

智能扫码 快速入门



腾讯云

【 版权声明 】

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分內容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

快速入门

- 一分钟接入服务端 API
- 可视化调用文字识别服务
- 控制台操作指引
- 图片压缩示例
- 控制台访问管理
- 敏感数据加密指引

快速入门

一分钟接入服务端 API

最近更新时间：2025-03-27 10:57:02

操作场景

本文将为您介绍如何使用 API 3.0 Explorer 在线调试文字识别相关接口，并快速将该接口对应的腾讯云开发者工具套件（SDK）集成到本地项目中。

操作步骤

只需三步，轻松完成腾讯云文字识别接口调用。

第一步：开通文字识别服务

在调用文字识别相关接口前，您需要进入 [文字识别控制台](#)，阅读《文字识别服务条款》后勾选同意并单击**立即开通**，即可一键开通**通用文字识别**、**卡证文字识别**、**票据文字识别**等服务。



服务开通成功后，您将获得各项服务对应的免费调用额度，可在 [资源包管理页](#) 查看。同时您也可以可以在 [文字识别购买页](#) 中购买对应文字识别服务的资源包，若免费额度以及资源包调用次数耗尽，接口计费将自动转为后付费方式按月进行结算，具体计费标准可以查看文字识别的 [购买指南](#)。

文字识别

[返回产品详情](#)

[产品文档](#) [计费说明](#) [产品控制台](#)

购买须知

使用说明 文字识别服务调用量的扣费顺序为“免费资源包->付费资源包->后付费”。当您的免费资源包耗尽时，服务将面临不可用风险，为保证业务不受影响，请及时在此页面购买预付资源包或前往 [控制台设置页](#) 开通后付费模式。
退订规则 资源包购买后未使用，支持7天无理由退款。若购买后已使用，不支持退款及剩余次数冻结。

选择配置

使用须知 资源包当日零点生效 资源包有效期一年 资源包规格可叠加 资源包有效时长不可叠加 资源包不可抵扣已产生的调用量

计费方式

预付费资源包 QPS叠加包

服务类别

通用文字识别 卡证文字识别 票据单据识别 特定场景识别 智能结构化识别 文本图像增强 智能扫码 API 2022 商户场景照识别

计费名称

通用印刷体识别 通用印刷体识别（高精度版） 表格识别 通用证照识别

适用接口

接口名称	接口描述
通用印刷体识别	支持多场景、任意版面下整图文字的识别，包括中英文、字母、数字和日语、韩语、西班牙语等十余种多语言识别。

套餐包内容

1千次 有效期自购买之日起一年内	1万次 有效期自购买之日起一年内	10万次 有效期自购买之日起一年内
100万次 有效期自购买之日起一年内	1000万次 有效期自购买之日起一年内	

协议条款

我已阅读并同意 [《文字识别服务条款》](#)、[《文字识别服务等级协议》](#)、[《计费概述》](#) 和 [《退费说明》](#)

数量 1

配置费用

[立即购买](#)

联系销售



第二步：调试文字识别接口

文字识别服务开通成功后，进入文字识别 [API 3.0 Explorer](#) 在线接口调试页面，选择需要调用的接口，并填写输入参数。

API Explorer
文字识别 (OCR)

身份证

卡证文字识别相关接口

- 身份证识别 (安全加密版)
- 有效身份证件识别 (鉴别版)
- 身份证识别
- 机动车登记证书识别
- 智能卡证分类
- 外国人永久居留身份证识别
- 港澳台居住证识别
- 中国香港身份证识别

文字识别API2022相关接口

IDCardOCR

ocr 2018-11-19 [查看API文档](#)

!

- 在线调用模块中当您发起请求时，平台通过已登录用户信息获取当前账号临时Access Keys，对当前账号发起操作。
- 发起请求为敏感操作，在您进行敏感操作前，需要先完成身份验证以确保是您本人操作；该操作等同于真实操作，建议您仔细阅读相关产品文档了解费用等详情，谨慎操作！

更多选项 ▾

输入参数

Region ^①

本接口不需要传递该参数

参数输入方式

表单 JSON 参数推荐

ImageBase64 (选项) ^[*] ^①

string

ImageUrl (选项) ^[*] ^①

string

CardSide (选项) ^[*] ^①

string

Config (选项) ^[*] ^①

string

EnableRecognitionRectify (选项) ^[*] ^①

展示英文接口
发起调用
调用历史
展示所有参数 ▾

在 API 3.0 Explorer 界面的【文档说明】选项卡中可以查看对应接口输入参数的具体含义。

API Explorer

文字识别 (OCR)

IDCardOCR

ocr 2018-11-19 [查看API文档](#)

!

- 在线调用模块中当您发起请求时，平台通过已登录用户信息获取当前账号临时Access Keys，对当前账号发起操作。
- 发起请求为敏感操作，在您进行敏感操作前，需要先完成身份验证以确保是您本人操作；该操作等同于真实操作，建议您仔细阅读相关产品文档了解费用等详情，谨慎操作！

更多选项 ▾

输入参数

Region ^①

本接口不需要传递该参数

参数输入方式

表单 JSON 参数推荐

ImageBase64 (选项) ^[*] ^①

string

ImageUrl (选项) ^[*] ^①

string

CardSide (选项) ^[*] ^①

string

Config (选项) ^[*] ^①

string

EnableRecognitionRectify (选项) ^[*] ^①

boolean

文档说明

2. 输入参数

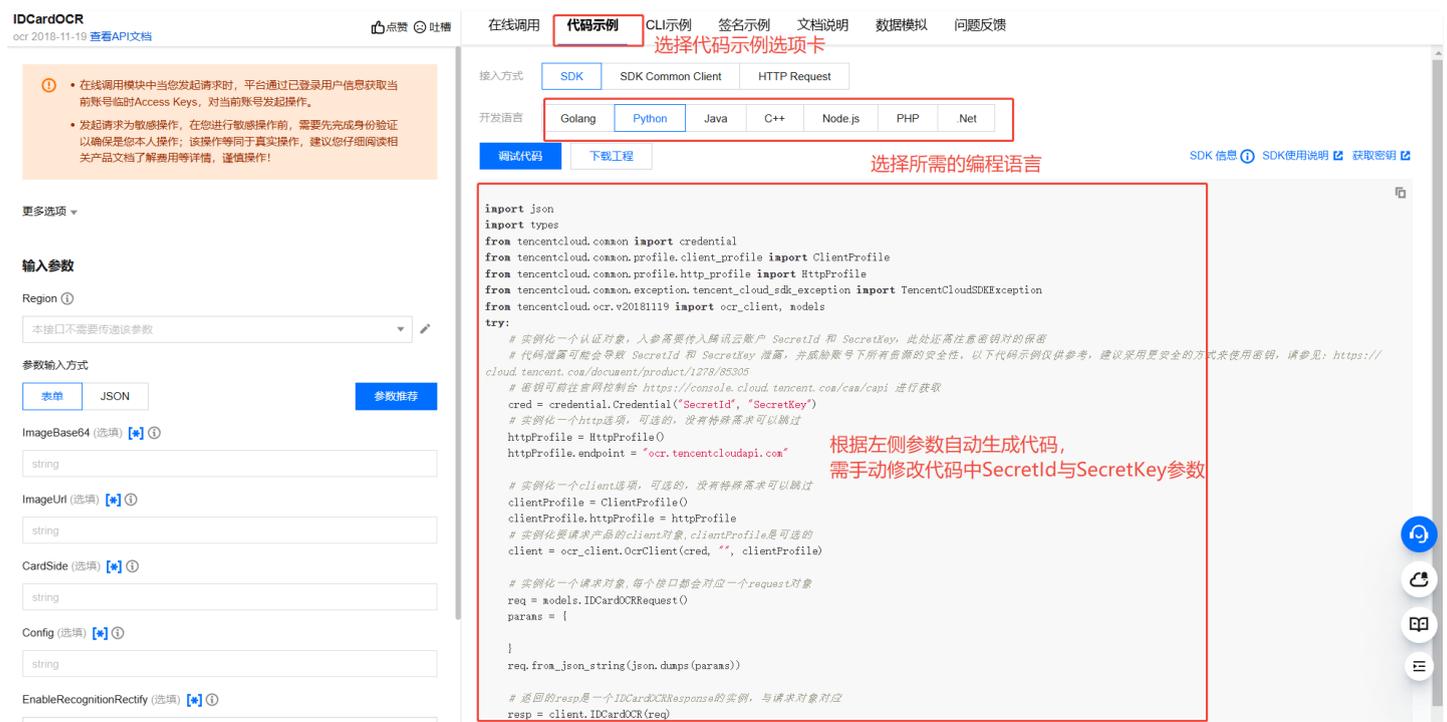
以下请求参数列表仅列出了接口请求参数和部分公共参数。完整公共参数列表见公共请求参数。

参数名称	必填	类型	描述
Action	是	String	公共参数。本接口取值：IDCardOCR。
Version	是	String	公共参数。本接口取值：2018-11-19。
Region	否	String	公共参数。此参数为可选参数。
ImageBase64	否	String	图片的 Base64 值。要求图片经Base64编码后不超过 7M，分辨率建议500*800以上，支持PNG、JPG、JPEG、BMP格式。建议卡片部分占图片2/3以上。 图片的 ImageUrl、ImageBase64 必须提供一个，如果都提供，只使用 ImageUrl。 示例值：/9j4AAQSkZJRg...s97nI/Q==
ImageUrl	否	String	图片的 URL 地址。要求图片经Base64编码后不超过 7M，分辨率建议500*800以上，支持PNG、JPG、JPEG、BMP格式。建议卡片部分占图片2/3以上。 建议图片存储于腾讯云，可保障更高的下载速度和稳定性。 示例值：https://ocr-demo-1254418846.cos.ap-guangzhou.myqcloud.com/cardIDCardOCR/IDCardOCR1.jpg
CardSide	否	String	FRONT：身份证有照片的一面（人像面）， BACK：身份证有国徽的一面（国徽面）， 该参数如果不填，将为您自动判断身份证正反。 示例值：FRONT
EnableRecognitionRectify	否	boolean	以下可选字段均为bool 类型，默认false： CropIdCard，身份证照片裁剪（去除证件外多余的边缘、自动矫正拍摄角度） CropPortrait，人像照片裁剪（自动截取身份证头像区域） CopyWater，水印信息

其中，Region 表示提供文字识别服务的资源所在区，单击 Region 的下拉框可以选择不同地域的腾讯云服务器，建议选择与访问点 IP 地址距离相近的地域：例如您的访问点在深圳，建议 Region 选择距离较近的华南地区(广州)。



填写好输入参数之后，选择代码示例选项卡，可以看到自动生成的不同编程语言代码（可支持 Java、Python、Node.js、PHP、GO、.NET、C++ 语言），生成代码中的部分字段信息和填写内容是关联的，如需调整传入参数，可在左侧修改参数值后重新生成代码。



填写好输入参数之后，选择在线调用选项卡，单击发送请求按钮可进行真实请求，供您调试、参考。



第三步：集成文字识别 SDK

确认本地依赖环境满足以下条件：

编程环境	SDK 集成要求
------	----------

Node.js	需要7.10.1版本及以上
Python	需要2.7至3.6版本
Java	需要 JDK 7版本及以上
Go	需要 Go 1.9版本及以上
.Net	需要 .NET Framework 4.5+ 和 .NET Core 2.1
PHP	需要5.6.33版本及以上
C++	需要 GCC 4.8版本及以上的 C++编辑器和 cmake 3.0版本及以上的编译工具，暂时仅支持 Linux 环境，不支持 Windows 环境

安装与本地依赖环境对应的腾讯云文字识别 SDK，下面将以 Node.js 为例说明 SDK 安装、使用的方法，其他语言的 SDK 使用方法请参考 [腾讯云 SDK 使用手册](#)。

(1) 通过 npm 安装（推荐）：npm 是 Node.js 的包管理工具，执行以下安装命令：

```
npm install tencentcloud-sdk-nodejs --save
```

(2) 通过源码包安装：前往 [GitHub 代码托管地址](#)，下载源码压缩包。解压源码包到您项目合适的位置。

SDK 安装完成后，可在您的项目代码中引用 API 3.0 Explorer 自动生成的代码，以 Node.js 为例，简易 demo 示例如下：

```
const tencentcloud = require("tencentcloud-sdk-nodejs")

// 导入对应产品模块的client models。
const OcrClient = tencentcloud.ocr.v20181119.Client

// 实例化要请求产品的client对象
const client = new OcrClient({
  // 腾讯云认证信息
  credential: {
    secretId: "secretId",
    secretKey: "secretKey",
  },
  // 产品地域
  region: "ap-shanghai",
  // 可选配置实例
  profile: {
    signMethod: "TC3-HMAC-SHA256", // 签名方法
    httpProfile: {
      reqMethod: "POST", // 请求方法
      reqTimeout: 30, // 请求超时时间，默认60s
    },
  },
})

const params = {
  "ImageUrl": "https://ocr-demo-1254418846.cos.ap-guangzhou.myqcloud.com/card/IDCardOCR/IDCardOCR1.jpg"
};

// 通过client对象调用想要访问的接口（Action），需要传入请求对象（Params）以及响应回调函数
// 即：client.Action(Params).then(res => console.log(res), err => console.error(err))
client.IDCardOCR(params).then(
  (data) => {
    console.log(data)
  },
  (err) => {
```

```
console.error("error", err)
}
)
```

注意事项

- SDK 调用公共参数时只需要关注 Region 字段，建议域名和 Region 统一使用 “ap-guangzhou”。
- 图片/视频转 Base64 时，需要去掉相关前缀 data:image/jpg;base64, 和换行符 \n。如果SDK调用时出现如下情况报错:

```
[TencentCloudSDKException]message:AuthFailure.SignatureFailure-The provided credentials could not be validated because of exceeding request size limit, please use new signature method `TC3-HMAC-SHA256`. requestId:719970d4-5814-4dd9-9757-a3f11ecc9b20
```

则需要手动设置签名类型:

```
signMethod: "TC3-HMAC-SHA256", // 指定签名算法"TC3-HMAC-SHA256" (默认为 HmacSHA256)
```

如果接口请求内容超过 1M，只能使用 V3 鉴权(TC3-HMAC-SHA256)。

- 文字识别 API 3.0 SDK 目前支持的语言包括: Node.js、Python、Java、PHP、Go、.Net、C++。如果您使用其他编程语言或者不想使用腾讯云 SDK，需要自行实现 V3 鉴权进行接口调用，如何实现请参考 [Github 示例](#)。建议使用 API 3.0 Explorer 中签名串生成工具进行核验签名有效性。



