



# 内容安全

## 产品文档





# 文档目录

## 产品简介

产品概述

功能优势

节约人力

快速便捷

应用场景

社区论坛

即时通讯

电子商务

约束与限制

## 快速入门

快速入门

## API文档

API概览

请求结构

服务地址

通信协议

公共参数

使用签名方法 v1 的公共参数

签名方式

申请安全凭证

申请安全凭证

生成签名串

概述

对参数排序

拼接请求字符串

拼接签名原文字符串

生成签名串

签名串编码

签名失败

签名演示

Python签名演示

反馈结果

返回结果

正确返回结果



错误返回结果

错误码

公共错误码

业务错误码

文本内容安全相关接口

接口描述

输入参数

输出参数

错误码

接口请示示例

POST文本请求示例

图片内容安全相关接口

接口描述

输入参数

输出参数

错误码

接口请求示例

POST图片请求示例

常见问题

文本内容安全、旅途片内容安全支持识别哪些违规内容？

如何接入文本内容安全、图片内容安全？

文本内容安全、图片内容安全的 API 检测效率如何？

文本内容安全、图片内容安全支持检测哪几种语言？

图片base64转码上传服务器，总是返回-1003？



# 产品简介

## 产品概述

最近更新时间: 2022-10-26 10:50:13

建行云内容安全服务包含文本内容安全和图片内容安全：

- **文本内容安全 (Text Moderation System, TMS)** 服务使用了深度学习技术，可有效识别涉黄、涉政、涉恐等有害内容，支持用户配置词库，打击自定义的违规文本。通过 API 接口，能检测内容的危险等级，对于高危部分直接过滤，可疑部分人工复审，从而节省审核人力，规避业务风险。
- **图片内容安全 (Image Moderation System, IMS)** 能自动扫描图片，识别涉黄、涉恐、涉政、涉毒等违规内容，同时支持图片黑名单配置，打击自定义的违规图片。通过 API 获取检测的标签及置信度，用户可直接采信置信度高的结果，人工复审置信度低的结果，从而降低人工成本，提高审核效率。



# 功能优势 节约人力

最近更新时间: 2020-07-12 13:43:00

已经为多家平台提供图片内容安全服务，最多减少90%的人力审核成本。



# 快速便捷

最近更新时间: 2020-07-12 13:43:00

服务毫秒级响应，每秒超过万级并发。无需安装任何脚本文件，通过 API 方式即可直接使用，只需三步轻松接入。



# 应用场景

## 社区论坛

最近更新时间: 2020-07-12 13:43:00

内容安全可以广泛应用于 BBS、博客，以及有用户 UGC 内容的各类网站，包括发帖、回帖、站内信等场景。实时检测文本中的涉黄、涉政、广告、谩骂等垃圾文本。



# 即时通讯

最近更新时间: 2020-07-12 13:43:00

内容安全可以针对性地检测识别昵称、头像、签名、C2C 消息、群发消息等信息中藏匿的垃圾内容，防止恶意用户骚扰，预防风险诈骗。对聊天群内传播的消息进行管控，拦截打击违法信息，规避平台运营风险。





# 电子商务

最近更新时间: 2020-07-12 13:43:00

有效识别电子商务场景中的商品图片、买家秀，拦截敏感内容，提高人工审核效率。



# 约束与限制

最近更新时间: 2020-07-12 13:37:17

- 仅限租户在overlay层调用内容安全接口，无法通过互联网调用。
- 文本大小不能超过600k。
- 图片大小不能超过1M；需将图片转码为base64格式，再通过post方式提交。



