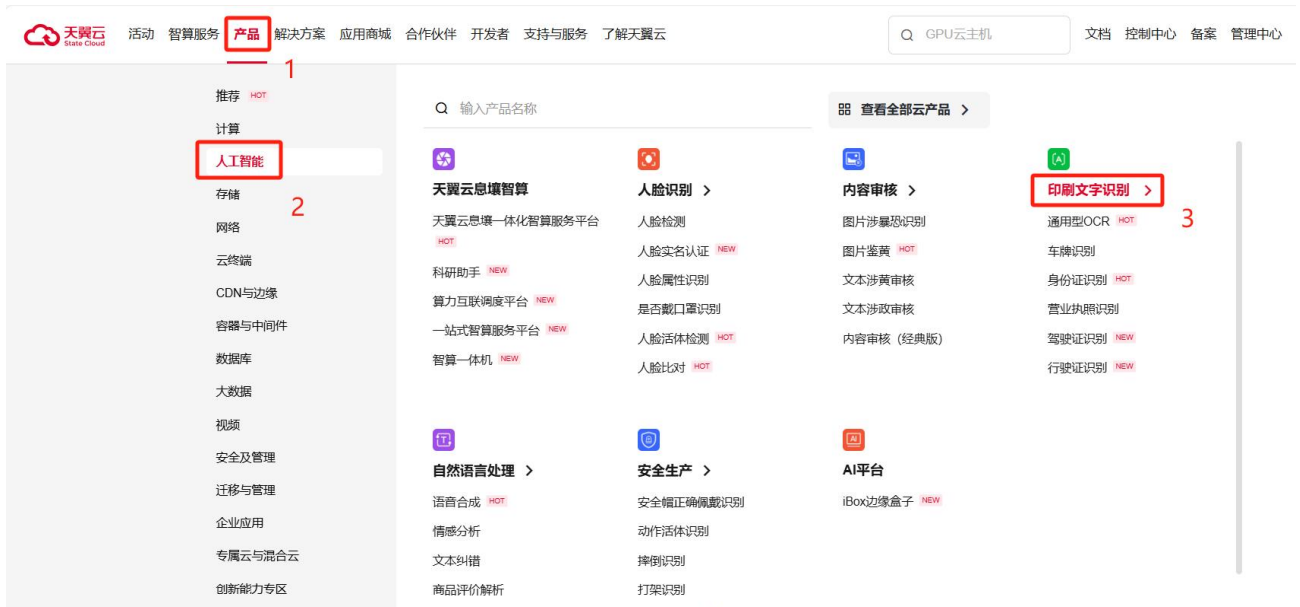
点击左侧菜单栏【应用监控】，可以查看所创建应用的请求次数、响应时间与请求流量。



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【印刷文字识别】，打开对应的产品聚合页。



2.打开帮助中心

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

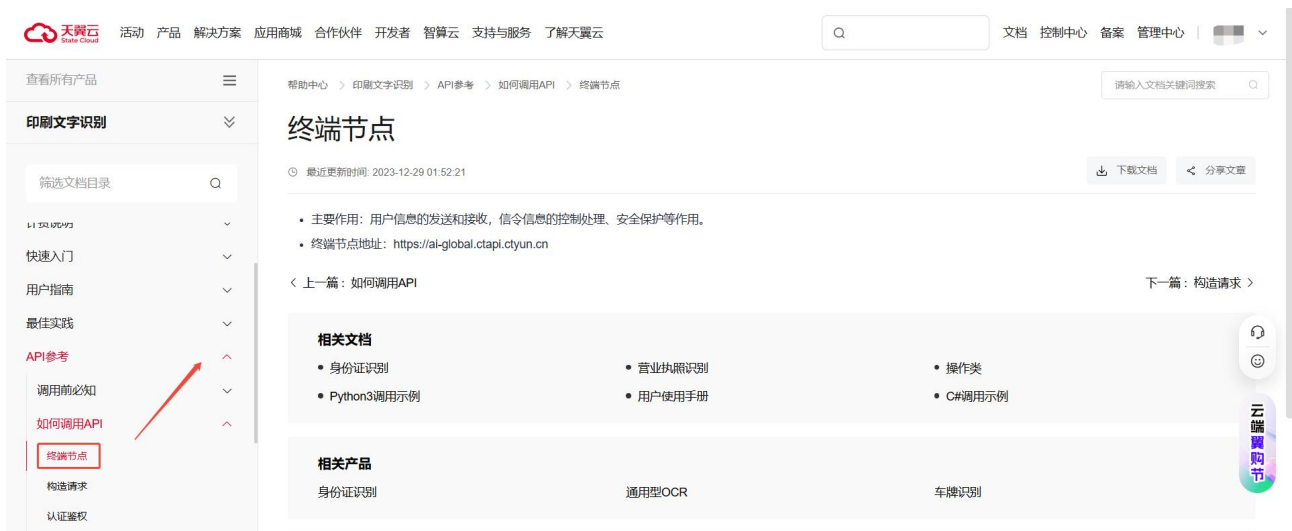


3.查看API的请求地址

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

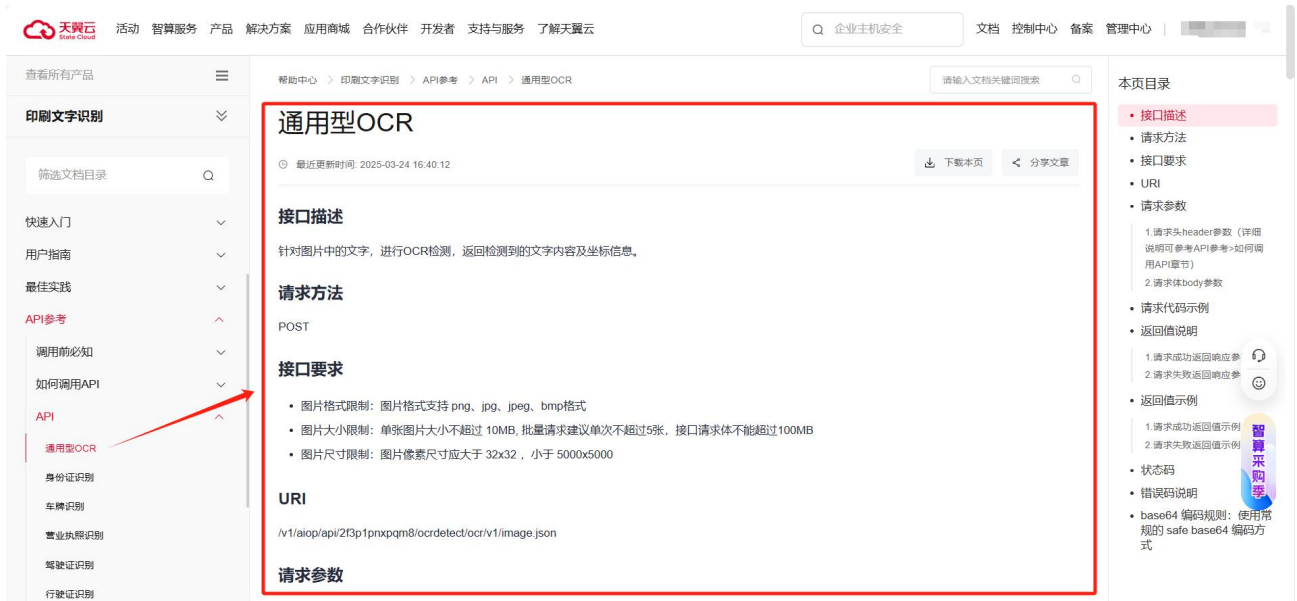
<https://ai-global.ctapi.ctyun.cn>。



3.2对应接口URI

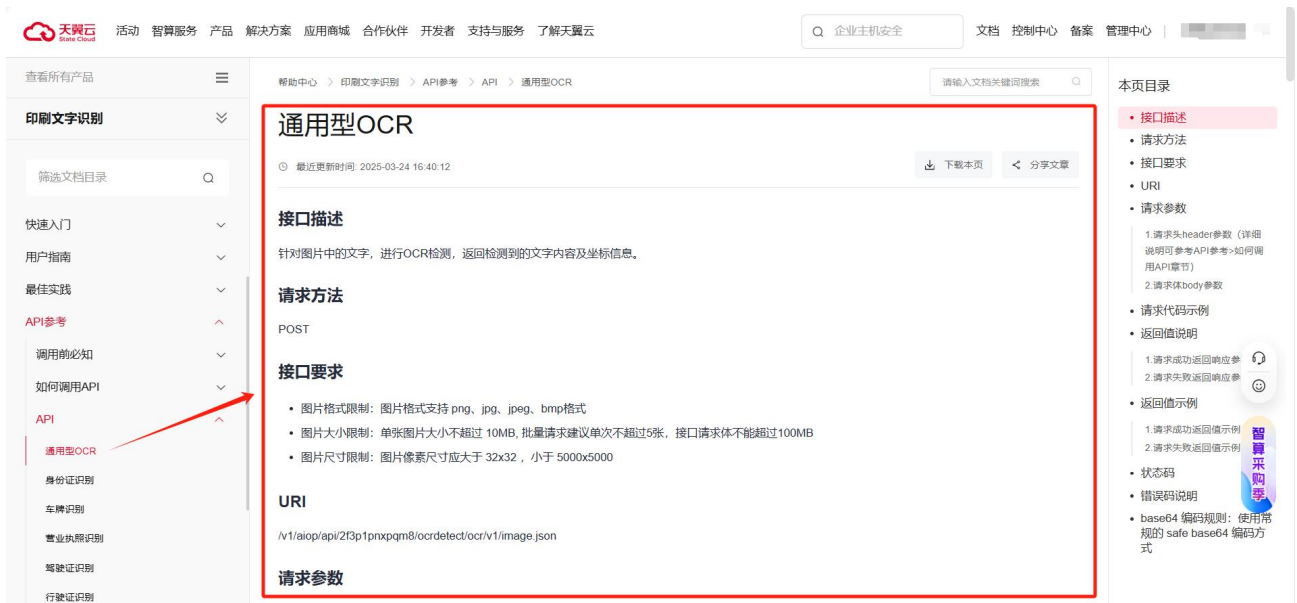
对应印刷文字识别下的具体产品。

以通用型OCR为例，选择【API参考-API-通用型OCR】，右侧即可查看URL。



4. 查看接口文档

以通用型OCR为例：点击【API参考-API】，选择【通用型OCR】，右侧即可查看、请求方法、接口要求、请求URI、请求参数、返回值说明、状态码等信息。



5. 构造请求

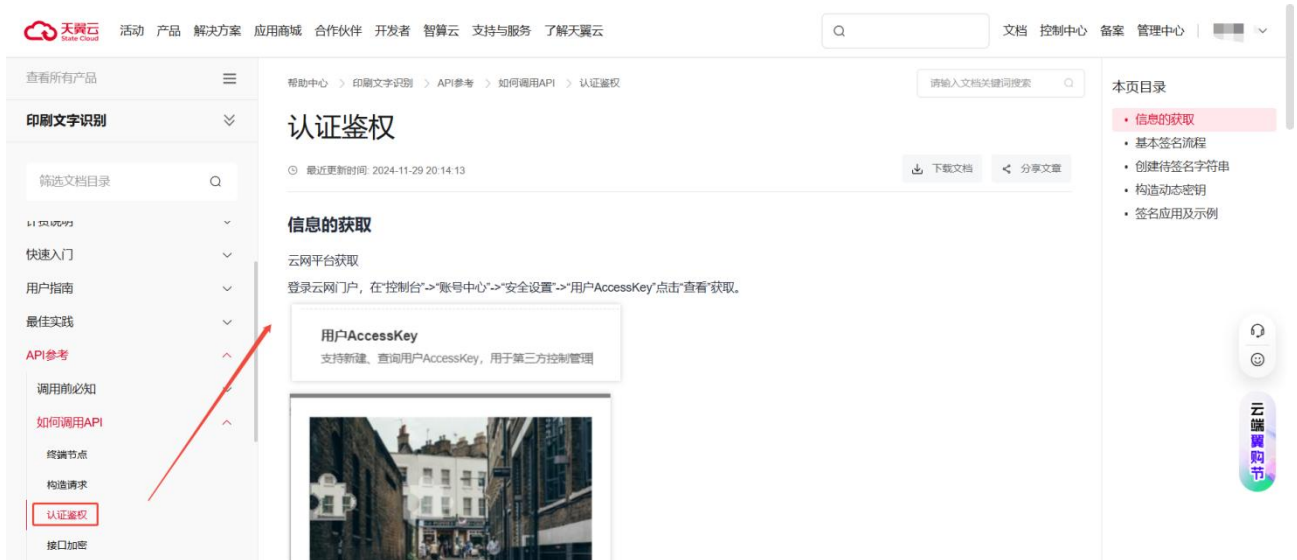
点击【API参考-如何调用API】，选择【构造请求】。



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

印刷文字识别

筛选文档目录

快速入门

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

API

通用型OCR

```
<body>
<center><h1>413 Request Entity Too Large</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

9. 状态码

http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。全局请求返回错误码请参考章节API概览>状态码

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
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



5 最佳实践

5.1 通用型OCR

背景信息

对于首次使用通用型OCR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

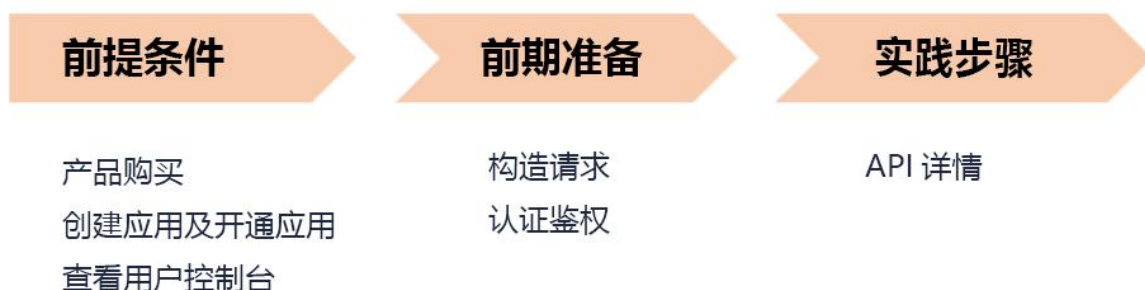
文档数字化：OCR可以将纸质文件转换为数字格式，方便文件存储和信息传输，提高办公效率。

办公扫描：OCR可以将纸质文件扫描为电子文档，方便存储和检索，减少纸张使用。

产品功能

