

# 文本内容安全 实践教学



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

### 实践教程

#### 接入指引

接入准备

API 接入指引

SDK 接入指引

接入后验证

错误码

#### 业务实践

文本 AI 生成识别

第三方送审文本审核

即时通信 IM 云端审核

论坛评论区文本内容审核

#### 权限管理

访问控制和权限管理

# 实践教程

## 接入指引

### 接入准备

最近更新时间：2025-02-05 16:02:22

本文档将介绍如何接入和使用文本内容安全（Text Moderation System，TMS）。

#### 编程访问配置

参考 [访问控制和权限管理](#) 提前配置好 API 权限，并获取腾讯云的 SecretID 和 SecretKey，以便对开发者的身份和权限进行校验。

#### 控制台配置

参考 [快速入门](#)，开通 TMS 服务并领取试用包。根据您的业务需求，配置策略并获取 Biztype，这个字段将在后续测试中使用。

#### 接入方式

支持原生 API 和 SDK 两种接入方式。

原生 API 接入需要用户根据腾讯云的签名指引，构建 Demo 生成签名进行鉴权，目的是为了验证请求者身份以及保护传输中的数据；建议用户使用 SDK 接入，SDK 封装了签名的过程，开发时只需关注产品提供的具体接口即可，接入流程较为便捷。

# API 接入指引

最近更新时间：2024-11-08 11:28:23

腾讯云 API 会对每个请求进行身份验证，用户使用原生API接入需要使用安全凭证，经过特定的步骤对请求进行签名（Signature），每个请求都需要在公共参数中指定该签名结果并以指定的方式和格式发送请求，以下内容主要介绍 API 接口文档参数以及V3签名方法。

## API文档

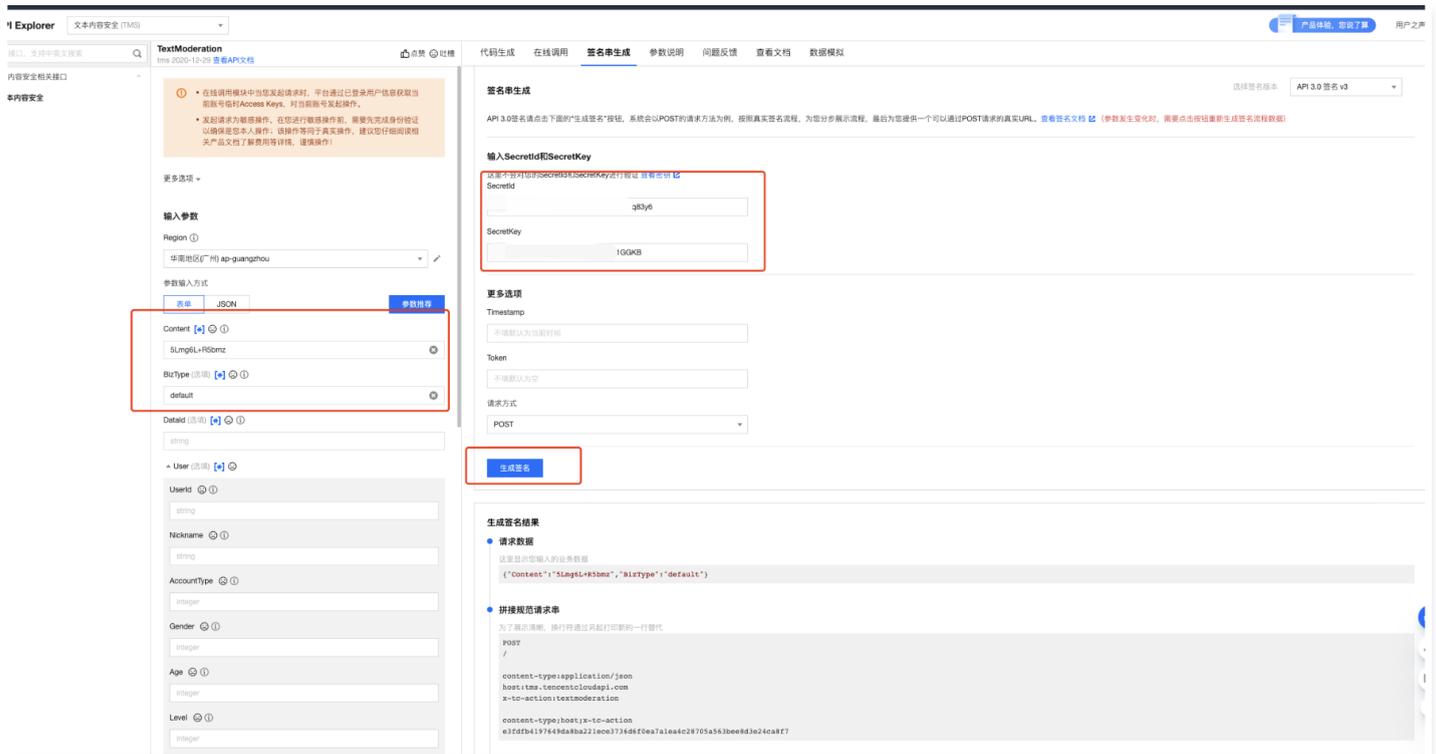
参考 [文本内容安全](#) 接口文档，了解请求API所需要的公共参数、域名、接口输入输出参数值。