# 快速入门

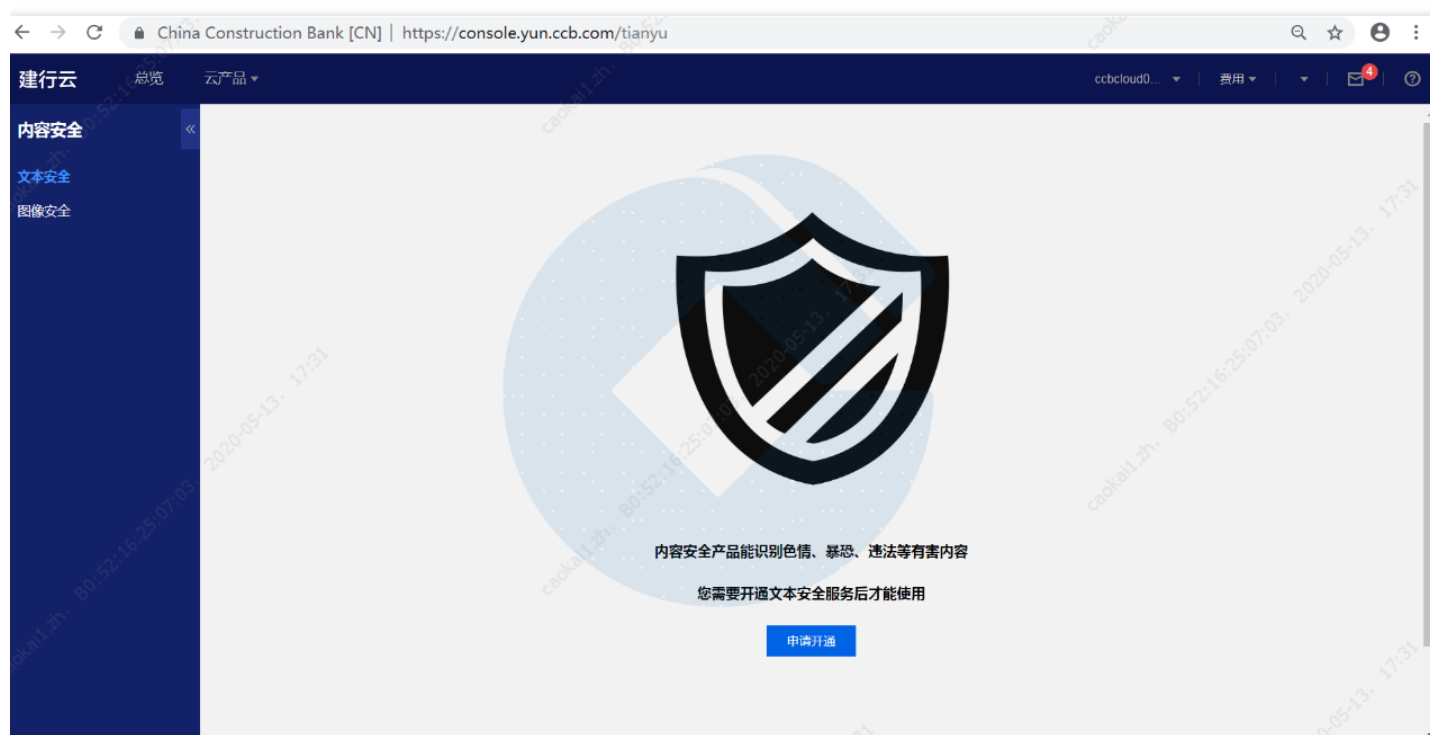
# 快速入门

最近更新时间: 2020-07-12 13:50:24

如果您已拥有建行云的租户账号，只需几分钟可以申请。

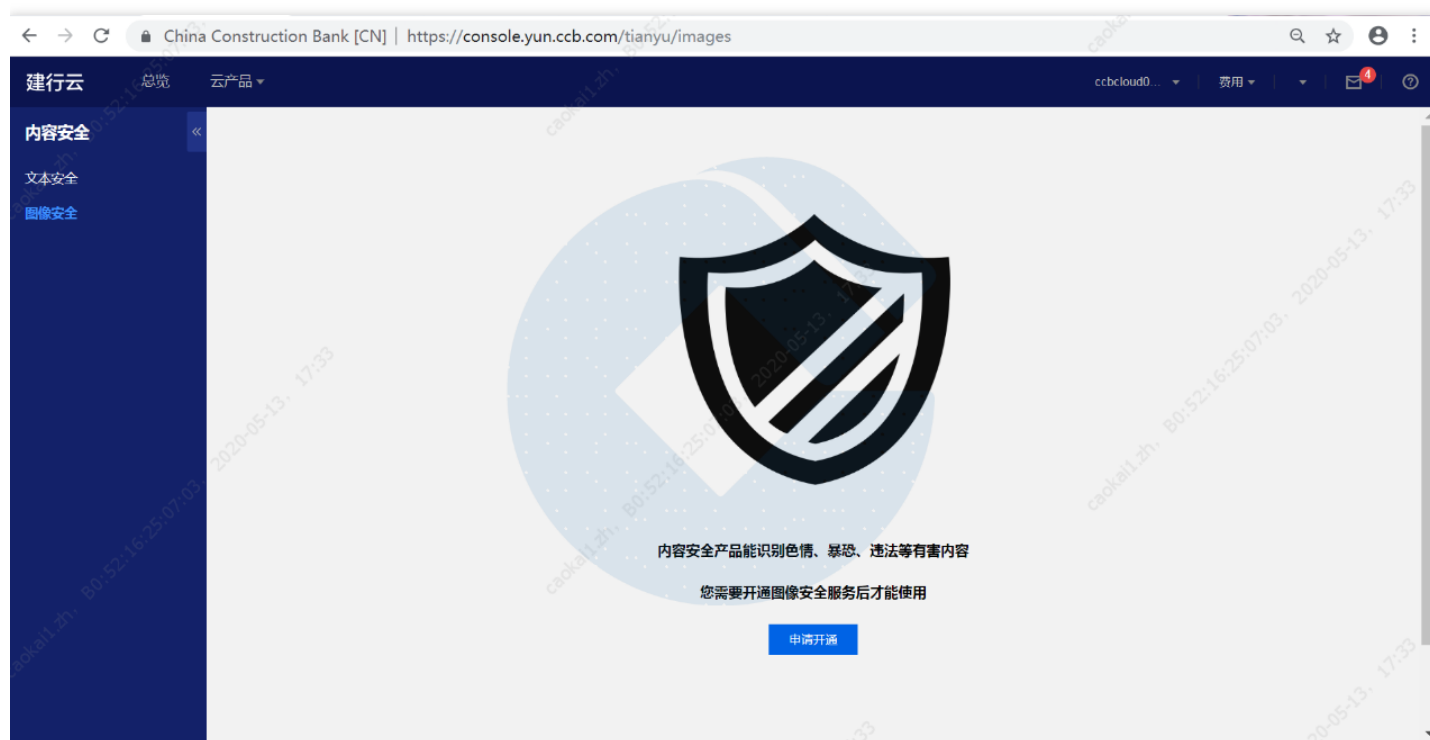
## 步骤一：线上开通

登录 内容安全控制台，找到文本安全，单击【申请开通】，即可开通服务。开通以后，可以查看调用量信息。





登录 内容安全控制台，找到图像安全，单击【申请开通】，即可开通服务。



## 步骤二：调用检查

可点击图像安全，文本安全查看调用防护总量和恶意占比





# API文档

## API概览

最近更新时间: 2022-10-26 11:03:10

内容安全相关接口如下:

接口名称	接口功能
BspTextRecognition	BsplImageRecognition
文本内容安全	图片内容安全



# 请求结构

## 服务地址

最近更新时间: 2022-10-26 14:05:39

本产品接入域名:

区域	域名
标准武汉一区、标准武汉二区	contentsecurity.api3.yun.ccb.com
自用北京一区、自用北京二区	contentsecurity.api3.yun004.ccb.com



# 通信协议

最近更新时间: 2022-10-26 14:05:39

建行云API 的所有接口均通过 HTTPS 进行通信，提供高安全性的通信通道，包含：

- □ GET 请求的请求包大小不得超过32KB。
- □ （推荐）POST 请求支持的 Content-Type 类型：application/x-www-form-urlencoded，必须使用签名方法 v1（HmacSHA1）。
- □ POST 请求使用签名方法 v1（HmacSHA1）时，文本不得超过600k，图片不得超过1MB。
- □ 均使用UTF-8编码。



# 公共参数

最近更新时间: 2022-10-26 14:05:39

公共参数是用于标识用户和接口签名的参数，如非必要，在每个接口单独的接口文档中不再对这些参数进行说明，但每次请求均需要携带这些参数，才能正常发起请求。

公共参数的具体内容会因您使用的签名方法版本不同而有所差异。





# 使用签名方法 v1 的公共参数

最近更新时间: 2022-10-26 15:20:32

使用签名方法 v1 （有时会称作 HmacSHA256 和 HmacSHA1），公共参数需要统一放到请求串中，完整介绍详见《签名方法》文档部分详细说明。

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	- -	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在云API密钥上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中输入公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	即安全凭证服务所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

假设用户想要查询广州地域的云服务器实例列表，其请求结构按照请求 URL、请求头部、请求体示例如下：



---

HTTP GET 请求结构示例:

---

[https://contentsecurity.api3.yun.ccb.com/?Action=BspTextRecognition&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=wh&Nonce=23823223&SecretId=AKID\\*\\*\\*\\*\\*](https://contentsecurity.api3.yun.ccb.com/?Action=BspTextRecognition&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=wh&Nonce=23823223&SecretId=AKID*****)

EXAMPLE

Host: contentsecurity.api3.yun.ccb.com Content-Type: application/x-www-form-urlencoded

---

HTTP POST 请求结构示例:

---

[https:// contentsecurity.api3.yun.ccb.com/](https://contentsecurity.api3.yun.ccb.com/)

Host: contentsecurity.api3.yun.ccb.com Content-Type: application/x-www-form-urlencoded

Action=BspTextRecognition&Version=2017-03-

12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=wh&Nonce=23823223&SecretId=AKID\*\*\*\*EXAMPLE