可视化调用文字识别服务

最近更新时间：2025-03-27 10:57:02

本文将为您介绍如何可视化调用文字识别服务，包括两种方式：[API 3.0 Explorer](#) 以及 Postman，建议您使用 API 3.0 Explorer 方式。具体步骤如下。

[观看视频](#)

开通文字识别服务

在调用文字识别相关接口前，您需要进入 [文字识别控制台](#)，阅读《[文字识别服务条款](#)》后勾选同意并单击**立即开通**，即可一键开通通用、卡证、票据、行业文档等多场景下的印刷体、手写体识别，以及智能扫码、卡证票据核验服务。



服务开通成功后，您将获得各项服务对应的免费调用额度，可在 [资源包管理页](#) 查看。同时您也可以[在文字识别购买页](#)中购买对应文字识别服务的资源包，若免费额度以及资源包调用次数耗尽，接口计费将自动转为后付费方式按月进行结算，具体计费标准可以查看文字识别的 [计费概述](#)。

文字识别 [返回产品详情](#)

[产品文档](#) [计费说明](#) [产品控制台](#)

购买须知

使用说明 文字识别服务调用量的扣费顺序为“免费资源包->付费资源包->后付费”。当您的免费资源包耗尽时，服务将面临不可用风险，为保证业务不受影响，请及时在此页面购买预付资源包或前往 [控制台设置页](#) 开通后付费模式。

订购规则 资源包购买后未使用，支持7天无理由退款。若购买后已使用，不支持退款及剩余次数冻结。

选择配置

使用须知 资源包当日零点生效 资源包有效期一年 资源包规格可叠加 资源包有效时长不可叠加 资源包不可抵扣已产生的调用量

计费方式 **预付资源包** QPS叠加包

服务类别 **通用文字识别** 卡证文字识别 票据单据识别 特定场景识别 智能结构化识别 文本图像增强 智能扫码 API 2022 商户场景照识别

计费名称 **通用印刷体识别** 通用印刷体识别 (高精度版) 表格识别 通用证照识别

接口名称	接口描述
通用印刷体识别	支持多场景、任意版面下整图文字的识别，包括中英文、字母、数字和日语、韩语、西班牙语等十余种多语言识别。

套餐包内容	1千次	1万次	10万次	100万次	1000万次
有效期自购买之日起一年内	有效期自购买之日起一年内	有效期自购买之日起一年内	有效期自购买之日起一年内	有效期自购买之日起一年内	有效期自购买之日起一年内

数量

配置费用 8元 0.05元

立即购买

说明

文字识别服务开通成功后，您可根据喜好选择如下两种方式来调用文字识别服务，建议选择 API 3.0 Explorer。

使用 API 3.0 Explorer 调用文字识别接口

输入参数

进入文字识别 [API 3.0 Explorer](#) 在线接口调试页面，选择需要调用的接口，并填写输入参数。

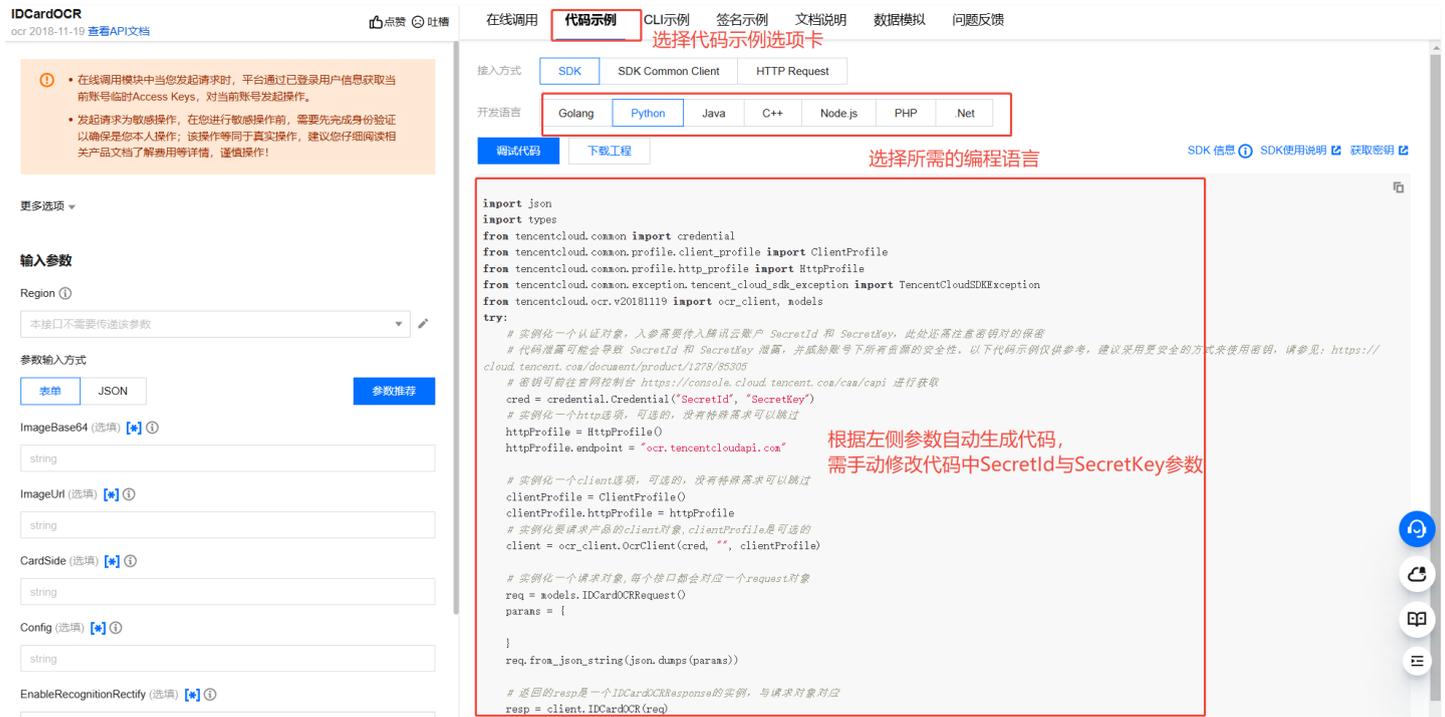
在 API 3.0 Explorer 界面的文档说明选项卡中可以查看对应接口输入参数的具体含义。

参数名称	是否	类型	描述
Action	是	String	公共参数，本接口取值：IDCardOCR。
Version	是	String	公共参数，本接口取值：2018-11-19。
Region	否	String	公共参数，本参数为可选参数。
ImageBase64	否	String	图片的 Base64 值。要求图片经Base64编码后不超过 7M，分辨率建议500*800以上，支持PNG、JPG、JPEG、BMP格式。建议卡片部分占图片2/3以上。 图片的 ImageURL、ImageBase64 必须提供一个，如果都提供，只使用 ImageURL。 示例值：/s/4AAQSKZfRg...s97n/2Q==
ImageUrl	否	String	图片的 URL 地址。要求图片经Base64编码后不超过 7M，分辨率建议500*800以上，支持PNG、JPG、JPEG、BMP格式。建议卡片部分占图片2/3以上。 建议图片存储于腾讯云，可保障更高的下载速度和稳定性。 示例值：https://ocr-demo-1254418846.cos.ap-guangzhou.myqcloud.com/cardIDCardOCR/IDCardOCR1.jpg
CardSide	否	String	FRONT：身份证有照片的一面（人像面）， BACK：身份证有国徽的一面（国徽面）。 该参数如果不填，将为您自动判断身份证正反面。 示例值：FRONT

其中，Region表示提供文字识别服务的资源所在区，单击 Region 的下拉框可以选择不同地域的腾讯云服务器，建议选择与访问点 IP 地址距离相近的地域：例如您的访问点在深圳，建议 Region 选择距离较近的华南地区(广州)。



填写好输入参数之后，选择代码生成选项卡，可以看到自动生成的不同编程语言代码（可支持 Java、Python、Node.js、PHP、GO、.NET、C++ 语言），生成代码中的部分字段信息和填写内容是关联的，如需调整传入参数，可在左侧修改参数值后重新生成代码。



发起请求

填写好输入参数之后，选择在线调用选项卡，单击发送请求按钮可进行真实请求，供您调试、参考。



使用 Postman 调用文字识别接口

Postman 下载安装

1. 下载地址：[Download Postman | Get Started for Free](#)，按照电脑类型选择安装包。
2. 下载完成后，双击打开安装包，按照提示安装，如无特殊需求选择默认即可。初次打开需要登录，若无账号，选择跳过。

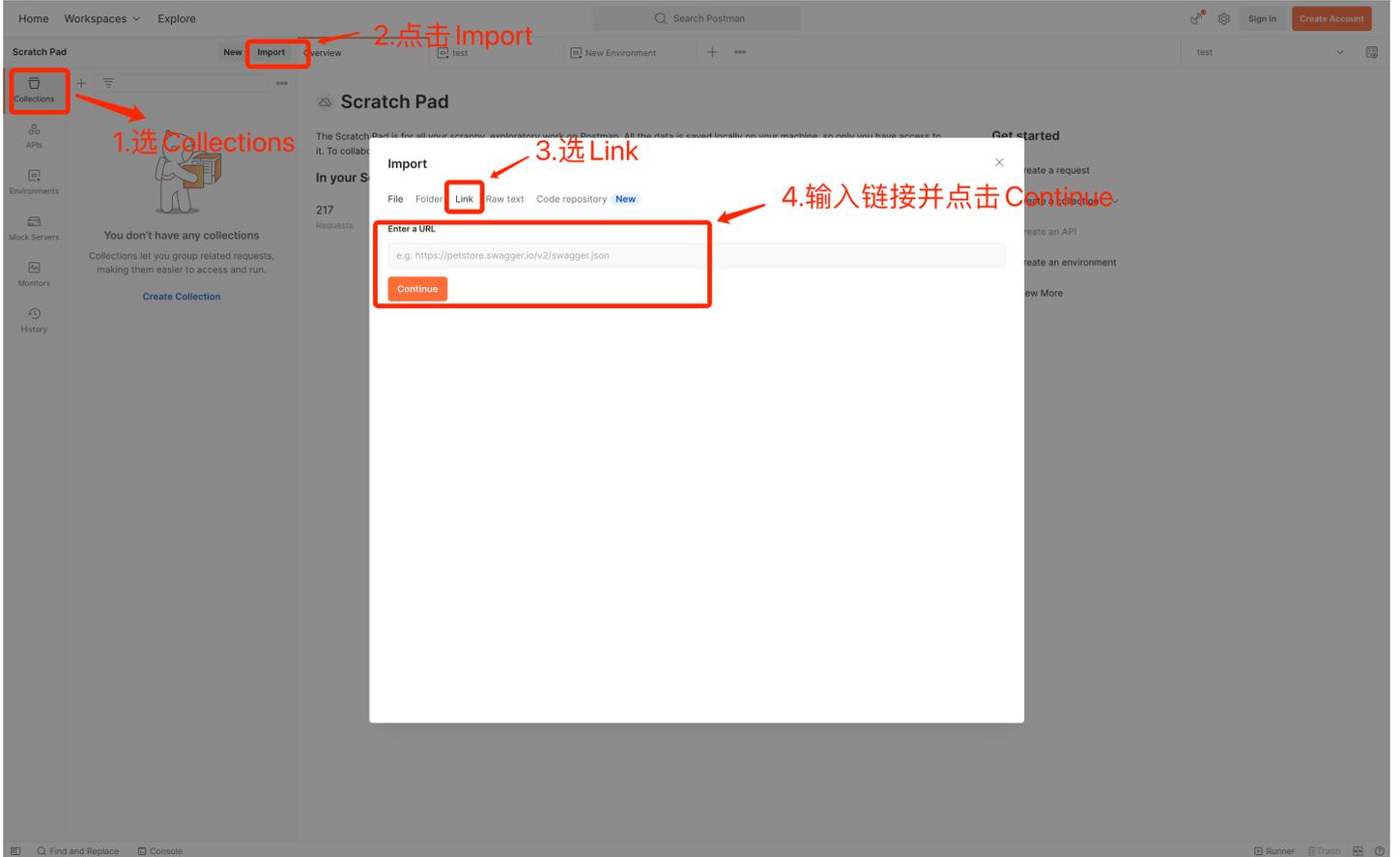
接口调用

导入 Postman 配置

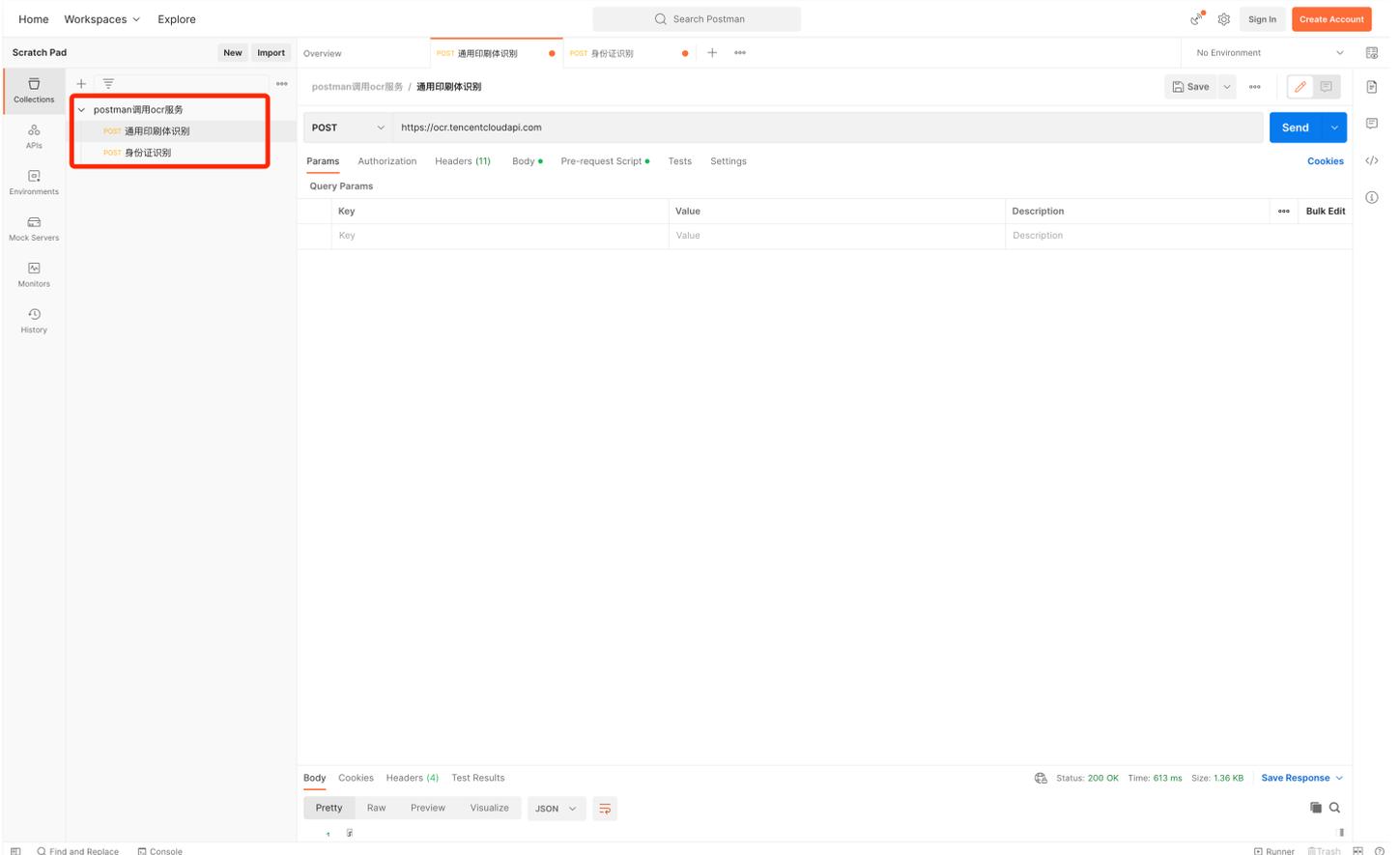
借助 Postman 的导入功能来初始化配置，配置链接：

https://ocr-documents-1258344699.cos.ap-guangzhou.myqcloud.com/postman_config.json