## 公共参数

公共参数是用于标识用户和接口签名的参数，API接口文档上不会展示公共参数说明，但每次请求API均需要携带这些参数，才能正常发起请求。

## 签名方法

腾讯云支持V1、V3两种签名方式，推荐使用V3签名方法计算签名，签名方法 v3（有时也称作 TC3-HMAC-SHA256）相比签名方法 v1更安全，支持更大的请求包，支持 POST JSON 格式，性能有一定提升。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 v3”，可以生成签名过程进行验证。



使用签名方法 v3 时，公共参数需要统一放到 HTTP Header 请求头部中，如下表所示：

参数名称	类型	必选	描述
Action	String	是	HTTP 请求头：X-TC-Action。操作的接口名称，该字段取值为 TextModeration。
Region	String	-	HTTP 请求头：X-TC-Region。地域参数，用来标识希望操作哪个地域的数据。取值参考接口文档中输入参数章节关于公共参数 Region 的说明。 <b>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。</b>
Timestamp	Integer	是	HTTP 请求头：X-TC-Timestamp。当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702。 <b>注意：如果与服务器时间相差超过5分钟，会引起签名过期错误。</b>
Version	String	是	HTTP 请求头：X-TC-Version。操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。该字段取值为2020-12-29。

Authorization	String	是	<p>HTTP 标准身份认证头部字段，例如：</p> <pre>TC3-HMAC-SHA256 Credential =AKID****/Date/service/tc3_request, SignedHeaders =content-type ;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024</pre> <p>其中，</p> <ul style="list-style-type: none"> <li>• TC3-HMAC-SHA256：签名方法，目前固定取该值。</li> <li>• Credential：签名凭证，AKID**** 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为具体产品名，通常为域名前缀，例如域名 cvm.tencentcloudapi.com 意味着产品名是 cvm。本产品取值为 tms；tc3_request 为固定字符串。</li> <li>• SignedHeaders：参与签名计算的头部信息，content-type 和 host 为必选头部。</li> <li>• Signature：签名摘要，计算过程请参见 <a href="#">签名版本 v3 签名过程</a>。</li> </ul>
Token	String	否	<p>HTTP 请求头：X-TC-Token。即 <a href="#">安全凭证服务</a> 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。</p>
Language	String	否	<p>HTTP 请求头：X-TC-Language。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。</p>

云 API 支持 GET 和 POST 请求。

- 对于 GET 方法，只支持 Content-Type: application/x-www-form-urlencoded 协议格式。
- 对于 POST 方法，目前支持 Content-Type: application/json 协议格式。

推荐使用 POST 请求，因为两者的结果并无差异，但 GET 请求只支持 32 KB 以内的请求包，POST 请求支持 10MB 以内的请求包，签名过程详见 [文档](#)。