---



# 签名方式

最近更新时间: 2022-10-26 11:03:10

签名方法 v1 简单易用。

首次接触，建议使用 API Explorer 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v1”，可以生成签名过程进行验证，并提供了部分编程语言的签名示例，也可直接生成 SDK 代码。推荐使用建行云 API 配套的 7 种常见的编程语言，支持 Python、Java、PHP、Go、NodeJS、.NET、C++。

建行云 API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证请求者身份。签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往 [云API密钥页面](#) 申请，否则无法调用云 API 接口。



# 申请安全凭证

## 申请安全凭证

最近更新时间: 2022-10-26 15:20:32

使用的安全凭证为密钥，密钥包括 SecretId 和 SecretKey。

- SecretId：用于标识 API 调用者身份，可以简单类比为用户名。
- SecretKey：用于验证 API 调用者的身份，可以简单类比为密码。
- 用户必须严格保管安全凭证，避免泄露，否则将危及财产安全。如已泄露，请立刻禁用该安全凭证。

申请安全凭证的具体步骤如下：

1. 登录 建行云管理中心控制台 。
2. 前往 【云 API 密钥】 的控制台页面
3. 在 【云 API 密钥】 页面，单击【新建密钥】即可以创建一对 SecretId/SecretKey。



# 生成签名串

## 概述

最近更新时间: 2022-10-27 10:30:02

有了安全凭证SecretId 和 SecretKey后，就可以生成签名串了。以下是生成签名串的详细过程：  
假设用户的 SecretId 和 SecretKey 分别是：

□ SecretId: AKIDz8krbsJ5yKBZQp \*\*\*\*\* □ SecretKey: Gu5t9xGARNpq86c \*\*\*\*\*

注意：这里只是示例，请根据用户实际申请的 SecretId 和 SecretKey 进行后续操作！

以云服务器查看实例列表请求为例，当用户调用这一接口时，其请求参数可能如下：

### Timestamp

参数名称	中文	参数值
Action	方法名	BspTextRecognition或BsplmageRecognition
SecretId	密钥 ID	AKIDz8krbsJ5yKBZQp*****
当前时间戳	1465185768	
Nonce	随机正整数	11886
Region	实例所在区域	wh
Version	接口版本号	2019-03-05

这里只展示了部分公共参数和接口输入参数，用户可以根据实际需要添加其他参数，例如 Language 和 Token 公共参数。



# 对参数排序

最近更新时间: 2022-10-27 10:30:02

首先对所有请求参数按参数名的字典序（ASCII 码）升序排序。

注意： 1) 只按参数名进行排序，参数值保持对应即可，不参与比大小；

2) 按 ASCII 码比大小，如 InstanceIds.2 要排在 InstanceIds.12 后面，不是按字母表，也不是按数值。

用户可以借助编程语言中的相关排序函数来实现这一功能，如 PHP 中的 ksort 函数。

上述示例参数的排序结果如下：

```
- { 'Action' : 'BspTextRecognition', 'Nonce' : 11886, 'Region' : 'wh', 'SecretId' :  
'AKIDz8krbsJ5yKBZQp *****', 'Timestamp' : 1465185768, 'Version' : '2019-03-05', }
```

使用其它程序设计语言开发时，可对上面示例中的参数进行排序，得到的结果一致即可。



# 拼接请求字符串

最近更新时间: 2022-10-31 18:51:55

此步骤生成请求字符串。

将把上一步排序好的请求参数格式化成“参数名称=参数值”的形式，如对 Action 参数，其参数名称为 "Action"，参数值为 'BspTextRecognition'，因此格式化后就为 Action='BspTextRecognition'。

注意：“参数值”为原始值而非 url 编码后的值。然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为：

```
https://contentsecurity.api3.yun.ccb.com/?Action=BspTextRecognition&MessageContent=test+info&Nonce=11886&Region=wh&SecretId=AKIDmDJ4SUPSO[REDACTED]&Timestamp=1590062689&Version=2019-03-05&Signature=pKKhkaJ0WBYDt6Q0RVLkKBhy334%3D
```



# 拼接签名原文字符串

最近更新时间: 2022-10-27 10:30:02

此步骤生成签名原文字符串。  
签名原文字符串由以下几个参数构成:

- 1.请求方法: 支持 **POST** 和 **GET** 方式, 这里使用 **GET** 请求, 注意方法为全大写。
- 2.请求主机: 查看实例列表(BspTextRecognitiona)的请求域名为:

区域	域名
标准武汉一区、标准武汉二区	contentsecurity.api3.yun.ccb.com
自用北京一区、自用北京二区	contentsecurity.api3.yun004.ccb.com