具体导入步骤如下：



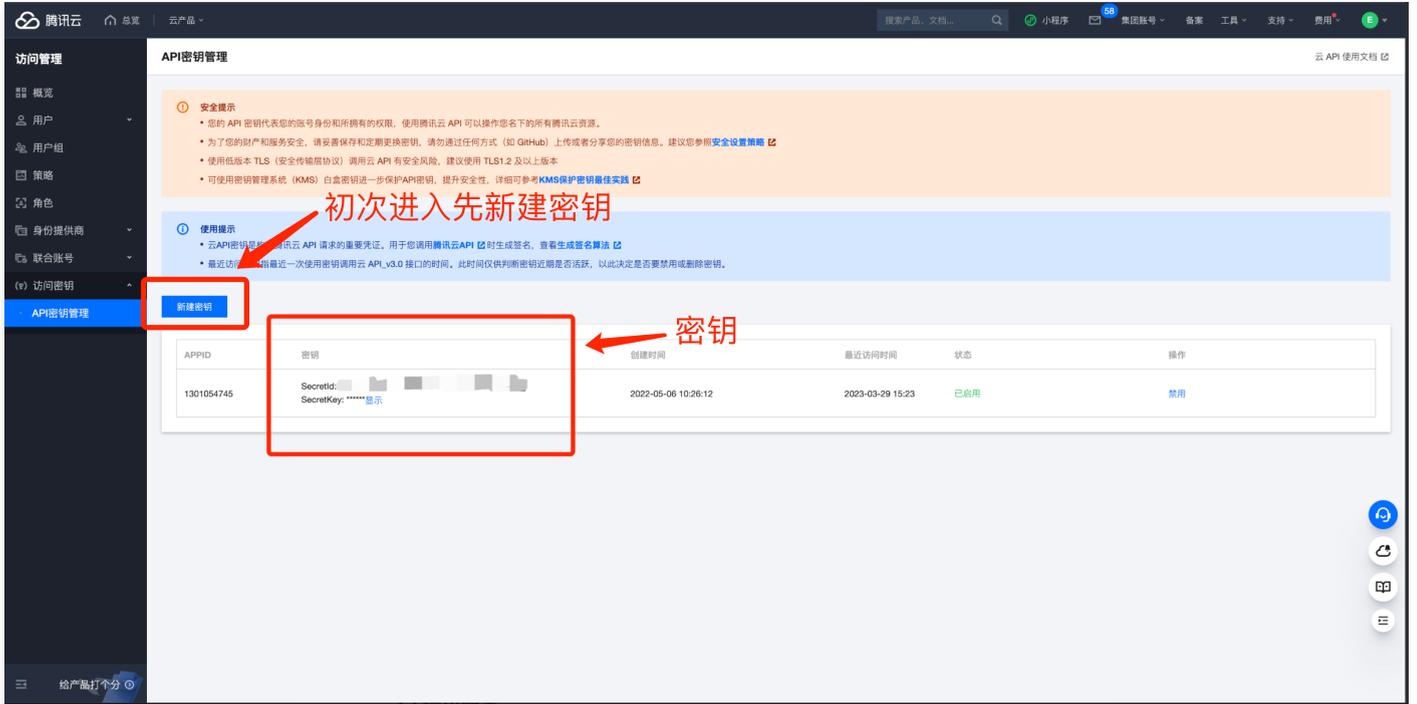
初始配置包含两个接口示例，其所需配置信息都已包含在内。看到如下内容表示导入成功。