快速、准确的OCR文字检测和识别服务，针对多场景、多颜色的OCR文字检测和识别需求，最终输出检测到的文字内容和文本行坐标信息，解决不易识别、手动输入等场景操作效低下的问题。

实践流程



前提条件

购买通用型OCR相关产品，订购流程详情请查看产品购买；

创建通用型OCR应用及开通通用型OCR应用，开通服务流程详情请查看[创建应用及开通应用](#)；
查看购买通用型OCR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URI}，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/2f3p1pnxpqm8/ocrdetect/ocr/v1/image.json>。

认证鉴权

认证鉴权详细版本，[点击查看认证鉴权](#)。

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\ncctyun-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。

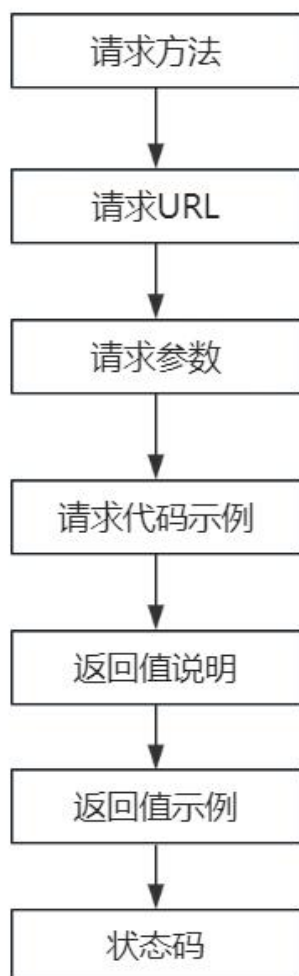
5.签名应用

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_sigtire、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

详情查看API通用型OCR。



5.2 身份证识别

背景信息

对于首次使用身份证识别业务的用户，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

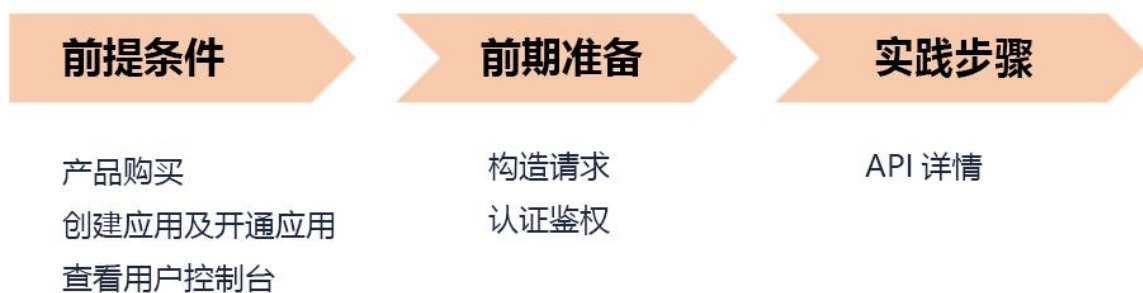
实名认证：在互联网服务、移动应用中，要求用户进行实名认证以确保安全性。身份证识别可以帮助用户快速识别证件信息，减少手动输入错误。

酒店入住：酒店需要在客人登记入住时验证身份信息，身份证识别可以提高办理入住手续的效率。

产品功能

身份证信息智能解析API，可高效准确识别正反面身份证图片的内容，并结构化输出姓名、性别、民族、出生日期、住址、身份证号、签发机关、有效期限共8个字段，同时支持返回头像位置及base64编码，帮助您快速获取身份证信息。

实践流程



前提条件

购买身份证识别相关产品，订购流程详情请查看[产品购买](#)；

创建身份证识别应用，开通服务流程详情请查看[创建应用及开通应用](#)；

查看购买身份证识别产品情况，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URI}，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/2f3os7qq79xc/IdentityCard/ocr/v1/idcard.json。

认证鉴权

认证鉴权详细版本，点击查看认证鉴权。

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。