实际的请求域名根据接口所属模块的不同而不同, 详见各接口说明。

- 3.请求路径: 当前版本云API的请求路径固定为 / 。
- 4.请求字符串: 即上一步生成的请求字符串。

签名原文串的拼接规则为: 请求方法 + 请求主机 +请求路径 + ? + 请求字符串。  
示例的拼接结果为:

```
- contentsecurity.api3.yun.ccb.com/?
Action=BspTextRecognitiona&Nonce=11886&Offset=0&Region=wh&SecretId=AKIDz8krbsJ5yKBZQp
***** &Timestamp=1465185768&Version=2019-03-05
```





# 生成签名串

最近更新时间: 2022-10-27 10:30:01

此步骤生成签名串。

首先使用 **HMAC-SHA1** 算法对上一步中获得的**签名原文字符串**进行签名，然后将生成的签名串使用**Base64** 进行编码，即可获得最终的签名串。

具体代码如下，以 PHP 语言为例：

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3 *****';  
$srcStr='contentsecurity.api3.yun.ccb.com/?  
BspTextRecognition&Nonce=11886&Offset=0&Region=wh&SecretId=AKIDz8krbsJ5yKBZQp *****  
**** &Timestamp=1465185768&Version=2019-03-05';  
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));  
echo $signStr;
```

最终得到的签名串为：

zmmjn35mikh6pM3V7sUEuX4wyYM=

使用其它程序设计语言开发时，可用上面示例中的原文进行签名验证，得到的签名串与例子中的一致即可。



# 签名串编码

最近更新时间: 2022-10-26 15:20:31

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。

如上一步生成的签名串为zmmjn35mikh6pM3V7sUEuX4wyYM=，最终得到的签名串请求参数Signature为zmmjn35mikh6pM3V7sUEuX4wyYM%3D，它将用于生成最终的请求 URL。注意：

1. 如果用户的请求方法是 GET，或者请求方法为 POST 同时 Content-Type 为 application/x-www-form-urlencoded，则发送请求时所有请求参数的值均需要做 URL 编码，参数键和=符号不需要编码。非 ASCII 字符在 URL 编码前需要先以 UTF-8 进行编码。
2. 有些编程语言的库会自动为所有参数进行 urlencode，在这种情况下，就不需要对签名串进行 URL 编码了，否则两次 URL 编码会导致签名失败。
3. 其他参数值也需要进行编码，编码采用 RFC 3986。使用 %XY 对特殊字符例如汉字进行百分比编码，其中“X”和“Y”为十六进制字符（0-9 和大写字母 A-F），使用小写将引发错误。



# 签名失败

最近更新时间: 2022-10-26 15:20:31

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理。

错误码	错误描述
AuthFailure.SignatureExpire	签名过期。Timestamp 与服务器接收到请求的时间相差不得超过五分钟。
AuthFailure.SecretIdNotFound	密钥不存在。请到控制台查看密钥是否被禁用，是否少复制了字符或者多了字符。
AuthFailure.SignatureFailure	签名错误。可能是签名计算错误，或者签名与实际发送的内容不相符合，也有可能是密钥 SecretKey 错误导致的。
AuthFailure.TokenFailure	临时证书 Token 错误。
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。



# 签名演示

## Python签名演示

最近更新时间: 2022-10-27 10:30:02

注意：如果是在 Python 2 环境中运行，需要先安装 requests 依赖包： `pip install requests` 。 # -- coding: utf-

```
8 -- import base64
```

```
import hashlib
```

```
import hmac
```

```
import time
```

```
import requests
```

```
secret_id = "AKIDmDJ4SUPSOW4R8**" secret_key = "G9Gavsd8F9RfJ5JWf3**"
```

```
def get_string_to_sign(method, endpoint, params):
```

```
    s = method + endpoint + "/"
```

```
    query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
```

```
    return s + query_str
```

```
def sign_str(key, s, method):
```

```
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
```

```
    return base64.b64encode(hmac_str)
```

```
if name == 'main':
```

```
    endpoint = "contentsecurity.api3.yun.ccb.com"
```

```
    data = {
```

```
        'Action': 'BspTextRecognition',
```

```
        'Nonce': 11886,
```

```
        'MessageContent': "测试信息",
```

```
        'Region': 'wh', 'SecretId': secret_id,
```

```
        'Timestamp': int(time.time()), # int(time.time())
```

```
        'Version': '2019-03-05'
```

```
    } s = get_string_to_sign("GET", endpoint, data)
```

```
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
```

```
    print(data["Signature"])
```

```
    # 此处会实际调用，成功后可能产生计费
```

```
    resp = requests.get("https://" + endpoint, params=data)
```

```
    print(resp.url)
```



# 反馈结果 返回结果

最近更新时间: 2022-10-26 15:20:31

目前只要请求被服务端正常处理了，响应的 HTTP 状态码均为200。例如返回的消息体里的错误码是签名失败，但 HTTP 状态码是200，而不是401。



# 正确返回结果

最近更新时间: 2022-10-26 15:20:31

以文本内容安全接口为例，若调用成功，其可能的返回如下为：