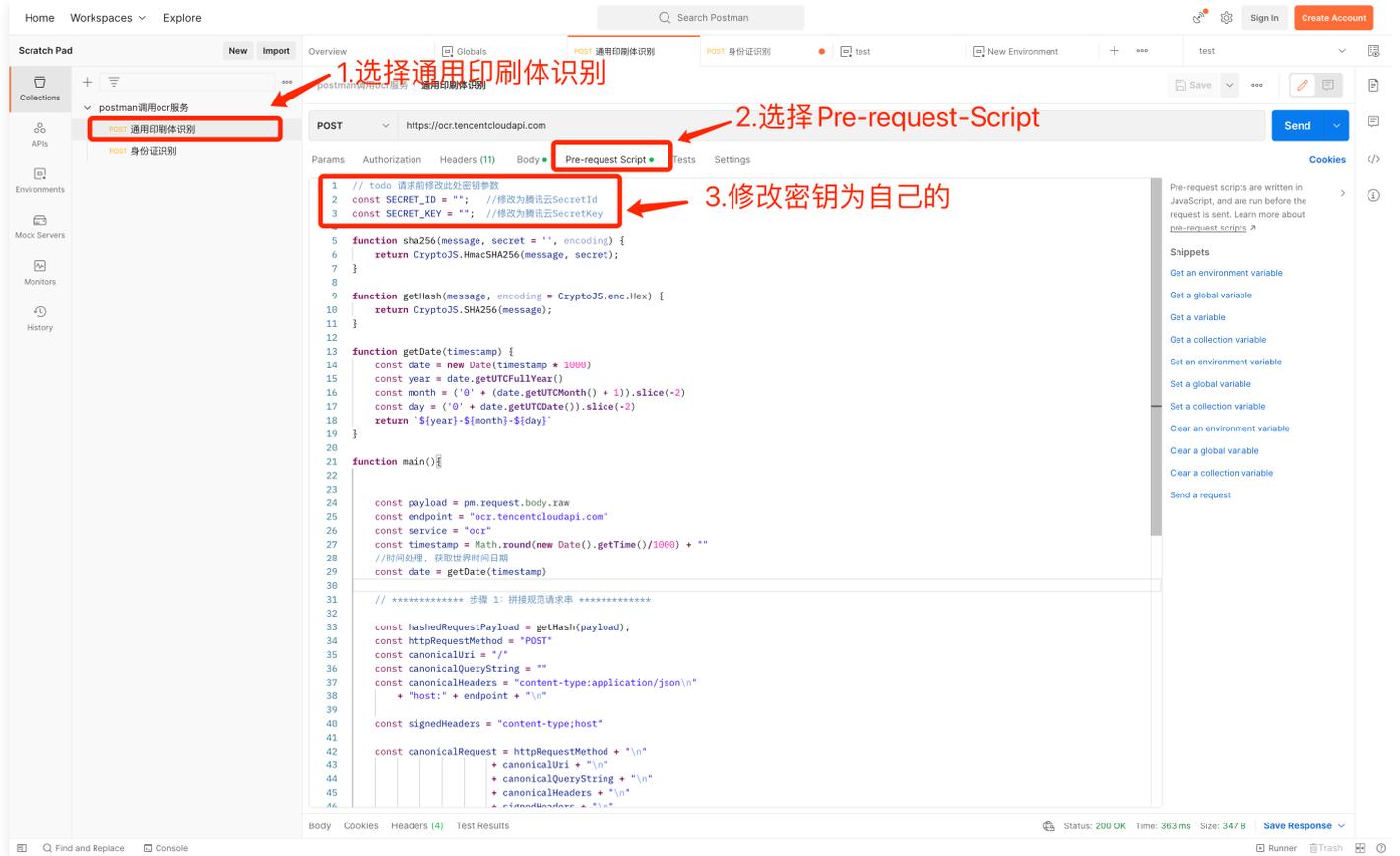
修改腾讯云密钥

发起请求前，需先将该请求对应 Pre-request-Script 中的密钥修改为自己的，这样脚本才能正确计算鉴权参数。

1. 登录访问管理控制台 [获取密钥](#)。

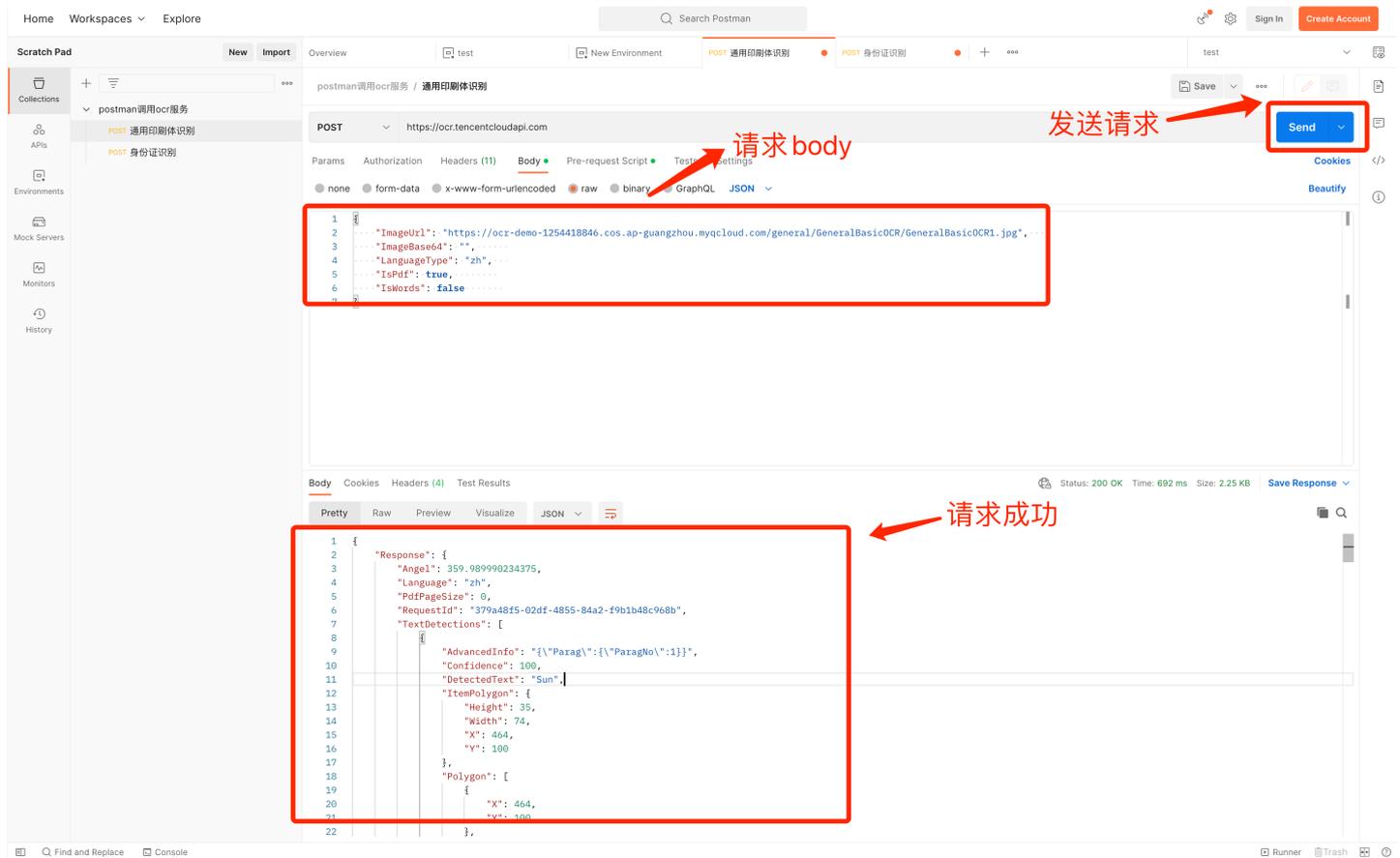


2. 例如我们要调用 [通用印刷体识别接口](#)，修改其预处理脚本中的密钥。



发起请求

密钥修改完后，单击 Send 发送请求，看到如下内容表示请求成功。



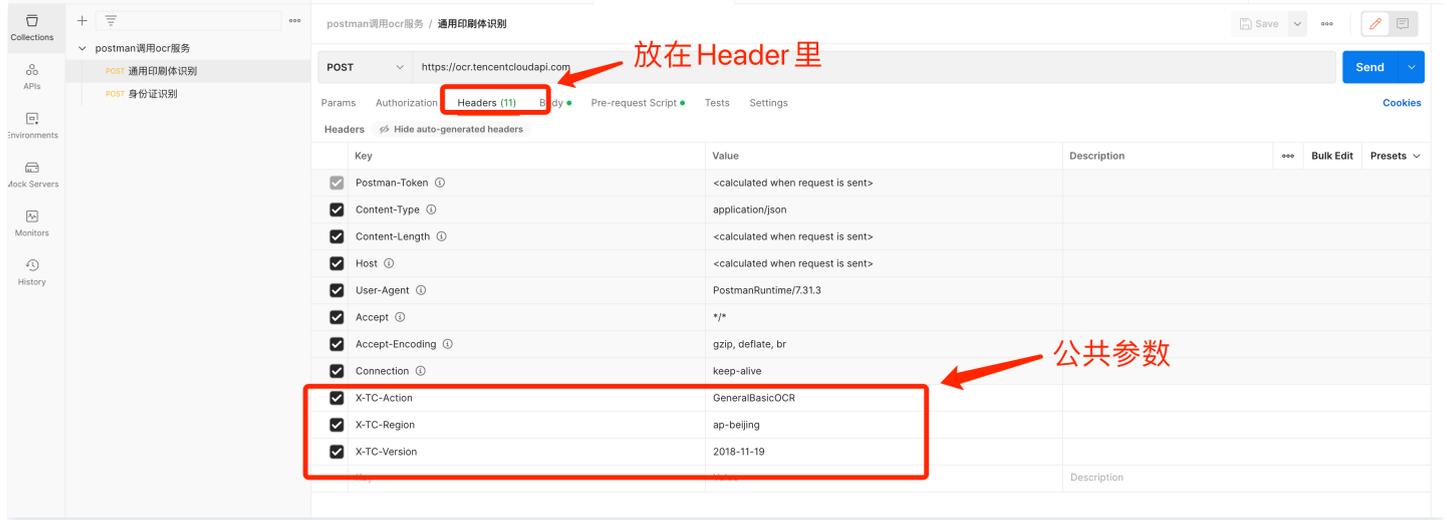
如何修改请求参数和请求接口

参数介绍

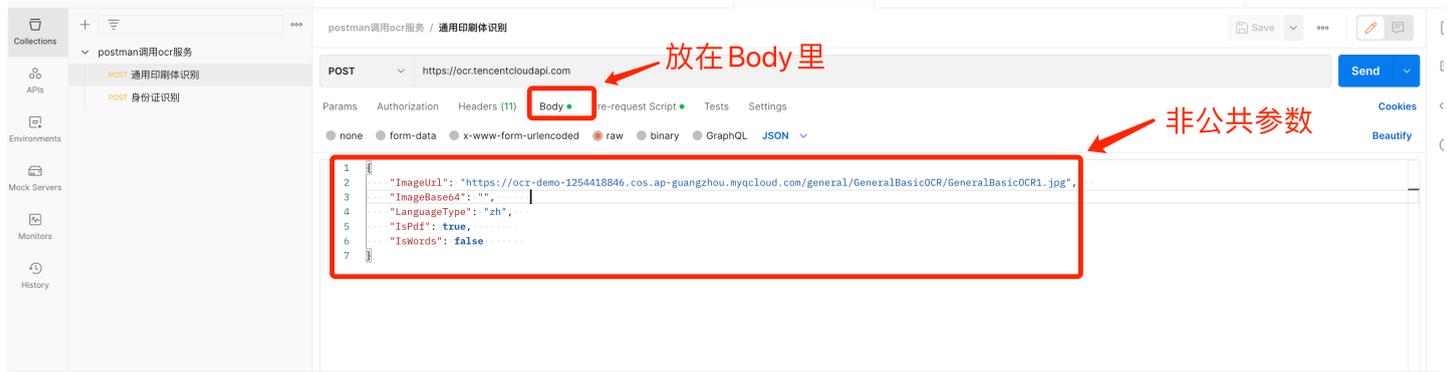
OCR 服务（云 API）参数分为公共参数和非公共参数，以 通用印刷体识别接口 为例：



对应在 Postman 当中：



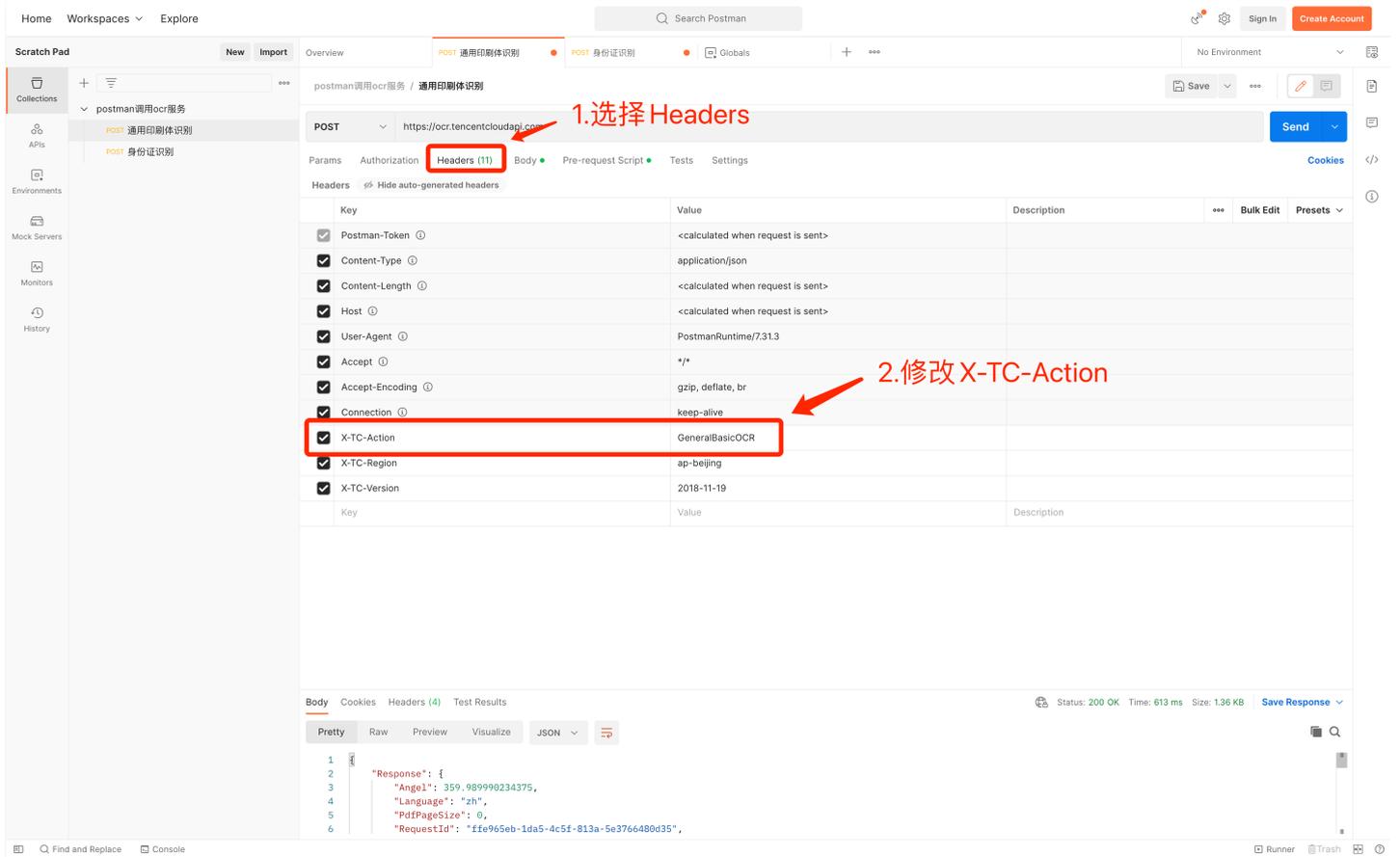
注意
公共参数传递时参数名需加上前缀“X-TC-”，详情见 [API 中心](#)。



请求参数变更

如果想解析不同的图片或者设置不同的请求参数，步骤如下：

如需请求 OCR 服务的其他接口，需修改 Header 中的 X-TC-Action。



X-TC-Action 参数对应接口文档的 Action 入参，如身份证识别接口的 Action 为 IDCardOCR。



修改 Action 之后，再按照接口文档修改请求 Body，发起请求即可。

控制台操作指引

最近更新时间：2025-03-27 10:57:02

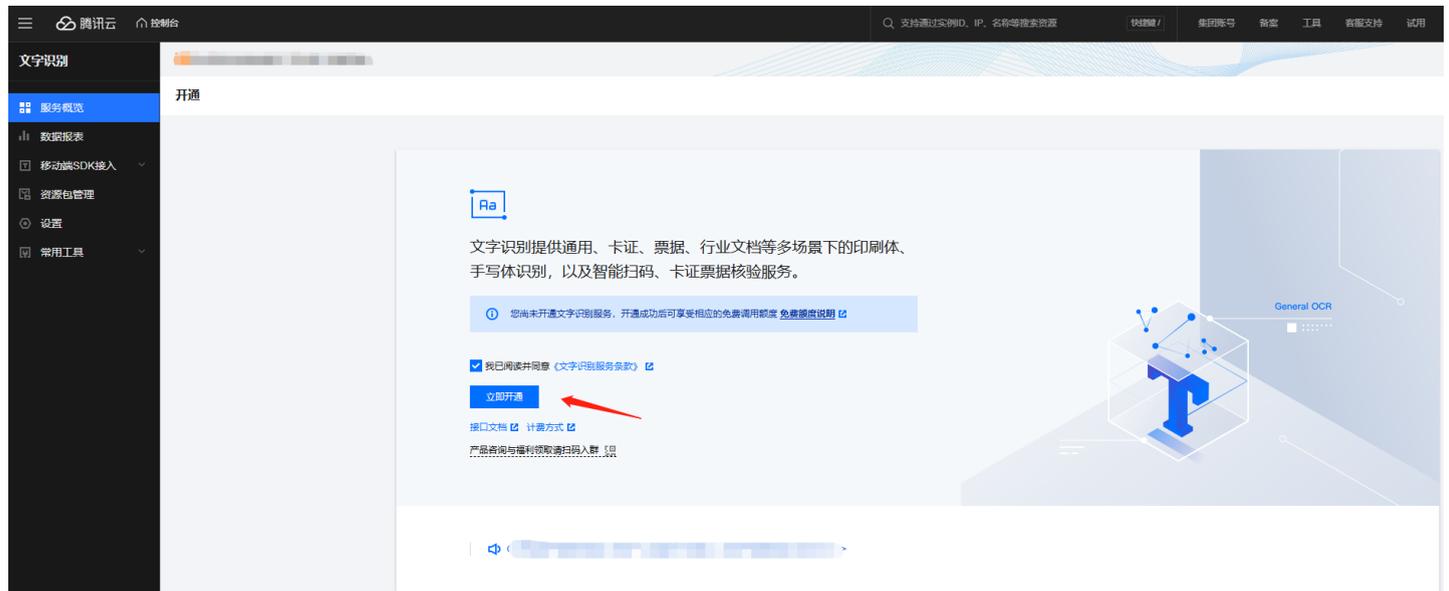
本文将为您介绍如何使用腾讯云 OCR 的新版控制台，帮助您更快速上手使用。

功能概述

1. 新版控制台功能包含服务概览、数据报表、移动端SDK 接入、资源包管理、设置、用量查询权限管理及常见工具的功能模块。
2. 服务概览模块包含接入指引、资源包使用概览，并提供产品动态、常用文档、功能体验及联系我们等相关指引。
3. 数据报表模块提供筛选和导出功能，可根据时间段、服务类别、接口名称、账号和渠道筛选查询各接口的使用情况，支持查看总调用量、计费量、成功数、成功率、失败数及错误码分布，支持将调用明细导出到本地。
4. 移动端 SDK 接入模块提供文字识别客户端SDK接入申请入口，支持 iOS SDK、Android SDK、Harmony SDK 下载；智能扫码 SDK 接入提供智能扫码 SDK 的申请入口，您可申请使用测试版，或在线购买正式版本。
5. 资源包管理模块提供资源包使用详情。
6. 设置模块支持用户自主设置是否开通后付费及关闭文字识别服务；支持是否开通后付费设置、关闭文字识别所有 API 接口和 OCR SDK 服务（智能扫码 SDK 除外）。

开通产品

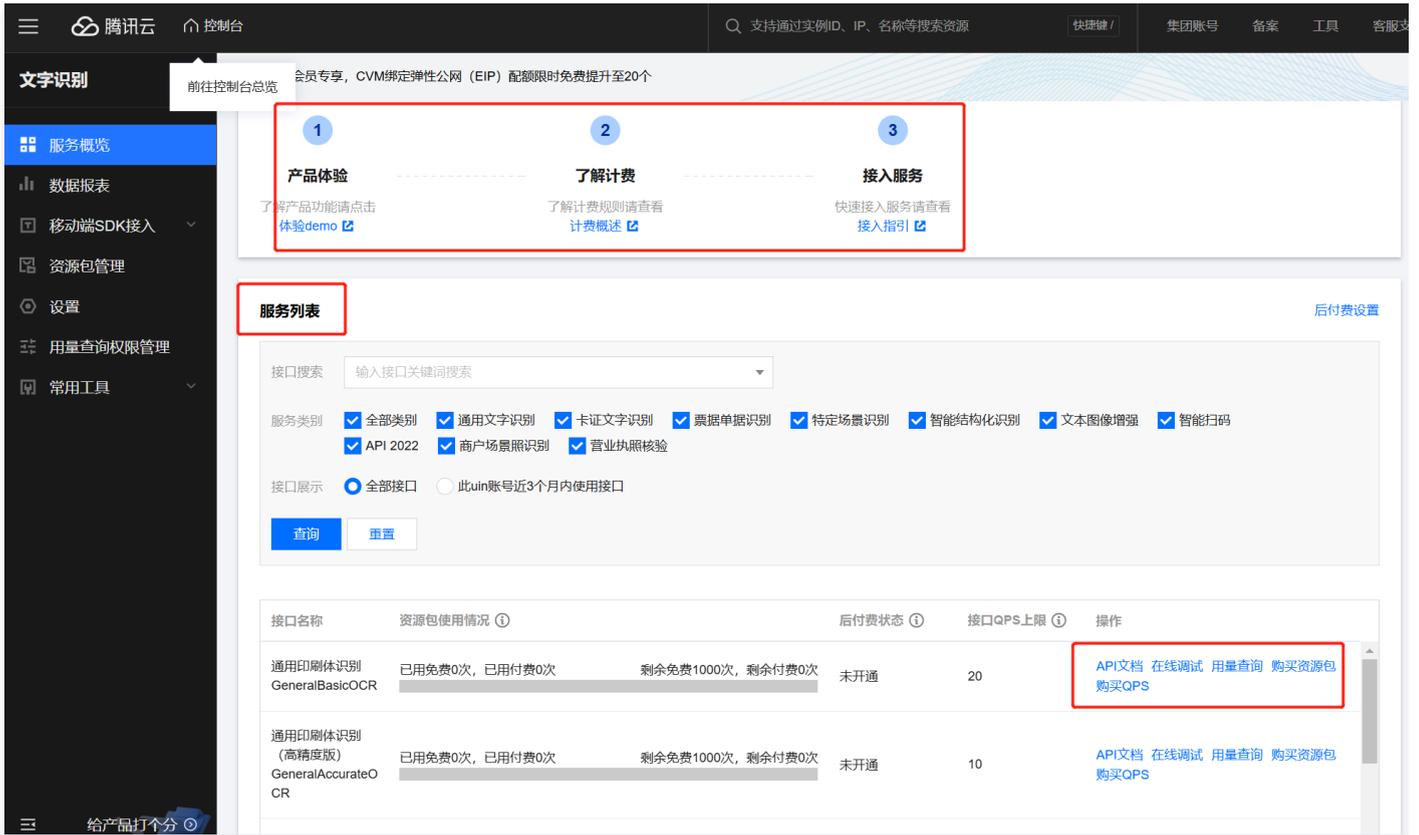
若您未开通文字识别服务，进入控制台后将进入服务开通页，阅读《文字识别服务条款》后单击勾选同意框，即可一键开通文字识别所有 API 接口服务，并将获得每个接口对应的免费资源包。