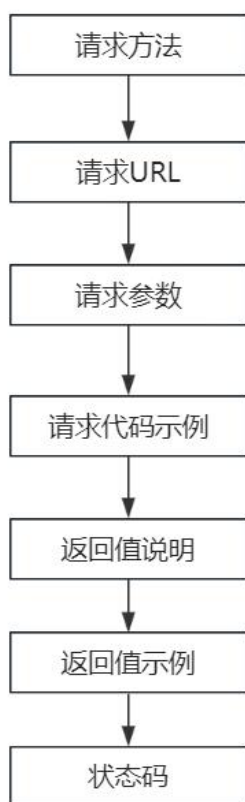
5. 签名应用

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_sigtire、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身份证识别。



6 API 参考

6.1 调用前必知

6.1.1 概述

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，提供通用型OCR、车牌识别、身份证识别、营业执照识别、驾驶证识别、行驶证识别六类，满足客户的图片识别需求。

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。

6.1.2 API 概览

类型	描述
通用型OCR	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，最终返回检测到的文字内容及文本行坐标信息
身份证识别	识别正反面身份证图片的内容，并结构化输出姓名、性别、民族、出生日期、住址、身份证号、签发机关、有效期限等字段
车牌识别	适用于中国大陆的车牌识别服务，支持蓝牌、黄牌（单层）、新能源车牌的检测与识别，同时支持图片内有多张车牌，结构化返回车牌内容及车牌位置坐标
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
驾驶证识别	支持驾驶证正页（主页）图片识别，返回检测到的驾驶证号、姓名、性别、国籍、住址、出生日期、初次领证日期、准驾车型、有效期限
行驶证识别	支持行驶证主页图片识别，返回检测到的号牌号码、车辆类型、所有人、住址、使用性质、品牌型号、车辆识别代号、发动机号码、注册日期、发证日期

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
CTAPI_0009	传入的body不符合json格式

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。

终端节点地址：<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URI}。

终端节点地址：<https://ai-global.ctapi.ctyun.cn>。

对应接口URI：每个API对应一个接口URI，请参考各个接口URI。

通信协议

天翼云API 的接口均通过 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	用户每次请求生成唯一字符串，例如JAVA中UUID，用于定位唯一请求
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”->“个人中心”->“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\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也要进行签名，则待签名的header组合：