```
- { "Response": { "RequestId": "9a69a4fd-f376-4158-be14-307d1a87c210", "Data": {  
  "StatusCode": 0, "Type": 100, "Score": 0, "BeatTips": [] } } }
```

- Response 及其内部的 RequestId 是固定的字段，无论请求成功与否，只要 API 处理了，则必定会返回。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。
- 除了固定的字段外，其余均为具体接口定义的字段，不同的接口所返回的字段参见接口文档中的定义，例如此例中的 Data 为接口定义的字段。



# 错误返回结果

最近更新时间: 2022-10-26 15:20:31

若调用失败，其返回值示例如下为：

```
- { "Response": { "Error": { "Code": "AuthFailure.SignatureFailure", "Message": "The provided credentials could not be validated. Please check your signature is correct." }, "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6" } }
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。



# 错误码

## 公共错误码

最近更新时间: 2022-10-26 15:55:17

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误码	错误描述
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation DryRun	操作，代表请求将会是成功的，只是多传了 DryRun 参数。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。





ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	HTTPS 请求方法错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。



# 业务错误码

最近更新时间: 2022-10-26 15:55:17

InternalError	内部错误
InternalError.Detect	检测服务内部错误。
InvalidParameter	参数错误
InvalidParameter.Appid	Appid异常。
InvalidParameter.AudioContent	音频内容不存在。
InvalidParameter.EndFdate	EndFdate日期格式错误。
InvalidParameter.Error	识别内容不存在。
InvalidParameter.ImageContent	图片内容不存在。
InvalidParameter.MessageContent	识别内容不存在。
InvalidParameter.Pid	Pid 参数错误。
InvalidParameter.ServicesId	Services_Id类型错误。
InvalidParameter.StartFdate	StartFdate日期格式错误。
InvalidParameter.SubType	SubType参数错误。
InvalidParameter.Today	当日日期参数错误。
InvalidParameter.Type	Type 参数错误。
InvalidParameter.UIN	UIN类型错误。
InvalidParameter.WeekFdate	WeekFdate日期格式错误。
InvalidParameter.Yesterday	昨天日期参数错误。
ResourceUnavailable.Error	服务权限未开通。
ResourceUnavailable.NoOpen	服务权限未开通。



# 文本内容安全相关接口

## 接口描述

最近更新时间: 2022-10-26 16:23:47

接口请求域名:

区域	域名
标准武汉一区、标准武汉二区	contentsecurity.api3.yun.ccb.com
自用北京一区、自用北京二区	contentsecurity.api3.yun004.ccb.com

本文档适用于文本内容安全自定义识别库的管理。

支持批量查询文本样本库。

默认接口请求频率限制: 500次/秒。



# 输入参数

最近更新时间: 2022-10-26 16:23:47

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	必选	类型	描述
Action	是	String	公共参数，本接口取值：BspTextRecognition。
Version	是	String	公共参数，本接口取值：2019-03-05。
Region	是	String	公共参数，本接口取值：wh。
MessageContent	是	String	文本内容Base64编码。原文长度需小于15000字节，即5000个汉字以内。



# 输出参数

最近更新时间: 2022-10-26 16:23:47

参数名称	类型	描述
Data	TextData	识别结果
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

## TextData

名称	类型	描述
StatusCode	Integer	公共错误码 0：表示成功 -1003表示图片处理超时；-1004表示文字处理超时；-1005表示内部处理异常；
Type	Integer	恶意类型：100:正常 20001:政治 20002:色情 20006:违法 20007:谩骂 20103:性感 20105 :广告 24001:暴恐
Score	Integer	置信度分值 0-100分值越大置信度越高
BeatTips	Array of BeatTips	打击原因

## BeatTips

名称	类型	描述
Keyword	String	命中关键字
EvilType	Integer	恶意类型，同type说明
Level	Integer	此字段3.6版本后不再使用



# 错误码

最近更新时间: 2022-10-26 16:23:47

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InternalServerError.Detect	检测服务内部错误。
InvalidParameter.Appid	Appid异常。
InvalidParameter.MessageContent	识别内容不存在。
ResourceUnavailable.NoOpen	服务权限未开通。