查看服务概览

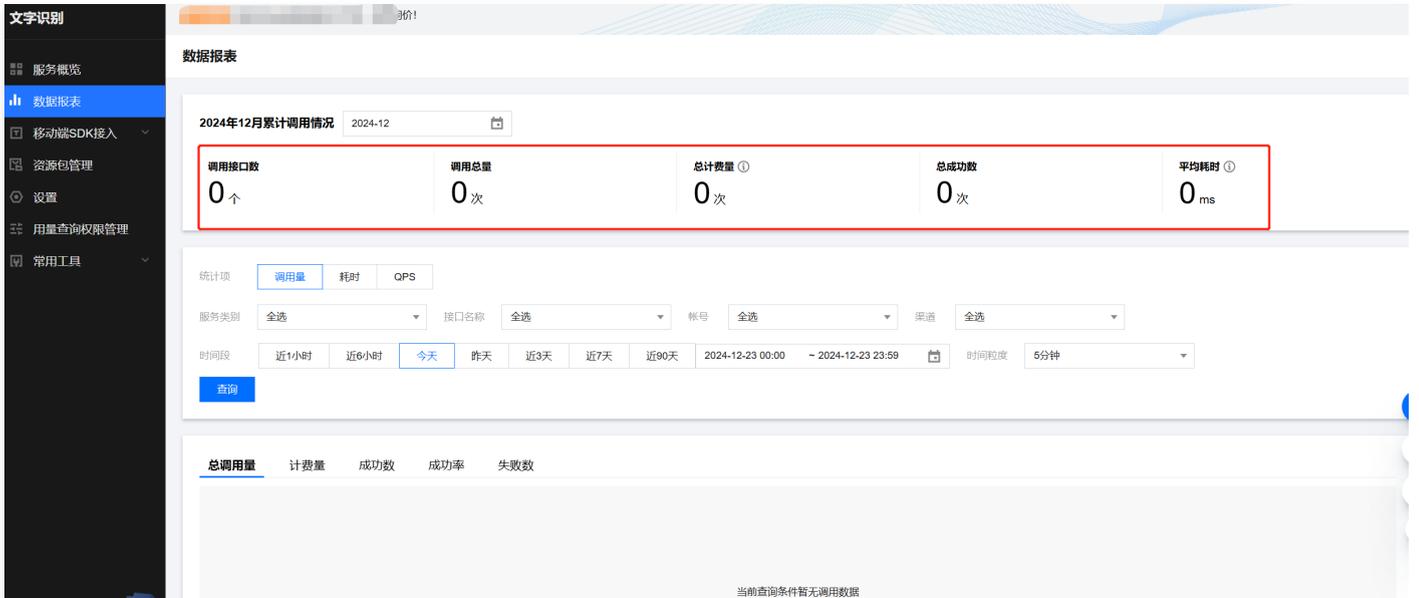
1. 新手指引

如果您是新用户，您可根据**服务概览**模块提供的步骤指引及文档了解产品、计费以及如何接入使用；可通过下图**服务列表**进行接口搜索、查询，也可以点击“API 文档、在线调试、用量查询、购买资源包、购买 QPS”等跳转对应链接进行下一步操作。



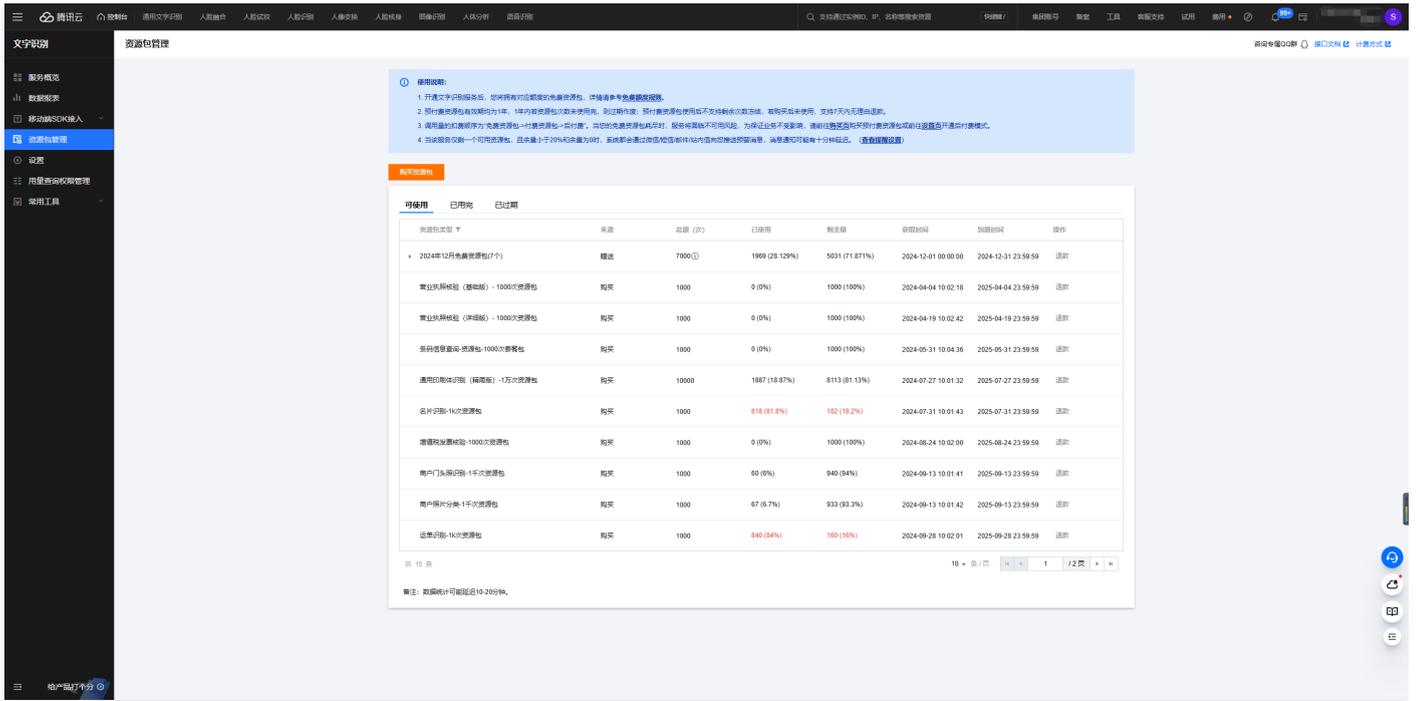
2. 接口调用量概览

若您已开始使用 OCR，您可在数据报表模块查看您近一个月的接口调用情况，包括调用接口数、调用总数、总成功数、总计费量等。



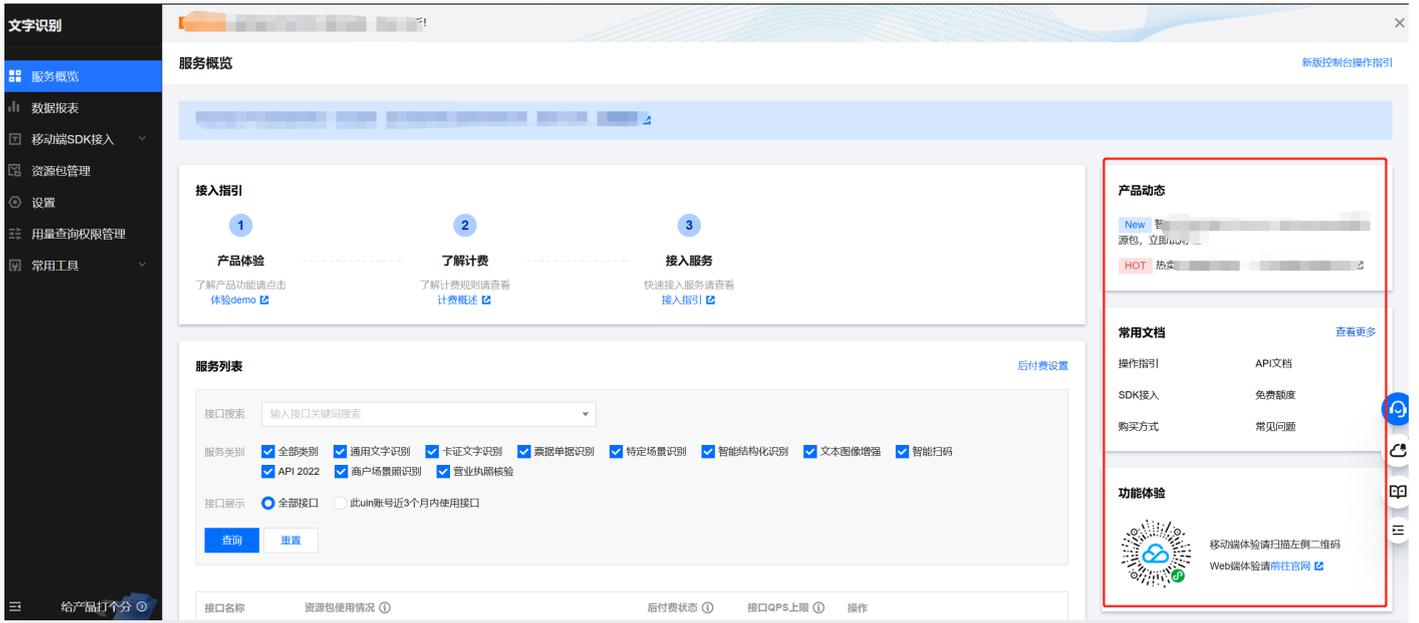
3. 资源包概览

资源包概览模块提供了资源包情况总览，您可以了解到资源包可使用的接口及剩余量/总量，当剩余量不多时，您可以通过立即购买跳转购买页购买资源包，当您需修改消息提醒的通知方式时，您可以单击使用说明里的查看提醒设置跳转后设置。



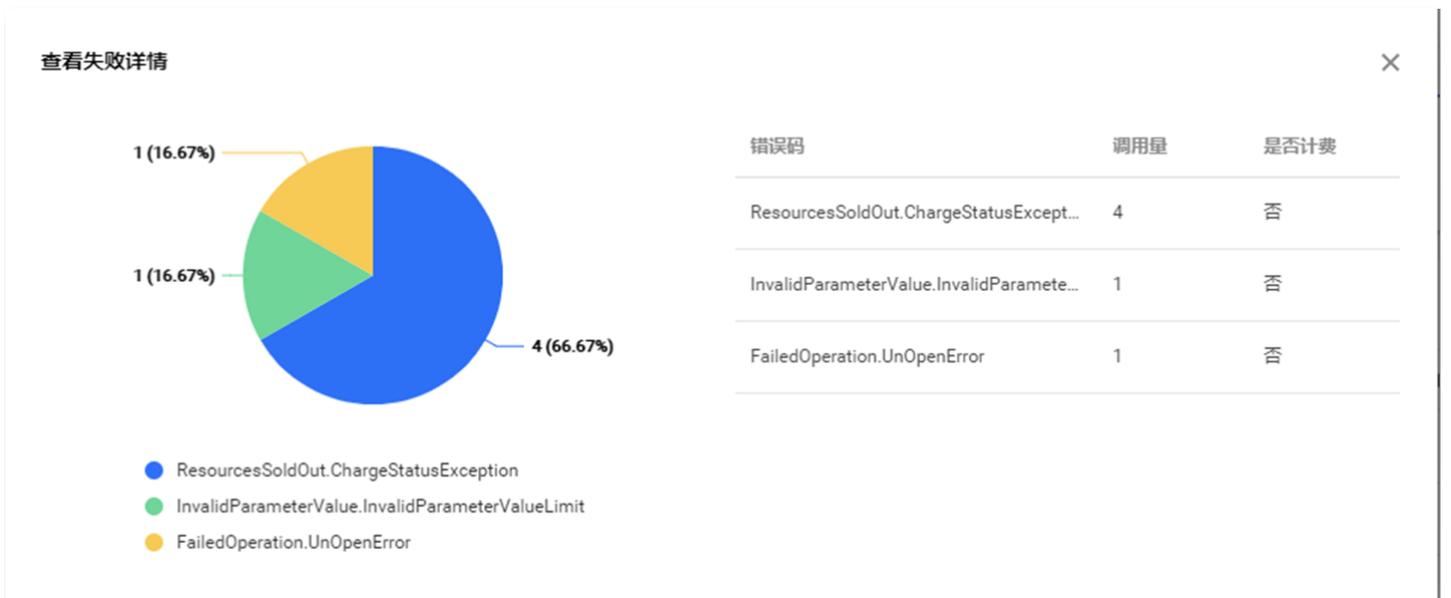
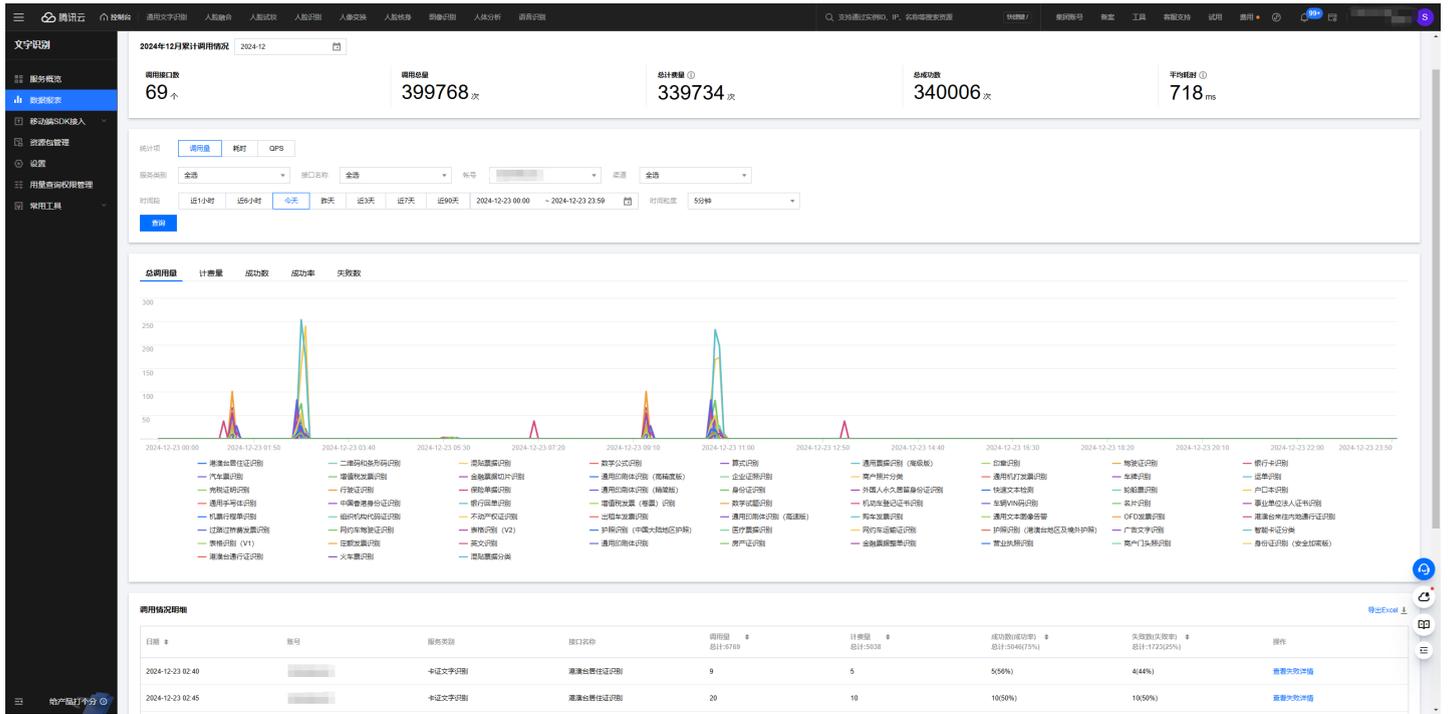
4. 产品动态及常用文档