```
ccad:123\ncctyun-eop-request-id:123456789\neop-
date:20210531T100101Z\nhost: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 = hmdate)

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，填入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，即可获取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
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)
```

6.3 API

6.3.1 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-Type	是	string	json格式	application/json	
appkey	是	string	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。	3dfa732-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、securityKey、平台的 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
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/2f3p1pnxpm8/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":["AAAAAAAA...."]}'
```

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
statusCode	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	
returnObj	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目，位置以及文本行内容		returnObj

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 returnObj

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
statusCode	是	Int	错误码，参见本文档错误码说明部分	400003	
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	请求体请求数据为空，没有包含内容	
returnObj	否	Object	返回对应的错误编码和错误信息		returnObj

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returnObj

参数	是否必填	参数类型	说明	示例	下级对象
err_code	是	Int	错误码	400011	

err_msg	是	String	错误信息	data 字段的 base64 字符串 转换字节码异 常	
---------	---	--------	------	---------------------------------------	--

8、返回值示例

请求成功返回值示例

```
{
  'statusCode': 0,
  'message': {'success' : 1, 'fail' : 0},
  'returnObj': [
    {
      '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,
```

```
299.9887625014,  
84.2472828771,  
295.8226080086  
]  
}  
]  
}  
]  
}
```

请求失败返回值示例

```
{  
  "statusCode": 400003,  
  "message": "error",  
  "details": "请求体请求数据为空，没有包含内容",  
  "error": "AI_OP_400003"  
}
```

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
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
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	必须的参数（data）未传
AI_OP_400008	请求体的参数字段类型错误	data 字段应该是 list 类型
AI_OP_400009	请求体的参数字段值为空	data 列表为空
AI_OP_400011	base64 数据处理异常	data 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，图片要求小于 10M
AI_OP_400018	超过个数限制	data 超过个数限制，最多传 50 张图片
AI_OP_410001	图片解码错误	字节码解码为图片错误
AI_OP_410002	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
AI_OP_500001	OCR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。

6.3.2 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 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 URI

/v1/aiop/api/2f3os7qq79xc/IdentityCard/ocr/v1/idcard.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	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。	3dfa732-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、securityKey、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
data	是	string	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，不包含前缀 "data:img/jpg;base64,"	_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":["_9j_4AAQSkZJRg..."]}'
```

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	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		message
returnObj	是	List	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、身份证号码等信息		returnObj

表 returnObj

参数	是否必填	参数类 型	说明	示例	下级对象
note	是	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	back	
detail	是	String	返回结果键值集合，包含姓名、性别、身份证号码等信		detail
idcard_face	是	String	此API无法截取人像面身份证的人像子图，此字段固定为'no face'	no face	
demo_image	是	String	无效字段（可忽略）	demo_image	
risk	是	String	此API无法对身份证复印件进行判断，此字段固定为'normal'	normal	

表detail

参数	是否必填	参数类 型	说明	示例	下级对象
name	是	String	姓名	陈XX	
sex	是	String	性别	男	
nation	是	String	民族	汉	
birthday	是	String	出生日期	1995XX25	
idn	是	String	身份证号	3101001995XX25XX16	

addr	是	String	住址	上海市XX区XX路1号	
org	是	String	签发机关	上海市公安局XX分局	
validperiod	是	String	有效期限	2007.07.27-2027.07.27	

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	Int	错误码，放置API对应的错误码	4004	
message	是	String/Object	请求失败时返回值为'error'或{'success':0,'fail':1}	error	message
details	否	String	返回对应的错误描述	data字段图片数据不是list格式	
returnObj	否	List	返回对应的错误编码和错误信息		returnObj
error	是	String	返回对应的错误码	AI_OP_4004	

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returnObj

参数	是否必填	参数类型	说明	示例	下级对象
err_code	是	Int	错误码	4008	
err_msg	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

```
{
  "statusCode": 0,
  "message": {"success": 1, "fail": 0},
  "returnObj": [{
    "note": "front",
    "detail": {
      "name": "",
      "sex": "",
      "nation": "",
      "birthday": "",
      "idn": "",
      "addr": "",
      "org": "兰州市公安局城关分局",
      "validperiod": "2007.07.27-2027.07.27"
    },
    "idcard_face": "no face",
    "demo_image": "demo_image",
    "risk": "normal"
  }]
}
```

请求失败返回值示例