# 接口请示例

最近更新时间: 2022-10-26 16:25:46

## 输入示例

```
-- https:// contentsecurity.api3.yf-1.tcepoc.fsphere.cn/?Action=BspTextRecognition  
&MessageContent="识别文本" &<公共请求参数>
```

```
-- https://contentsecurity.api3.yun.ccb.com/?  
Action=BspTextRecognition&Nonce=11886&MessageContent=%E6%B5%8B%E8%AF%95%E4%BF  
%A1%E6%81%AF&Region=wh&SecretId=AKIDmDJ4SUPSOzE6clK*****&Timestamp=159  
0064336&Version=2019-03-05&Signature=NwCH%2BYbHMrUv6GX9yVlzcNJ2fHw%3D
```

## 输出示例

```
- { "Response": { "RequestId": "9a69a4fd-f376-4158-be14-307d1a87c210", "Data": {  
  "StatusCode": 0, "Type": 100, "Score": 0, "BeatTips": [] } } }
```



# POST文本请求示例

最近更新时间: 2022-10-31 18:51:54

```
-- coding: utf-8 -- import base64
import hashlib
import hmac
import time
import requests

secret_id = " AKIDmDJ4SUPSOz*****" secret_key = " G9gAl3SUATQ0n *****"

def get_string_to_sign(method, endpoint, params):
    s = method + endpoint + "/"
    query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
    return s + query_str

def sign_str(key, s, method):
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
    return base64.b64encode(hmac_str)

if name == 'main':
    endpoint = "contentsecurity.api3.yun.ccb.com"
    data = {
        'Action': 'BspTextRecognition',
        'Nonce': 11886,
        'MessageContent': "测试信息",
        'Region': 'wh',
        'SecretId': secret_id,
        'Timestamp': int(time.time()), # int(time.time())
        'Version': '2019-03-05'
    } s = get_string_to_sign("POST", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
    print(data["Signature"])
    # 此处会实际调用
    resp = requests.post("https://" + endpoint, data=data)

    print(resp.url)
    print(resp.text)

返回结果如下
```





```
"Response": {"RequestId": "9799da39-3dc8-44b5-9895-a970d224d094", "Data": {"StatusCode": 0, "Type": 100, "BeatTips": []}}
```



# 图片内容安全相关接口

## 接口描述

最近更新时间: 2022-10-26 16:23:47

接口请求域名: contentsecurity.api3.yf-1.tcepoc.fsphere.cn。

识别图片是否存在恶意信息

默认接口请求频率限制: 500次/秒。



# 输入参数

最近更新时间: 2022-10-26 16:59:41

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	必选	类型	描述
Action	是	String	公共参数，本接口取值：BsplImageRecognition
Version	是	String	公共参数，本接口取值：2022-03-05
Region	否	String	公共参数，本接口不需要传递此参数。
FileName	否	String	图片名称
FileUrl	否	String	图片的 URL 地址 与FileContent二选一
FileContent	否	String	对图片内容 Base64 位编码值 与FileUrl二选一



# 输出参数

最近更新时间: 2022-10-26 16:59:41

参数名称	类型	描述
Data	ImageData	识别结果详情
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

## ImageData

名称	类型	描述
StatusCode	Integer	公共错误码 0：表示成功 -1003表示图片处理超时；-1004表示文字处理超时；-1005表示内部处理异常；
Type	Integer	恶意类型：100:正常 20001:政治 20002:色情 20006:违法 20007:谩骂 20103:性感 20105 :广告 24001:暴恐
Data	Array of ImageDetail	图片识别结果详情

## ImageDetail

名称	类型	描述
Category	String	识别类型 PornDetect HotDetect AdvertiseDetect CurseDetect PolityDetect IllegalDetect TerroristDetect OCRDetect SimDetect
HitFlag	Integer	是否恶意：0否1是
Score	Integer	0-100 分值越高越恶意，其中OCRDetect与SimDetect此字段固定为0
Label	String	特征中文描述，其中OCRDetect与SimDetect无此字段
BeatTips	String	打击原因，如命中的关键词(多个以分号分隔)
Level	Integer	此字段3.6版本后不再使用



# 错误码

最近更新时间: 2022-10-26 16:59:41

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter	参数错误
InvalidParameter.Appid	Appid异常。
InvalidParameter.Error	识别内容不存在。
InvalidParameter.ImageContent	图片内容不存在。
ResourceUnavailable.Error	服务权限未开通。