浏览页右侧为您提供产品动态及常用文档、功能体验及售后服务等指引，您可以根据需要使用相关功能。



查看接口用量统计

进入数据列表页面，您可以按时间段、服务类别、接口名称、使用账号、渠道筛选，下方将展示所选条件下的总调用量、计费量、成功数、成功率、失败数的数据折线图及调用明细表，您可以查看图表一览接口使用情况，也可以将在调用明细表查看使用详情，可以单击查看调用失败详情，查看错误码分布；您也可以将调用明细导出 Excel 下载到本地进行下一步的数据分析。



文字识别客户端 SDK 接入申请

控制台提供文字识别客户端 SDK 申请入口，支持 iOS SDK、Android SDK、Harmony SDK 等下载。



智能扫码 SDK 接入申请

控制台提供智能扫码 SDK 申请入口，登录控制台后您可以申请测试版或正式版。其中智能扫码 SDK 测试版提供三个月免费测试期，提交申请后等待审核通过，您将获得专属密钥和 Demo 工程文件下载链接，一个账号只能申请一次。您也可以在线上购买直接获得正式版本接入使用。

说明

若未购买智能扫码 SDK，正式版本的申请将不予通过。



查看资源包使用明细

您可以在资源包管理页查看所有资源包使用情况，包含来源、总额、已使用及剩余量次数及有效期范围，当资源包即将耗尽时您可以通过上方购买资源包按钮进入购买资源包，若免费额度以及预付费资源包调用次数耗尽，接口计费将自动转为后付费方式按月进行结算，具体计费标准可以查看文字识别的 [购买页](#)。

购买资源包

资源包类型	来源	总额 (次)	已使用	剩余量	获取时间	到期时间
2020年11月免费资源包(22个)	赠送	22000	0 (0%)	22000 (100%)	2020-11-01 00:00:00	2020-11-30 23:59:59
图片文字识别 (基础版) -1000次免费资源包	赠送	1000	0 (0%)	1000 (100%)	2020-11-01 00:00:00	2020-11-30 23:59:59
通用印刷体识别 (高精度版) 免费资源包	赠送	1000	0 (0%)	1000 (100%)	2020-11-01 00:00:00	2020-11-30 23:59:59
公式识别免费资源包	赠送	1000	0 (0%)	1000 (100%)	2020-11-01 00:00:00	2020-11-30 23:59:59
通用证照识别-1000次免费资源包	赠送	1000	0 (0%)	1000 (100%)	2020-11-01 00:00:00	2020-11-30 23:59:59
表格识别免费资源包	赠送	1000	0 (0%)	1000 (100%)	2020-11-01 00:00:00	2020-11-30 23:59:59
通用印刷体识别 (高速版) 月免费资源包	赠送	1000	0 (0%)	1000 (100%)	2020-11-01 00:00:00	2020-11-30 23:59:59
行业文档识别-1000次免费资源包	赠送	1000	0 (0%)	1000 (100%)	2020-11-01 00:00:00	2020-11-30 23:59:59
英文文档识别月免费资源包	赠送	1000	0 (0%)	1000 (100%)	2020-11-01 00:00:00	2020-11-30 23:59:59
车辆识别月免费资源包	赠送	1000	0 (0%)	1000 (100%)	2020-11-01 00:00:00	2020-11-30 23:59:59
通用印刷体识别 (标准版) -1000次免费资源包	赠送	1000	0 (0%)	1000 (100%)	2020-11-01 00:00:00	2020-11-30 23:59:59
运单识别月免费资源包	赠送	1000	0 (0%)	1000 (100%)	2020-11-01 00:00:00	2020-11-30 23:59:59
增值税发票识别月免费资源包	赠送	1000	0 (0%)	1000 (100%)	2020-11-01 00:00:00	2020-11-30 23:59:59
通用票据识别 (大量) -1000次免费资源包	赠送	1000	0 (0%)	1000 (100%)	2020-11-01 00:00:00	2020-11-30 23:59:59
通用票据识别 (少量) -1000次免费资源包	赠送	1000	0 (0%)	1000 (100%)	2020-11-01 00:00:00	2020-11-30 23:59:59

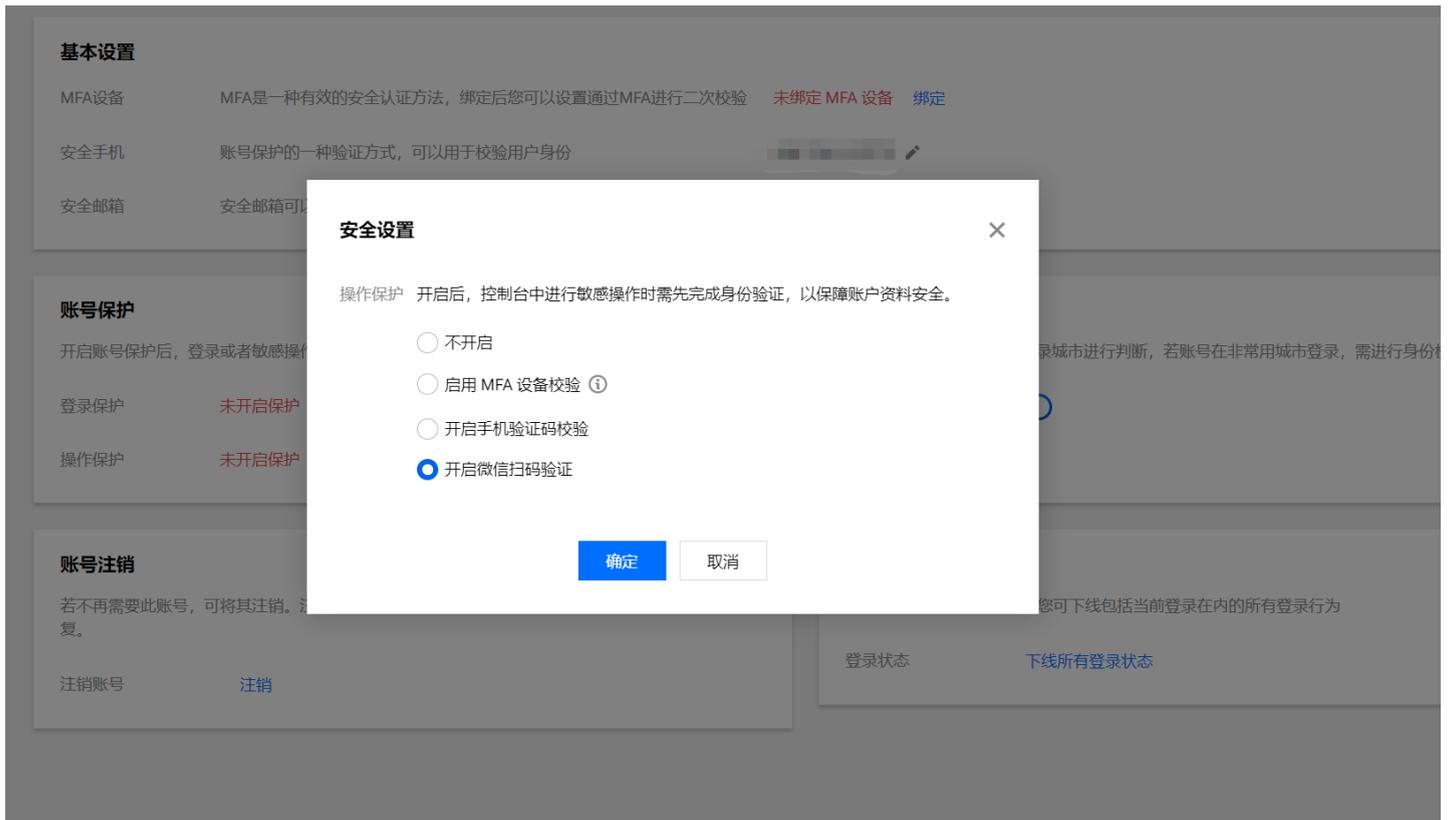
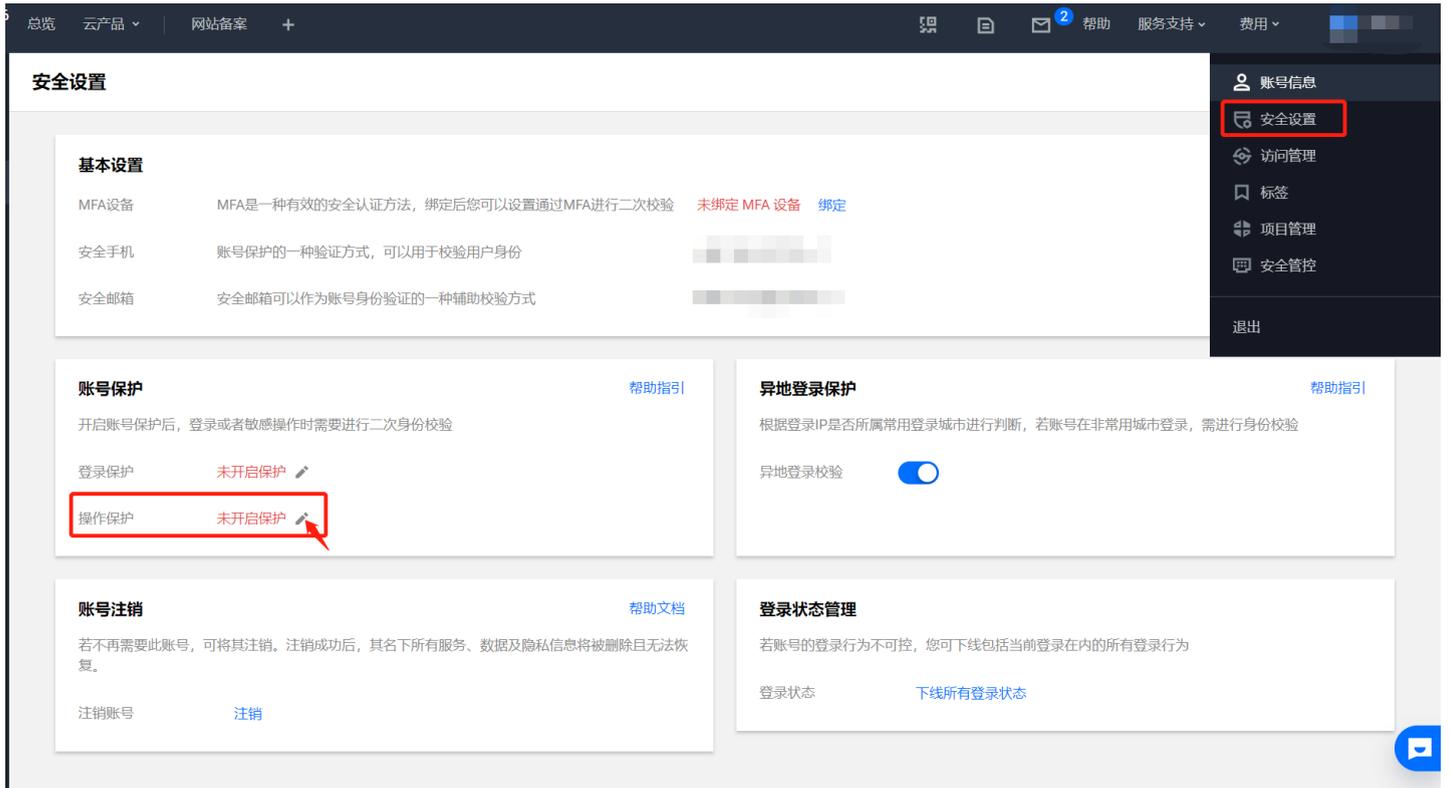
关闭文字识别服务

您可以在设置中关闭文字识别服务，关闭成功后，主账号及其下子账号将不可调用文字识别 API 和 SDK 服务（智能扫码 SDK 除外），恢复需重新开通服务。



关闭文字识别服务将可能影响您后续使用服务，属于敏感操作，建议您开启操作保护功能，在身份验证通过后才可进行功能设置，以确保是您本人操作。操作保护支持微信扫码校验、MFA 设备校验和手机验证码校验。建议开启微信扫码校验，步骤如下：

登录腾讯云账号后，单击头像下拉列表中的安全设置，进入安全设置页面，在账号保护模块下单击操作保护编辑功能，选择开启微信扫码验证，若您未关联微信号，扫码关联后即可开启。



账号保护 帮助指引

开启账号保护后，登录或者敏感操作时需要进行二次身份校验

登录保护 未开启保护

操作保护 已开启微信扫码校验

注意事项

1. 若您付费资源包未耗尽，将无法关闭文字识别服务。
2. 当您已关闭文字识别服务，再次进入控制台将无法查看原有数据使用情况，除非重新开通。再次开通后，可查看原先使用记录。