```
{
  "statusCode": 4004,
  "message": "error",
  "details": "data字段图片数据不是list格式",
  "error": "AI_OP_4004"
}

示例2:处理失败返回值示例

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

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
AI_OP_4001	请求JSON处理异常	请求JSON处理异常
AI_OP_4002	body传入的不是字典	body传入的不是字典
AI_OP_4003	请求中未包data字段	请求中未包data字段
AI_OP_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
AI_OP_4005	图片list内图片数量为0	图片list内图片数量为0
AI_OP_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
AI_OP_4008	base64数据处理异常	base64数据处理异常
AI_OP_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
AI_OP_4010	单张图片大小超过10M	单张图片大小超过10M
AI_OP_4011	图片尺寸不符合要求，分辨率长宽尺寸应小于5000大于32	图片尺寸不符合要求，分辨率长宽尺寸应小于5000大于32
AI_OP_4012	请求方法错误，请使用POST请求	请求方法错误，请使用POST请求
AI_OP_4013	无效的请求路径，请确保请求路径正确	无效的请求路径，请确保请求路径正确
AI_OP_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

6.3.3 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 URI

/v1/aiop/api/2gt54ed8660w/driven-plate-ocr/platelicense.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	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	3dfa732-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、securityKey、平台的 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
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..."}}'
```

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
statusCode	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	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	
returnObj	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		returnObj

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returnObj

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	Int	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
statusCode	是	Int	错误码，放置API对应的错	4003	

			误码		
message	是	String/Object	请求失败时返回值为'error'或{'success':0,'fail':1}	error	message
details	是	String	返回对应的错误描述	请求中未包data字段	
returnObj	否	Object	返回对应的错误编码和错误信息		returnObj
error	是	String	返回对应的错误码	AI_OP_4003	

表returnObj

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

```
{
  "statusCode": 0,
  "message": {"success": 1, "fail": 0},
  "returnObj": [{
    "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
    }]
  }]
}
```

请求失败返回值示例

示例1:

```
{
  "statusCode": 4004,
  "message": "error",
  "details": "data字段图片数据不是list格式",
  "error": "AI_OP_4004"
}
```

示例2:处理失败返回值示例

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

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
AI_OP_4001	请求JSON处理异常	请求JSON处理异常
AI_OP_4002	body传入不是字典	body传入不是字典
AI_OP_4003	请求中未包data字段	请求中未包data字段
AI_OP_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
AI_OP_4005	图片list内图片数量为0	图片list内图片数量为0

错误码	错误信息	错误描述
AI_OP_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
AI_OP_4008	base64数据处理异常	base64数据处理异常
AI_OP_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
AI_OP_4010	单张图片大小超过10M	单张图片大小超过10M
AI_OP_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
AI_OP_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

6.3.4 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 URI

/v1/aiop/api/2z0yhhrzgv0g/tts/predict

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	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。	3dfa732-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、securityKey、平台的 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
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/jpeg;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_4QAwRXhpZgAATU0AKgAAA..."}'
```

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
statusCode	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statusCode为0，返回success；如果statusCode非0，则返回对应的可读错误信息	success	
returnObj	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		returnObj

			所等详细键值对信息		
--	--	--	-----------	--	--

表returnObj

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_capital	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
statusCode	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	
error	是	String	返回对应的错误码	AI_OP_400009	

8、返回值示例

请求成功返回值示例

```
{
  "statusCode": 0,
  "message": "success",
  "returnObj": {
    "license_type": "Business License",
    "detail": {
      "union_id": "",
      "regist_no": "32050XXXXX8749",
      "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XXXXXXXX",
      "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"
    }
  }
}
```

请求失败返回值示例

```
{
  "statusCode": 400009,
  "message": "Action值设置错误",
  "details": "Action值设置错误",
  "error": "AI_OP_400009"
}
```

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
AI_OP_400003	body内容为空	Body请求数据为空，没有包含内容
AI_OP_400004	body请求体非json格式	body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	必须的参数（'Action'、'ImageData'）未传
AI_OP_400007	传递非法参数	body字典内有除（'Action'、'ImageData'）外的参数
AI_OP_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
AI_OP_400009	Action值设置错误	Action值设置错误

错误码	错误信息	错误描述
AI_OP_400010	'ImageData'字段值为空字符	'ImageData'字段的base64字符串转换字节码异常
AI_OP_400011	'ImageData'字段base64数据处理异常	'ImageData'字段base64数据处理异常
AI_OP_400012	请求文件格式不合法	仅支持 jpeg/png/jpg/bmp 格式
AI_OP_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
AI_OP_400014	图片解码错误	字节码解码为图片错误
AI_OP_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
AI_OP_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

6.3.5 驾驶证识别

1、接口描述

针对驾驶证主页图片，进行OCR检测，返回检测到的驾驶证号、姓名、性别、国籍、住址、出生日期、初次领证日期、准驾车型、有效期限。驾驶证识别只支持主页识别。

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，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 格式。

图片大小限制：base64 编码后，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URI

/v1/aiop/api/3131mkjqbqck/driver-lisence/ocr/detect

5、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。 详见文档： Python3调用示例	33dfa732-b27b-464f-b15a-21ed6845afd5	
eop-date	是	String	请求时间，由用户构造，形如 yyyyymmddTHHMMSSZ。 详见文档： Python3调用示例	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。 签名逻辑详见文档： 认证鉴权 和 Python3调用示例		

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
Action	是	String	算法名称，固定值"DriversLicense"	DriversLicense	
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:image/jpeg;base64,"	_9j_4AAQSkZJRgA...	