# 接口请求示例

最近更新时间: 2022-10-26 16:59:41

## 输入示例

– [https://contentsecurity.api3.yun.ccb.com/?Action=](https://contentsecurity.api3.yun.ccb.com/?Action=BsplmageRecognition) BsplmageRecognition &FileContent=字符串型 &  
<公共请求参数>

## 输出示例

```
- { "Response": { "RequestId": "6ae9376f-d0c1-4508-9905-01c722eae73b", "Data": {  
  "StatusCode": 0, "Type": 100, "Data": [{ "Category": "PornDetect", "HitFlag": 0, "Score": 0, "Label":  
    "", "BeatTips": "", "Level": "" }, { "Category": "HotDetect", "HitFlag": 0, "Score": 0, "Label": "",  
    "BeatTips": "", "Level": "" }, { "Category": "AdvertiseDetect", "HitFlag": 0, "Score": 0, "Label": "",  
    "BeatTips": "", "Level": "" }, { "Category": "CurseDetect", "HitFlag": 0, "Score": 0, "Label": "",  
    "BeatTips": "", "Level": "" }, { "Category": "PolityDetect", "HitFlag": 0, "Score": 0, "Label": null,  
    "BeatTips": "", "Level": "" }, { "Category": "IllegalDetect", "HitFlag": 0, "Score": 0, "Label": "",  
    "BeatTips": "", "Level": "" }, { "Category": "TerroristDetect", "HitFlag": 0, "Score": 0, "Label": "",  
    "BeatTips": "", "Level": "" }, { "Category": "OCRDetect", "HitFlag": 0, "Score": 0, "Label": "",  
    "BeatTips": "", "Level": "" }, { "Category": "SimDetect", "HitFlag": 0, "Score": 0, "Label": "",  
    "BeatTips": "", "Level": "" } ] } }
```



# POST图片请求示例

最近更新时间: 2022-10-26 16:59:41

```
-- coding: utf-8 -- import base64
import hashlib
import hmac
import time
import os
import requests

secret_id = "AKIDmDJ4SUPSOz****" secret_key = "G9gAI3SUATQ0n****"

def get_string_to_sign(method, endpoint, params):
    s = method + endpoint + "/"
    query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
    return s + query_str

def sign_str(key, s, method):
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
    return base64.b64encode(hmac_str)

def image_to_string(image):
    pic = open(image, 'rb')
    pic_base64 = str(base64.b64encode(pic.read()))
    pic.close()
    return pic_base64

if name == 'main':
    endpoint = "contentsecurity.api3.yun.ccb.com"
    data = {
        'Action': 'BsplImageRecognition',
        'Nonce': 11886,
        'FileContent': image_to_string('simple'+os.sep+"dalai.jpg"),
        'Region': 'wh',
        'SecretId': secret_id,
        'Timestamp': int(time.time()), # int(time.time())
        'Version': '2019-03-05'
    }
    s = get_string_to_sign("POST", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
```



```
print(data["Signature"])  
# 打印图片转换为base64编码后的字符串信息  
print(image_to_string('simple'+os.sep+"dalai.jpg"))  
resp = requests.post("https://" + endpoint, data=data)  
  
print(resp.url)  
print(resp.text)
```





## 常见问题

# 文本内容安全、旅途片内容安全支持识别哪些违规内容？

最近更新时间: 2021-10-09 14:09:37

支持识别色情、涉政、暴恐、广告、毒品以及自定义违规内容。



# 如何接入文本内容安全、图片内容安全？

最近更新时间: 2021-10-09 14:09:37

在文本内容安全控制台 开通服务后，即可通过 API 调用的方式使用服务。



# 文本内容安全、图片内容安全的 API 检测效率如何？

最近更新时间: 2021-10-09 14:09:37

文本检测、图片内容安全通常在1秒以内返回结果，短于200字符的文本，通常在200毫秒内返回结果，200字符以上的内容会根据字符数有所波动。



# 文本内容安全、图片内容安全支持检测哪几种语言？

最近更新时间: 2021-10-09 14:09:37

目前文本内容安全、图片内容安全支持检测中文和英文两种语言，可以检测和识别中文和英文文本中包含的色情、涉政、暴恐、广告、垃圾等信息，同时支持词库配置，打击自定义的违规文本。



# 图片base64转码上传服务器，总是返回-1003？

最近更新时间: 2021-10-09 14:09:37

StatusCode返回-10003分为两种情况。一是服务器确实无法获取目标文件，导致服务器超时。二是服务器获取目标文件后，因编码问题导致无法识别文件，从而返回超时信息。一般遇到-10003状态码，应当使用工具将图片与base64编码互相转换多次无问题后，将base64编码带入代码中进行开发