子账号如何在文字识别控制台上查看其他账号的调用量？

主账号可以查看名下所有子账号的调用量，子账号仅允许查看该子账号自身的调用量，若需要给某个子账号开通查看其他所有账号调用量的权限，需要进行以下操作（此页面目前仅白名单用户可以使用）：

1. 主账号登录文字识别控制台 [用量权限查询管理页面](#) 为该子账号进行授权，此页面涉及权限管理操作，**仅主账号具有访问权限**。

2. 用已授权子账号登录文字识别控制台 [用量统计页面](#)，即可选择其他子账号查看其调用量。

图片压缩示例

最近更新时间：2024-12-30 16:58:22

iOS 的压缩示例代码：

```
@implementation ImageCompressTool

/** 下面主要介绍了两种方式，具体使用的时候可以考虑两种方式合并使用 */

/**
 * 质量压缩图片
 * @param image 原始图片
 * @param quality 质量压缩的目标 0~1.0f
 * @return
 */
+ (UIImage *)doCompressQuality:(UIImage *)image quality:(CGFloat)quality {
    NSData * data = UIImageJPEGRepresentation(image, quality);
    return [UIImage imageWithData:data];
}

/**
 * 指定大小缩放一张图片
 * @param inputImage
 * @param newWidth 新的尺寸宽度
 * @param newHeight 新的尺寸高度
 * @return
 */
+ (UIImage *)scaleSizeImage:(UIImage *)inputImage newWidth:(CGFloat)newWidth newHeight:
(CGFloat)newHeight {
    UIGraphicsBeginImageContext(CGSizeMake(newWidth, newHeight));
    [inputImage drawInRect:CGRectMake(0.0f, 0.0f, newWidth, newHeight)];
    UIImage *newImage = UIGraphicsGetImageFromCurrentImageContext();
    UIGraphicsEndImageContext();
    // 如果要结合上面的方法，这里可以再做质量压缩
    return newImage;
}

@end
```

Android 的压缩示例代码：

```
/**
 * 图片压缩的帮助类型，可以参考的工具类
 *
 * @author jerrydong
 * @since 2023/5/23
 */
public class ImageCompressUtils {

    private static final String TAG = "ImageCompressUtils";

    // Jpg的压缩比,可以根据需求修改
    private static final int QUALITY_DEFAULT = 85;

    /**
     * 主动压缩YUV图片的算法
     */
}
```

```
*
* @param image 待压缩的图片
* @return
*/
public static byte[] doCompressYuvImage(YuvImage image) {
    int dataWidth = image.getWidth();
    int dataHeight = image.getHeight();
    try (ByteArrayOutputStream outputSteam = new ByteArrayOutputStream()) {
        image.compressToJpeg(new Rect(0, 0, dataWidth, dataHeight), QUALITY_DEFAULT, outputSteam);
        // 转Bitmap并压缩
        Options options = new BitmapFactory.Options();
        // 不使用透明通道
        options.inPreferredConfig = Config.RGB_565;
        // 修改图片的采样率
        options.inSampleSize = 2;
        Bitmap bmp = BitmapFactory.decodeByteArray(outputSteam.toByteArray(), 0,
outputSteam.size(), options);
        outputSteam.reset();
        // 主动压缩图片
        bmp.compress(CompressFormat.JPEG, QUALITY_DEFAULT, outputSteam);
        return outputSteam.toByteArray();
    } catch (IOException e) {
        Log.e(TAG, "Compress error");
    }
    return null;
}

/**
* 主动压缩BitMap
*
* @param inputMap
* @return
*/
public Bitmap doCompressBitMap(Bitmap inputMap) {
    try (ByteArrayOutputStream outputSteam = new ByteArrayOutputStream()) {
        inputMap.compress(CompressFormat.JPEG, QUALITY_DEFAULT, outputSteam);
        // 转Bitmap并压缩 (可以写通用方法!)
        Options options = new BitmapFactory.Options();
        // 不使用透明通道
        options.inPreferredConfig = Config.RGB_565;
        // 修改图片的采样率 (可以根据实际情况计算)
        options.inSampleSize = 2;
        Bitmap bmp = BitmapFactory.decodeByteArray(outputSteam.toByteArray(), 0,
outputSteam.size(), options);
        outputSteam.reset();
        return bmp;
    } catch (IOException e) {
        Log.e(TAG, "Compress error");
    }
    return null;
}

/**
* 缩放方法进行缩放
*
* @param inputMap
* @param scale 缩放大小
* @return
```

```
*/
public Bitmap matrixBitmap(Bitmap inputMap, float scale) {
    // 根据旋转角度,生成旋转矩阵
    Matrix matrix = new Matrix();
    matrix.setScale(scale, scale);
    Bitmap bitmap = Bitmap.createBitmap(inputMap, 0, 0, inputMap.getWidth(), inputMap.getHeight(),
matrix, true);
    return bitmap;
}
}
```

控制台访问管理

最近更新时间：2025-03-27 10:57:02

文字识别已接入了腾讯云资源访问管理系统 CAM（Cloud Access Management），您可以在 [访问管理控制台](#) 进行用户组、用户、角色、策略等一系列相关管理操作。

通过设置 CAM 策略，您可以控制哪些子用户或协作者账号拥有文字识别 API 和控制台的查看和操作权限。

目前文字识别支持预设策略和自定义策略两种方式。

注意：

2023年9月21日之前开通文字识别服务的客户暂不支持使用新版控制台访问管理功能，如有需要请 [联系我们](#) 添加配置。

预设策略

说明

文字识别的预设策略分为以下3个，全读写权限、只读权限和只读自己权限。

- QcloudOCRFullAccess 全读写权限。**将预设的全读写策略授予子账户时，其将能够和您一样访问文字识别所有功能。
- QcloudOCRReadOnlyaccess 只读权限。**将预设的只读策略授予子账户时，子账户可以访问识别类、核验类接口，并可以访问文字识别控制台的只读接口，例如查看调用量、查看资源包等。
- QcloudOCRReadSelfUiUsage 只读自己权限。**将预设的只读自己策略授予子账户时，子账户可以访问识别类、核验类接口，并可以访问文字识别控制台的只读接口，例如查看调用量、查看资源包等。但查看调用量、耗时、QPS和智能SDK申请列表时，只能查询子账号自己的。

预设策略的区别：

大类别	类别细分	全读写 (QcloudOCRFullAccess)	只读 (QcloudOCRReadOnlyaccess)	只读自己 (QcloudOCRReadSelfUiUsage)
	识别类接口	✓	✓	✓
	核验类接口	✓	✓	✓
读接口	查看调用量\耗时 \QPS\查看智能扫码 SDK申请列表	✓	可查看全部账号的用量和SDK 申请列表	仅查看自己账号的用量和SDK申请 列表
	其他读接口	✓	✓	✓
写接口	申请智能扫码 SDK	✓	×	×
	开通服务	✓	×	×
	关闭服务	✓	×	×
	开通/关闭后付费	✓	×	×
	其他写接口	✓	×	×

配置方式

预设策略可以在 [访问管理控制台](#) 中进行授权。若无特殊的权限要求，推荐使用预设策略。

方式一：如下图所示，可搜索对应的预设策略，请根据策略描述关联用户、组或者角色：



方式二：如图所示，可在子用户中，关联所需的预设策略：



自定义策略

说明

文字识别的自定义策略支持接口级和资源级授权两种方式：

1. 接口级授权：您可以在 CAM 控制台中按接口维度给予子账户授权，子账户将只能使用您已授权的接口。可参考 [使用示例](#) 中授予单个接口权限的操作。
2. 资源级授权：调用量、耗时以及QPS的查看支持资源级授权，您可以在 CAM 控制台中按子账户维度给予子账户授权，子账户将只能查看您已授权子账户的调用量、耗时以及QPS。

支持资源级授权的接口列表

API 名	API 描述	资源类型	资源六段式示例
QueryQpsForConsole	查询QPS数据	子账户 ID	qcs::ocr:::subUin/* qcs::ocr:::subUin/\${uin}
QueryCallForConsole	查询调用量数据	子账户 ID	qcs::ocr:::subUin/* qcs::ocr:::subUin/\${uin}
QueryCostTimeForConsole	查询耗时数据	子账户 ID	qcs::ocr:::subUin/* qcs::ocr:::subUin/\${uin}
GetIntelligentScanSDKListForConsole	查询智能扫码SDK列表	子账户 ID	qcs::ocr:::subUin/* qcs::ocr:::subUin/\${uin}

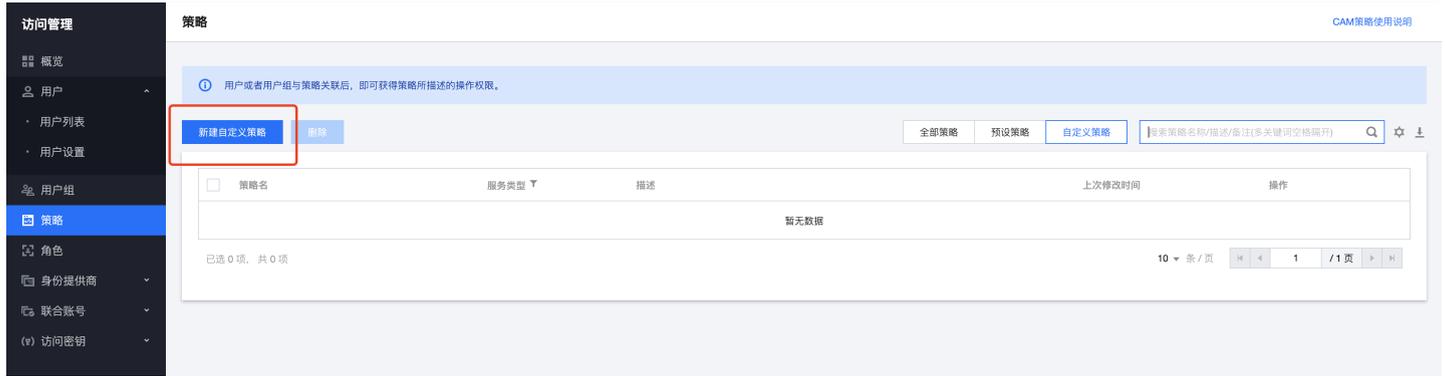
注意:

其他接口仅支持接口级授权。

配置方式

自定义策略也需要在 [访问管理控制台](#) 中进行授权，和预测策略一致。

创建自定义策略的方式可参考 [通过策略生成器创建自定义策略](#)。



- 创建自定义策略时，接口级接口、资源级接口均遵循一下规则。
- resource 字段填 " * "，相当于授予 action 里的接口所有的权限。
- 接口级的接口resource必须配置 " * "才可使用。
- 资源级的接口resource可配置 " * " 或通过资源六段式进行配置。
- 权限范围大的策略会覆盖权限范围小的策略，例如资源级的接口resource字段填 " * "，则对应该接口配置的资源六段式就不会生效了。

自定义策略示例

- 授予单个接口权限

以下为给予账户授予开通服务（ ConsoleServiceOpen ）、关闭服务（ ConsoleServiceSetting ）授权。在 action 中填入对应的接口名：

```

{
  "version": "2.0",
  "statement": [
    {
      "effect": "allow",
      "resource": [
        "*"
      ],
      "action": [
        "ocr:ConsoleServiceOpen",
        "ocr:ConsoleServiceSetting"
      ]
    }
  ]
}
    
```

其中 action 可以指定多个接口和支持通配符 * 。上述的授权方式也可以简化为以下的方式：

```

{
  "version": "2.0",
  "statement": [
    {
      "effect": "allow",
      "resource": [
        "*"
      ],
    }
  ]
}
    
```

```
    "action": [
      "ocr:ConsoleService*"
    ]
  }
]
```

- 授予接口的某些资源权限

以下为给子账户授予可查询子用户（subuin:0001）、子用户（subuin:0002）、以及自己的调用量数据。

```
{
  "version": "2.0",
  "statement": [
    {
      "effect": "allow",
      "resource": [
        "qcs::ocr::subUin/${uin}",
        "qcs::ocr::subUin/0001",
        "qcs::ocr::subUin/0002",
      ],
      "action": [
        "ocr:QueryCallForConsole"
      ]
    }
  ]
}
```

敏感数据加密指引

最近更新时间：2024-12-30 16:58:22

此指引适用于使用文字识别服务接口，需要对传输敏感数据进行加密的场景。

如果您使用的开发语言是 Java、Go、Nodejs、Python，推荐使用 [方式一](#) 实现加解密。如果开发语言不在上述范围，推荐使用 [方式二](#) 实现加解密功能。

方式一：使用官方 Demo 实现加解密（推荐）

官方 Demo 当前支持 Java、Go、Nodejs、Python 开发语言，使用步骤如下：

获取加解密 SDK

联系文字识别 [售后支持](#)，获取对应开发语言的 Demo 代码。

使用 SDK

引入公共依赖

```
<dependency>
  <groupId>org.bouncycastle</groupId>
  <artifactId>bcprov-jdk15on</artifactId>
  <version>1.70</version>
</dependency>
<dependency>
  <groupId>org.bouncycastle</groupId>
  <artifactId>bcpkix-jdk15on</artifactId>
  <version>1.70</version>
</dependency>
```

引入加解密 SDK

下载最新的 [release](#) 版本 jar 包，并在项目工程中引入。

参考下方的接口 Demo 实现敏感信息加解密功能。

支持敏感信息加解密的接口

- 身份证识别（安全加密版）：[RecognizeEncryptedIDCardOCR](#)

方式二：不使用官方 SDK 实现加解密

加解密流程说明

- 当前敏感信息加解密支持标准加密算法 AES-CBC 和国密算法 SM4-GCM，可以根据业务要求从中选择适合您的加密算法。
- 敏感信息加解密的本质是对接口传输的请求、响应参数做对称加密后传输，文字识别服务收到加密数据后先解密，然后在进行识别业务。如果文字识别接口返回结果涉及敏感信息，则会使用相同的对称密钥加密，调用方在收到接口响应时需要敏感信息解密。
- 对称密钥由调用方本地随机生成，为保证对称密钥的安全传输，需要调用方配合使用非对称加密算法加密对称密钥。加密的公钥可以联系 [售后支持](#) 获取。