6、请求代码示例

```
Curl -X POST
"https://ai-global.ctapi.ctyun.cn/v1/aiop/api/3131mkjqbqck/driver-lisence/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": "DriversLicense","ImageData": "_9j_4AAQSkZJRgA..."}'
```

7、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	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	String	如果statusCode为0，返回success；如果statusCode非0，则返回对应的可读错误信息	success	
returnObj	是	Object	返回的驾驶证识别结果对象		returnObj

表returnObj

参数	是否必填	参数类型	说明	示例	下级对象
No	是	String	身份证号	3101151992XX24XXXX	
name	是	String	姓名	刘XX	
sex	是	String	性别	男	
nation	是	String	国籍	中国	
address	是	String	住址	上海市长宁区XXX路900号	
date_birth	是	String	出生日期	1992-12-24	
date_first	是	String	初次领证日期	2020-10-25	
class	是	String	准驾车型	C1	
period	是	String	有效期限	2020-10-25至2026-10-25	

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	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

1.请求成功返回值示例

```
{
  "statusCode": 0,
  "message": "success",
  "returnObj": {
    "No": "3101151992XX24XXXX",
    "name": "刘XX",
    "sex": "男",
    "nation": "中国",
    "address": "上海市长宁区XXX路900号",
    "date_birth": "1992-12-24",
    "date_first": "2020-10-25",
    "class": "C1",
    "period": "2020-10-25至2026-10-25"
  }
}
```

2.请求失败返回值示例

```
{
  "statusCode": 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
400001	请求路径错误	请求路径错误
400002	请求方法错误	请求方法错误，请使用POST请求
400003	请求体内容为空	请求体请求数据为空，没有包含内容

400004	请求体非json格式	请求体内容需要符合json请求
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
400008	请求体的参数字段类型错误	Action、ImageData字段应该是String类型
400009	请求体的参数字段值为空	Action、ImageData字段值为空字符
400010	请求体的参数字段值设置错误	Action值设置错误
400011	base64数据处理异常	ImageData字段的base64字符串转换字节码异常
400012	文件格式不合法	仅支持 jpeg/png/jpg/bmp 格式
400013	文件大小不符合要求	该文件大小不符合要求,图片要求小于7M
410001	图片解码错误	字节码解码为图片错误
410002	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().toString()` 函数进行编码。

6.3.6 行驶证识别

1、接口描述

针对行驶证主页图片，进行OCR检测，返回检测到的号牌号码、车辆类型、所有人、住址、使用性质、品牌型号、车辆识别代号、发动机号码、注册日期、发证日期。行驶证识别只支持主页识别。

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：base64 编码后图片大小不超过 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URI

`/v1/aiop/api/3131km7vva4g/driving-lisence/ocr/detect`

5、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	

参数	是否必填	参数类型	说明	示例	下级对象
appkey	是	String	平台应用 appkey	562b8949 3b1a40e1b97 ea05e50XXXX X	
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。 详见文档: Python3调用示例	33dfa732- b27b-464f- b15a- 21ed6845afd5	
eop-date	是	String	请求时间, 由用户构造, 形如 yyyymmddTHHMMSSZ。 详见文档: Python3调用示例	20211221 T163014Z	
host	是	String	终端节点域名 , 固定字段	ai- global.ctapi.ctyun.cn	
Eop-Authoriz	是	String	由天翼云官网 accessKey 和		

参数	是否必填	参数类型	说明	示例	下级对象
ation			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。 签名逻辑详见文档： 认证鉴权和Python3调用示例		

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
Action	是	String	固定值 "VehicleRegistration"	VehicleRegistration	
ImageData	是	String	图片 Base64 编码数据，图片需要使用常规 safe	_9j_4AAQSkZJ	

参数	是否必填	参数类型	说明	示例	下级对象
			base64 编码方式，不包含前缀 "data:img/jpg;base64,"	RgA...	

6、请求代码示例

```

Curl -X POST
"https://ai-global.ctapi.ctyun.cn/v1/aiop/api/3131km7vva4g/driving-lisence/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:20241109T104641Z"
-H "host:ai-global.ctapi.ctyun.cn"
--data '{"Action": "VehicleRegistration", "ImageData": "_9j_4AAQSkZJRgA..."}'

```

7、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	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	String	如果statusCode为0，返回success；如果statusCode非0，则返回对应的可读错误信息	success	
returnObj	是	Object	返回的行驶证识别结果对象		returnObj

表returnObj

参数	是否必填	参数类型	说明	示例	下级对象
No_plate	是	String	号牌号码	沪AAXXX64	
type	是	String	车辆类型	小型轿车	
owner	是	String	所有人	XXX	
address	是	String	住址	上海市长宁区XXX路900号	
use	是	String	使用性质	非营运	
model	是	String	品牌型号	XXX牌 XXXXXXEV1	
vin	是	String	车辆识别代号	LC0CEXXXXXX X3856	
No_engine	是	String	发动机号码	1A1XXXX06	
date_register	是	String	注册日期	2024-10-25	
date_issue	是	String	发证日期	2024-10-25	

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
status 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

1.请求成功返回值示例

```
{
  "statusCode": 0,
  "message": "success",
  "returnObj": {
    "No_plate": "沪AAXX64",
    "type": "小型轿车",
    "owner": "XX",
    "address": "上海市长宁区XXX路900号",
    "use": "非营运",
    "model": "XXX牌XXXXXE1",
    "vin": "LC0CEXXXXXX3856",
    "No_engine": "1A1XXXX06",
    "date_register": "2021-10-25",
    "date_issue": "2021-10-25"
  }
}
```

2.请求失败返回值示例

```
{
  "statusCode": 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
-----	----

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
400001	请求路径错误	请求路径错误
400002	请求方法错误	请求方法错误，请使用POST请求
400003	请求内容为空	请求体请求数据为空，没有包含内容
400004	请求体非json格式	请求体内容需要符合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
400008	请求体的参数字段类型错误	Action、ImageData字段应该是String类型 NeedBackPage字段应该是bool类型
400009	请求体的参数字段值为空	Action、ImageData字段值为空字符
400010	请求体的参数字段值设置错误	Action值设置错误
400011	base64数据处理异常	ImageData字段的base64字符串转换字节码异常
400012	文件格式不合法	仅支持 jpeg/png/jpg/bmp 格式
400013	文件大小不符合要求	该文件大小不符合要求，图片要求小于7M

错误码	错误信息	错误描述
410001	图片解码错误	字节码解码为图片错误
410002	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。

6.4 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
2023-12-21	第三次正式发布。本次更新说明如下：修改API文档内容