### 示例代码

以下提供 Java、Go、Python 示例 demo，填写密钥信息、接口入参即可调用。其他语言请参考 [签名方法 V3-签名演示](#)，签名演示是云服务器服务作为示例，实际调用时需要根据 API 接口文档填写参数。

```
Java

import java.nio.charset.Charset;
import java.io.OutputStream;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TextModeration {
    private final static Charset UTF8 = StandardCharsets.UTF_8;
    private final static String SECRET_ID = "";
    private final static String SECRET_KEY = "";
```

```
private final static String CT_JSON = "application/json; charset=utf-8";

public static byte[] hmac256(byte[] key, String msg) throws Exception {
    Mac mac = Mac.getInstance("HmacSHA256");
    SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
    mac.init(secretKeySpec);
    return mac.doFinal(msg.getBytes(UTF8));
}

public static String sha256Hex(String s) throws Exception {
    MessageDigest md = MessageDigest.getInstance("SHA-256");
    byte[] d = md.digest(s.getBytes(UTF8));
    return DatatypeConverter.printHexBinary(d).toLowerCase();
}

public static void main(String[] args) throws Exception {
    String service = "tms";
    String host = "tms.tencentcloudapi.com";
    String region = "ap-guangzhou";
    String action = "TextModeration";
    String version = "2020-12-29";
    String algorithm = "TC3-HMAC-SHA256";
    String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
    String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

    String httpRequestMethod = "POST";
    String canonicalUri = "/";
    String canonicalQueryString = "";
    String canonicalHeaders = "content-type:application/json; charset=utf-8\n"
        + "host:" + host + "\n" + "x-tc-action:" + action.toLowerCase() + "\n";
    String signedHeaders = "content-type;host;x-tc-action";

    String payload = "{\"Content\": \"xxxx\", \"BizType\": \"xxxx\"}"; // body数据为json
    String hashedRequestPayload = sha256Hex(payload);
    String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryString
        + "\n"
        + canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;

    String credentialScope = date + "/" + service + "/" + "tc3_request";
    String hashedCanonicalRequest = sha256Hex(canonicalRequest);
    String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" +
        hashedCanonicalRequest;

    byte[] secretDate = hmac256(("TC3" + SECRET_KEY).getBytes(UTF8), date);
    byte[] secretService = hmac256(secretDate, service);
    byte[] secretSigning = hmac256(secretService, "tc3_request");
    String signature = DatatypeConverter.printHexBinary(hmac256(secretSigning,
        stringToSign)).toLowerCase();
}
```

```
String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ",  
"  
    + "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;  
  
try {  
    URL url = new URL("https://" + host);  
    HttpURLConnection connection = (HttpURLConnection) url.openConnection();  
    connection.setRequestMethod("POST");  
    connection.setRequestProperty("Authorization", authorization);  
    connection.setRequestProperty("Content-Type", CT_JSON);  
    connection.setRequestProperty("Host", host);  
    connection.setRequestProperty("X-TC-Action", action);  
    connection.setRequestProperty("X-TC-Timestamp", timestamp);  
    connection.setRequestProperty("X-TC-Version", version);  
    connection.setRequestProperty("X-TC-Region", region);  
  
    // Enable input/output streams and write the payload  
    connection.setDoOutput(true);  
    OutputStream os = connection.getOutputStream();  
    os.write(payload.getBytes(UTF8));  
    os.flush();  
    os.close();  
  
    // Get the response  
    int responseCode = connection.getResponseCode();  
    if (responseCode == 200) {  
        BufferedReader in = new BufferedReader(new  
InputStreamReader(connection.getInputStream()));  
        String inputLine;  
        StringBuilder response = new StringBuilder();  
  
        while ((inputLine = in.readLine()) != null) {  
            response.append(inputLine);  
        }  
        in.close();  
  
        // Process the response as needed  
        System.out.println("Response: " + response.toString());  
    } else {  
        // Handle the error response  
        System.out.println("Error Response Code: " + responseCode);  
    }  
  
    connection.disconnect();  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

Go

```
package tms

import (
    "crypto/hmac"
    "crypto/sha256"
    "encoding/base64"
    "encoding/hex"
    "encoding/json"
    "fmt"
    "io/ioutil"
    "net/http"
    "strconv"
    "strings"
    "time"
)

type textmods struct {
    Content string
}

func sha256hexs(s string) string {
    b := sha256.Sum256([]byte(s))
    return hex.EncodeToString(b[:])
}

func hmacsha256s(s, key string) string {
    hashed := hmac.New(sha256.New, []byte(key))
    hashed.Write([]byte(s))
    return string(hashed.Sum(nil))
}

func DetectTextSingles() {
    secretId := "" // 密钥ID
    secretKey := "" // 密钥Key
    host := "tms.tencentcloudapi.com" // 接口域名
    algorithm := "TC3-HMAC-SHA256"
    service := "tms"
    version := "2020-12-29" // 版本为固定值
    action := "TextModeration"
    region := "ap-guangzhou" // 地域
    var timestamp int64 = time.Now().Unix()

    // step 1: build canonical request string
    httpRequestMethod := "POST"
    canonicalURI := "/"
    canonicalQueryString := ""
    canonicalHeaders := "content-type:application/json; charset=utf-8\n" + "host:" + host + "\n"
    signedHeaders := "content-type;host"
    var a = textmods{
        Content: base64.StdEncoding.EncodeToString([]byte("测试文本内容")), // 接口入参
    }
    b, err := json.Marshal(a)
    if err != nil {
        fmt.Print(err.Error())
        return
    }
    payload := string(b)
    hashedRequestPayload := sha256hexs(payload)
    canonicalRequest := fmt.Sprintf("%s\n%s\n%s\n%s\n%s\n%s",
```