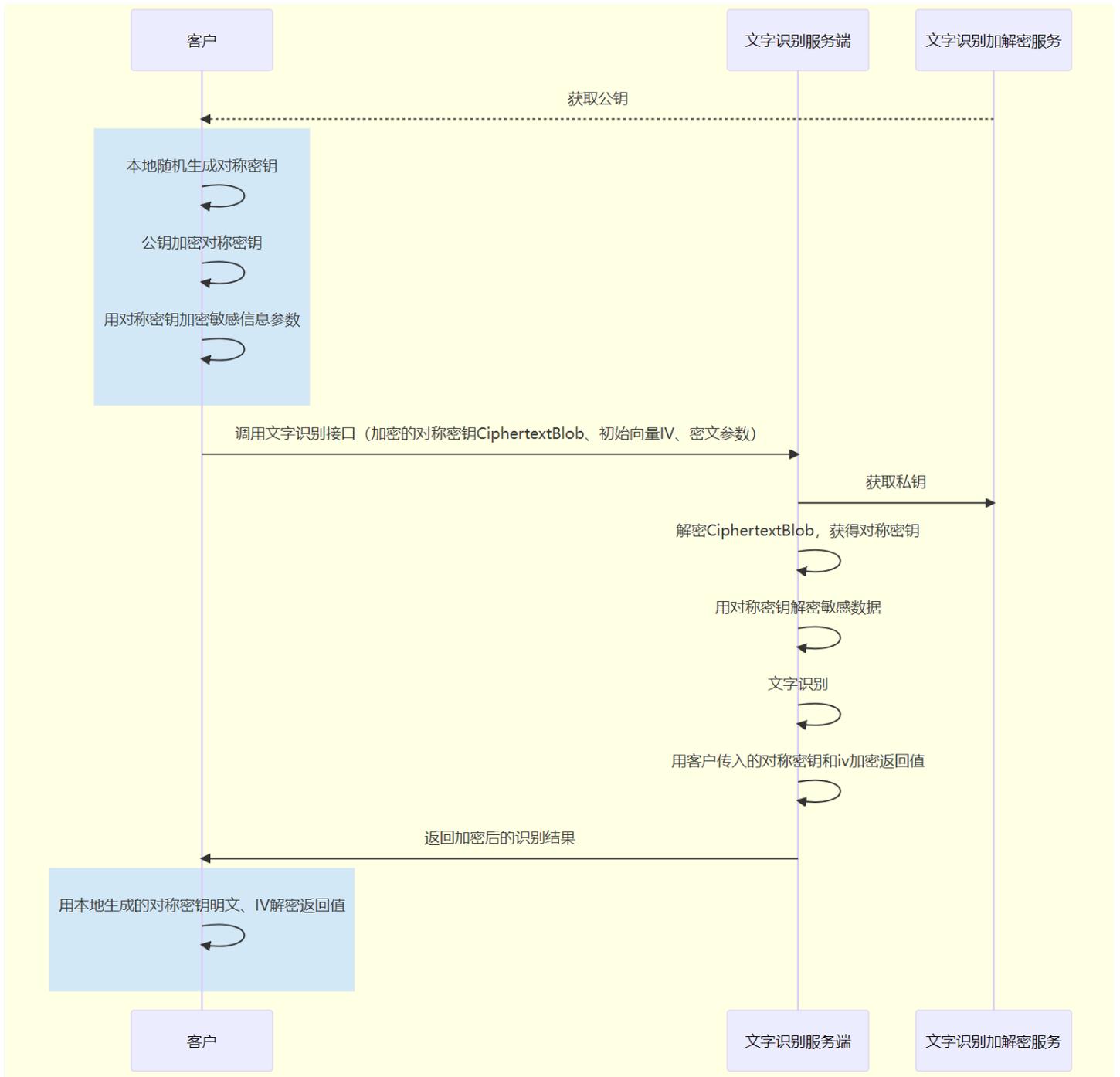
```
sequenceDiagram
    participant customer as 客户
    participant faceid as 文字识别服务端
    participant console as 文字识别加解密服务
    console-->>customer: 获取公钥

    rect rgb(191, 223, 255, 0.7)
    customer->>customer: 本地随机生成对称密钥
    customer->>customer: 公钥加密对称密钥
    customer->>customer: 用对称密钥加密敏感信息参数
```

```

end

customer->>faceid: 调用文字识别接口（加密的对称密钥CiphertextBlob、初始向量IV、密文参数）
faceid->>console: 获取私钥
faceid->>faceid: 解密CiphertextBlob，获得对称密钥
faceid->>faceid: 用对称密钥解密敏感数据
faceid->>faceid: 文字识别
faceid->>faceid: 用客户传入的对称密钥和iv加密返回值
faceid->>customer: 返回加密后的识别结果
rect rgba(191, 223, 255, 0.7)
customer->>customer: 用本地生成的对称密钥明文、IV解密返回值
end
    
```



使用 AES-256-CBC 算法

加载 RSA 公钥

加载 RSA 公钥：对获取到的公钥字符串先做 Base64 Decode，然后按对应格式加载。

- 格式：PKCS1
- 保存格式：pem 格式 Base64编码
- 长度：1024

```
// 加载RSA公钥
bytes, err := base64.StdEncoding.DecodeString(publicKey)
if err != nil {
    return nil, err
}
block, _ := pem.Decode(bytes)
x509.ParsePKCS1PublicKey(block.Bytes)
```

生成对称密钥和向量

随机生成32字节长度的对称密钥以及16字节的初始向量，初始向量表示为 IV。

```
// 生成对称密钥
key := make([]byte, 32)
rand.Read(key)

// 生成16字节iv
iv := make([]byte, 16)
rand.Read(iv)
```

RSA 公钥加密对称密钥

使用 RSA 公钥加密前面生成的对称密钥，将加密后的结果表示为 CiphertextBlob。

```
// 加密对称密钥Key
buffer := bytes.Buffer{}
bytes, _ := rsa.EncryptPKCS1v15(rand.Reader, publicKey, key)
buffer.Write(bytes)
buffer.Bytes()
```

加密敏感数据

使用前面生成的对称密钥和初始向量，采用 AES-256-CBC 算法加密敏感数据（姓名、身份证号）。

```
// AES-CBC加密明文数据
// 明文密钥key
// 初始向量iv
// 敏感数据plaintext
block, _ := aes.NewCipher(key)
blockSize := block.BlockSize()
pkcs7 := func(cipherText []byte, blockSize int) []byte {
    padding := blockSize - len(cipherText)%blockSize
    padText := bytes.Repeat([]byte{byte(padding)}, padding)
    return append(cipherText, padText...)
}
plaintext = pkcs7(plaintext, blockSize)
blockMode := cipher.NewCBCEncrypter(block, iv)
ciphertext := make([]byte, len(plaintext))
blockMode.CryptBlocks(ciphertext, plaintext)
```

```
fmt.Print(ciphertext)
```

调用文字识别接口

以 **身份证识别（安全加密版）** RecognizeEncryptedIDCardOCR 接口为例：

原始数据为：

```
{
  "Action": "RecognizeEncryptedIDCardOCR",
  "Version": "2018-11-19",
  "ImageUrl": "https://xx/a.jpg",
  "Config": "{\"TempIdWarn\":true}",
  "CardSide": "FRONT"
}
```

CiphertextBlob、IV、TAG、加密后请求入参在传输前均需要做一次 Base64 编码：

- 生成的对称密钥为：aqPpVPUV9XMCCr5QQSu4p6ml8BAgbMz3（32字节）
- 生成的初始向量（IV）：Pcr1F3d9Bkn6ub7jduJcBw==（16字节IV，Base64编码）
- 公钥加密后的对称密钥（CiphertextBlob）：
Ows39pc3d7x34m/g6uWmo/HlQbxw+1IA4C1QiTCbbYuMt1W3tzxG7Wp3UEAB38kE1PTgUt2tHKOE8FDboEbkR9Bgnti+Hj1aNiG0lxii9ABz6vTKA4Rx3F1FvnsM9+PulDPQWb7PgCEYIYWOAXNtfaa6iDAMW+qT0H8LBEg4Mss=（Base64编码）

使用对称密钥加密请求入参，加密后的结果分别为：

- 加密后的请求入参 EncryptedBody：
MrUQbU7Tkva7sWSxEScx7U9zOikFtM5fCf7weC21z96kPE93MFnGbbMKR+CgsTSClpqmfBKSDeErek4HAyyRGwzctETaJOJ7cFD8E4gBXk0R7yZhFn9INamfTMJusIhWGkmXPP0/+lrZoqGhOm01FNTAaUf6qHNwWUXOiaMntBSRmC5vscx9+arg4IW9XtTwENsK2ho68s7bw21uKNNJPw==（Base64编码）
- 最终加密后的接口数据为：

```
{
  "Action": "RecognizeEncryptedIDCardOCR",
  "Version": "2018-11-19",
  "ImageUrl": "",
  "Config": "",
  "CardSide": "",
  "EncryptedBody":
  "MrUQbU7Tkva7sWSxEScx7U9zOikFtM5fCf7weC21z96kPE93MFnGbbMKR+CgsTSClpqmfBKSDeErek4HAyyRGwzctETaJOJ7cFD8E4gBXk0R7yZhFn9INamfTMJusIhWGkmXPP0/+lrZoqGhOm01FNTAaUf6qHNwWUXOiaMntBSRmC5vscx9+arg4IW9XtTwENsK2ho68s7bw21uKNNJPw==",
  "Encryption": {
    "EncryptList": [
      "EncryptedBody"
    ],
    "TagList": [],
    "CiphertextBlob":
    "Ows39pc3d7x34m/g6uWmo/HlQbxw+1IA4C1QiTCbbYuMt1W3tzxG7Wp3UEAB38kE1PTgUt2tHKOE8FDboEbkR9Bgnti+Hj1aNiG0lxii9ABz6vTKA4Rx3F1FvnsM9+PulDPQWb7PgCEYIYWOAXNtfaa6iDAMW+qT0H8LBEg4Mss=",
    "Iv": "Pcr1F3d9Bkn6ub7jduJcBw==",
    "Algorithm": "AES-256-CBC"
  }
}
```

使用 SM4-GCM 算法

加载 SM2公钥

加载 SM2公钥：对控制台获取到的公钥字符串先做 Base64 Decode，然后按对应格式加载。

- 格式: PKCS8
- 保存格式: pem格式Base64编码
- 长度: 512

```
// 加载SM2公钥
bytes, err := base64.StdEncoding.DecodeString(publicKey)
if err != nil {
    return nil, err
}
x509.ReadPublicKeyFromPem(bytes)
```

生成对称密钥和向量

随机生成16字节长度的对称密钥以及12字节的初始向量，初始向量表示为 IV。

```
// 生成对称密钥
key := make([]byte, 16)
rand.Read(key)

// 生成16字节iv
iv := make([]byte, 12)
rand.Read(iv)
```

SM2 公钥加密对称密钥

使用 SM2 公钥加密前面生成的对称密钥，将加密后的结果表示为 CiphertextBlob。

- 注意输出密文时采用 C1C3C2数据拼接方式
- 数据类型转换时，均使用 PC=04的方式，即未压缩。

```
// 加密对称密钥key
ciphertext, err := sm2.Encrypt(publicKey, plaintext, rand.Reader, sm2.C1C3C2)
if err != nil {
    return nil, err
}
return ciphertext, nil
```

加密敏感数据

使用前面生成的对称密钥和初始向量，采用 SM4-GCM 算法加密敏感数据（请求入参）。

```
// SM4-GCM加密明文数据
// 明文密钥key
// 初始向量iv
// 敏感数据plaintext
block, err := sm4.NewCipher(key)
if err != nil {
    return nil, nil, err
}
gcm, err := cipher.NewGCM(block)
if err != nil {
    return nil, nil, err
}
ciphertext := gcm.Seal(nil, iv, plaintext, nil)
tag := ciphertext[len(ciphertext)-gcm.Overhead():]
ciphertextWithoutTag := ciphertext[:len(ciphertext)-gcm.Overhead()]
return ciphertextWithoutTag, tag, nil
```

调用文字识别接口

以身份证识别（安全加密版）RecognizeEncryptedIDCardOCR 接口为例：

原始数据为：

```
{
  "Action": "RecognizeEncryptedIDCardOCR",
  "Version": "2018-11-19",
  "ImageUrl": "https://xx/a.jpg",
  "Config": "{\"TempIdWarn\":true}",
  "CardSide": "FRONT"
}
```

CiphertextBlob、IV、TAG、加密后的请求参数在传输前均需要做一次Base64编码。

- 生成的对称密钥为：zpBn0BkD44bvJuSF（16字节）
- 生成的初始向量（IV）：MUZsKuqk7N2u7Ayh（12字节IV，Base64编码）
- 公钥加密后的对称密钥（CiphertextBlob）：
BHknHJ1Wgzg8DAONoGJ7ch+ijF/rsrs7mmvxER1WlFKSJHY66BSJbmzc5OtQyVK84Ye2zAD2x63nmr4ZXoEyw1T2zqKcAbDWP2DTypjsbijvqdU6oSVOrlWE3TlQpA+k6ID3YoRdNdJOCjKnsMOnEls=（Base64编码）

使用对称密钥加密请求入参，加密后的结果分别为：

- 加密后的请求入参EncryptedBody：
12nTF6/CcyMqUDmbU63FPMFRpsye8NdZDwfGceRZljpSr9CIHKAqWJBhcbqOGolvuOqG+ES1Eru94+5NIJtK/1UE+BgUb9+1571TgS/7Nn0d8zpR320ZZoFZO3djdpkkyXL32sM3FyuwdOkQXMOYcS+3y8A8YCOsljDL6L7rAC2fP8sqqWWGndZAUdyK9fshwZ+rbCtf3zMI0uWY（Base64编码）
- 最终加密后的接口数据为：

```
{
  "Action": "RecognizeEncryptedIDCardOCR",
  "Version": "2018-11-19",
  "ImageUrl": "",
  "Config": "",
  "CardSide": "",
  "EncryptedBody":
  "12nTF6/CcyMqUDmbU63FPMFRpsye8NdZDwfGceRZljpSr9CIHKAqWJBhcbqOGolvuOqG+ES1Eru94+5NIJtK/1UE+BgUb9+1571TgS/7Nn0d8zpR320ZZoFZO3djdpkkyXL32sM3FyuwdOkQXMOYcS+3y8A8YCOsljDL6L7rAC2fP8sqqWWGndZAUdyK9fshwZ+rbCtf3zMI0uWY",
  "Encryption": {
    "EncryptList": [
      "EncryptedBody"
    ],
    "TagList": [],
    "CiphertextBlob":
    "BHknHJ1Wgzg8DAONoGJ7ch+ijF/rsrs7mmvxER1WlFKSJHY66BSJbmzc5OtQyVK84Ye2zAD2x63nmr4ZXoEyw1T2zqKcAbDWP2DTypjsbijvqdU6oSVOrlWE3TlQpA+k6ID3YoRdNdJOCjKnsMOnEls=",
    "Iv": "MUZsKuqk7N2u7Ayh",
    "Algorithm": "AES-256-CBC"
  }
}
```