7 常见问题

7.1 计费类

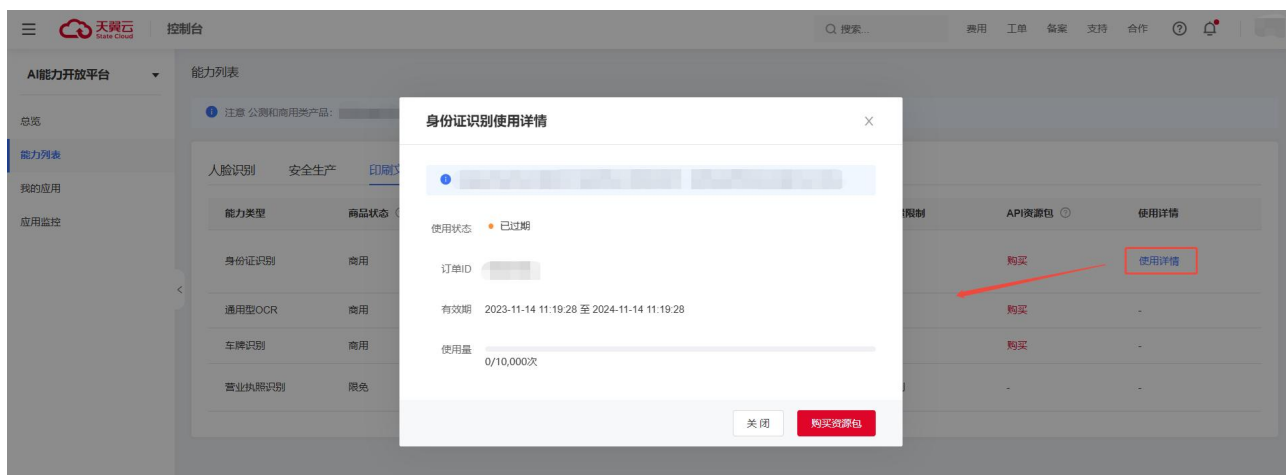
调用量在哪里查看？

调用明细您可在控制台内查看，进入控制台选择【控制台-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—我的能力—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。



调用量的扣费顺序是？

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。

服务调用时是否可保证一定的并发需求？

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，您可以尝试以下三种解决方法：

- 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。

印刷文字识别支持哪些计费方式？

我们提供封顶资源包的计费方式，资源包有效期一年，根据调用次数不同，费用也有所不同。

印刷文字识别API付费资源包价格表

● 印刷文字识别API付费资源包价格表

API名称	1万次	10万次	100万次	500万次	1000万次
通用OCR识别	—	450元	3200元	11500元	22000元
身份证识别	—	1350元	9600元	39000元	78000元
车牌识别	—	1350元	9600元	39000元	78000元
驾驶证识别	300元	2800元	15000元	45000元	80000元
行驶证识别	300元	2800元	15000元	45000元	80000元

注意：表格价格仅供参考，具体购买价格以价格发文为准。

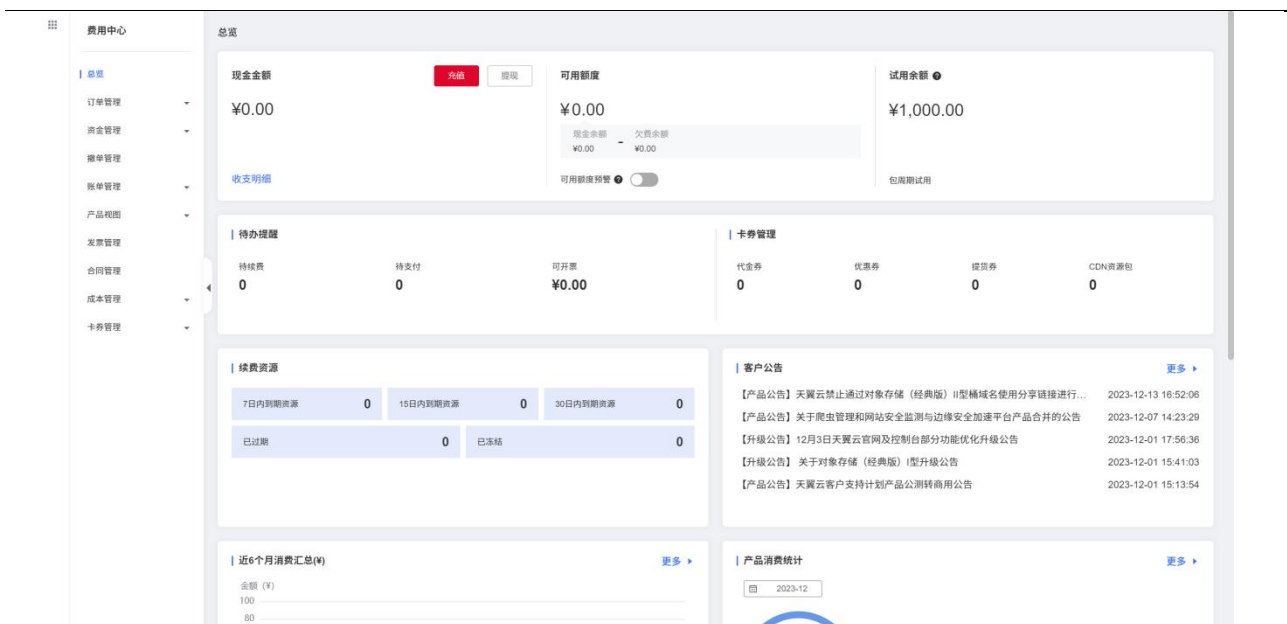
什么情况下会扣资源包次数，识别报错会扣费吗？

成功调用算入计费次数，单张图片算作一次调用。

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。

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。

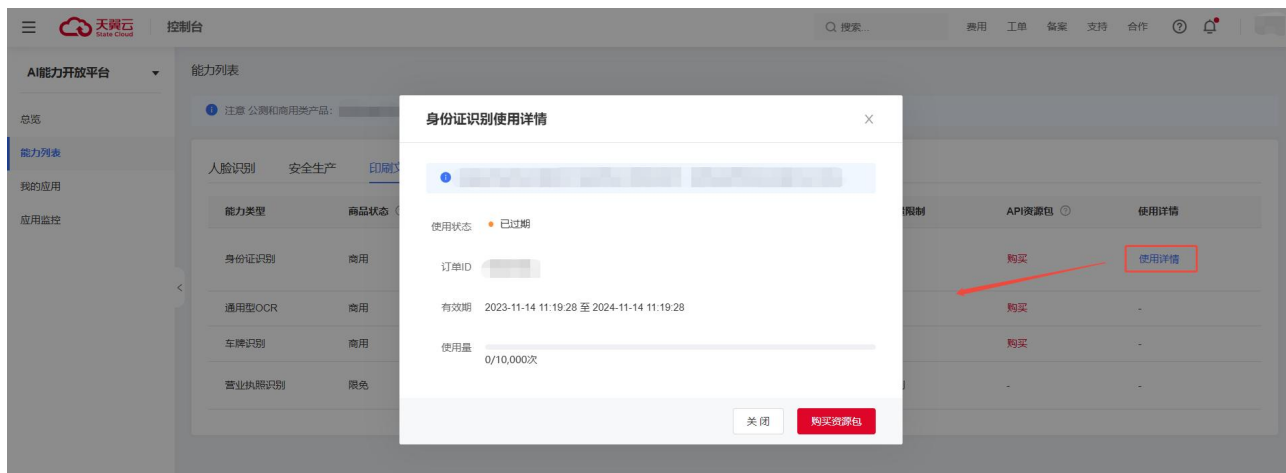


OCR服务剩余次数如何查询？

如您已购买印刷文字识别OCR资源包，资源包剩余次数可至控制台内查看，进入控制台-能力列表选择对应能力的条目，点击【使用详情】按钮查询。

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”。

”。



OCR资源包可以转移到别人的账号么？

目前不支持转移，建议购买前评估真实使用量，资源包用完可以叠加购买。

OCR服务开通后能否关闭？不使用是否会产生费用？

OCR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OCR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

资源包买错了可以退款吗？

若因产品介绍说明有误导导致的购买错误且资源包尚未使用，您可通过通过天翼云官网工单或者客服电话【400-810-9889】沟通申请手动退款，可能需要数个工作日审批，请耐心等待，款项会原路退回。在审批期间，您不能使用该资源包，一旦使用资源包就会导致退款失败。

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，谢谢配合。

印刷文字识别服务是否支持代金券付款？

支持代金券付款，同时还支持优惠券、提货券付款。具体使用方法：在我的>费用中心>卡券管理，选择相应的代金券进行使用。



印刷文字识别服务是否支持试用？

印刷文字识别服务商用产品暂不支持试用。限免产品直接开通即可使用，无需付费。

7.3 操作类

印刷文字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。

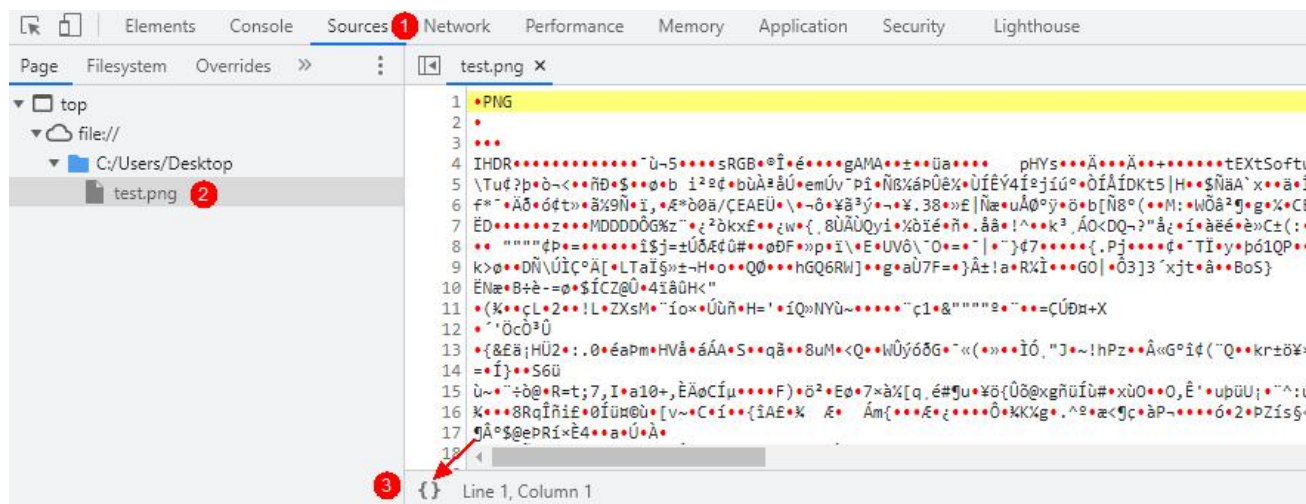
如何获取图片base64编码？

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。

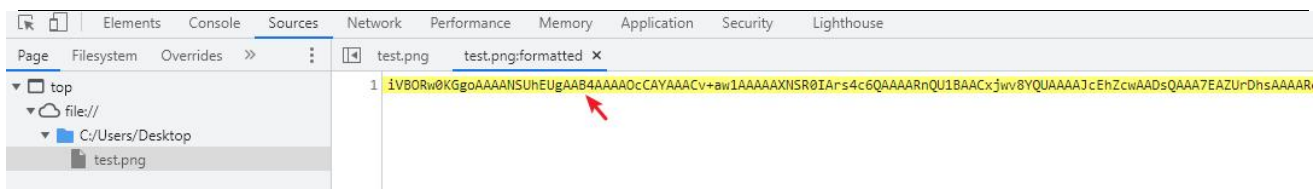
2.按下“F12”，在弹出窗口中选择“Sources”，在左侧导航树中选择需要编码的图片所在

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。



7.4 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？

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。

印刷文字识别是 HTTP GET 请求还是 HTTP POST 请求？

印刷文字识别以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。

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？

印刷文字识别的API接口对单次请求的图片限制不一致，具体可参考【API参考】-【API】-具体产品（如通用型OCR）“接口要求”中的详细说明。

车牌识别支持新能源车牌吗？

支持新能源车牌。车牌识别适用于中国大陆的车牌识别服务，支持蓝牌、黄牌（单层）、新能源车牌的检测与识别，同时支持图片内有多张车牌，结构化返回车牌内容及车牌位置坐标。

OCR能否提供100%识别准确率？

OCR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OCR产品服务有识别准确率相关问题，您可拨打通过天翼云官网工单或者客服电话【400-810-9889转1】联系我们。

服务调用时是否可保证一定的并发需求？

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，您可以尝试以下三种解决方法：

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。

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。

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。

QPS不够用怎么办？

产品默认支持10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10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。

OCR接口支持哪些格式？

OCR识别接口均支持图片格式数据，包括：png/.jpg/.jpeg/.bmp。

OCR服务是否支持私有化部署？

印刷文字识别OCR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。

注意

如用户操作手册说明与官网帮助中心说明不一致，以官网帮助中心说明为准。