```
    httpRequestMethod,
    canonicalURI,
    canonicalQueryString,
    canonicalHeaders,
    signedHeaders,
    hashedRequestPayload)
// fmt.Println(canonicalRequest)

// step 2: build string to sign
date := time.Unix(timestamp, 0).UTC().Format("2006-01-02")
credentialScope := fmt.Sprintf("%s/%s/tc3_request", date, service)
hashedCanonicalRequest := sha256hexs(canonicalRequest)
string2sign := fmt.Sprintf("%s\n%d\n%s\n%s",
    algorithm,
    timestamp,
    credentialScope,
    hashedCanonicalRequest)
// fmt.Println(string2sign)

// step 3: sign string
secretDate := hmacsha256s(date, "TC3"+secretKey)
secretService := hmacsha256s(service, secretDate)
secretSigning := hmacsha256s("tc3_request", secretService)
signature := hex.EncodeToString([]byte(hmacsha256s(string2sign, secretSigning)))
// fmt.Println(signature)

// step 4: build authorization
authorization := fmt.Sprintf("%s Credential=%s/%s, SignedHeaders=%s, Signature=%s",
    algorithm,
    secretId,
    credentialScope,
    signedHeaders,
    signature)

req, _ := http.NewRequest("POST", "https://"+host, strings.NewReader(payload))
req.Header.Set("Authorization", authorization)
req.Header.Set("Content-Type", "application/json; charset=utf-8")
req.Header.Set("Host", host)
req.Header.Set("X-TC-Action", action)
req.Header.Set("X-TC-Timestamp", strconv.FormatInt(timestamp, 10))
req.Header.Set("X-TC-Version", version)
req.Header.Set("X-TC-Region", region)

resp, err := (&http.Client{}).Do(req)
if err != nil {
    fmt.Print(err.Error())
    return
}
defer resp.Body.Close()

respByte, _ := ioutil.ReadAll(resp.Body)
fmt.Println(string(respByte))
}
```

## Python

```
#!/usr/bin/env python3
# -*- coding:utf-8 -*-
```

```
import hashlib
import hmac
import json
import base64
import time
from datetime import datetime
import requests
from concurrent.futures import ThreadPoolExecutor

secret_id = "" # 密钥ID
secret_key = "" # 密钥Key

service = "tms"
host = "tms.tencentcloudapi.com" # 接口域名
endpoint = "https://" + host
region = 'ap-guangzhou' # 地域
version = "2020-12-29" # 版本为固定值
algorithm = "TC3-HMAC-SHA256"

def do_action(action, params):
    timestamp = int(time.time())
    day = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")

    # ***** 步骤 1: 拼接规范请求串 *****
    http_request_method = "POST"
    canonical_url = "/"
    canonical_querystring = ""
    ct = "application/json; charset=utf-8"
    payload = json.dumps(params)
    canonical_headers = "content-type:%s\nhost:%s\n" % (ct, host)
    signed_headers = "content-type;host"
    hashed_request_payload = hashlib.sha256(
        payload.encode("utf-8")).hexdigest()
    canonical_request = (http_request_method + "\n" +
        canonical_url + "\n" +
        canonical_querystring + "\n" +
        canonical_headers + "\n" +
        signed_headers + "\n" +
        hashed_request_payload)
    # print(canonical_request)

    # ***** 步骤 2: 拼接待签名字符串 *****
    credential_scope = day + "/" + service + "/" + "tc3_request"
    hashed_canonical_request = hashlib.sha256(
        canonical_request.encode("utf-8")).hexdigest()
    string_to_sign = (algorithm + "\n" +
        str(timestamp) + "\n" +
        credential_scope + "\n" +
        hashed_canonical_request)
    # print(string_to_sign)

    secret_date = sign(("TC3" + secret_key).encode("utf-8"), day)
    secret_service = sign(secret_date, service)
    secret_signing = sign(secret_service, "tc3_request")
    signature = hmac.new(secret_signing, string_to_sign.encode(
```

```
    "utf-8"), hashlib.sha256).hexdigest()
# print(signature)

# ***** 步骤 4: 拼接 Authorization *****
authorization = (algorithm + " " +
                "Credential=" + secret_id + "/" + credential_scope + ", " +
                "SignedHeaders=" + signed_headers + ", " +
                "Signature=" + signature)
# print(authorization)

headers = {
    "Authorization": authorization,
    "Content-Type": "application/json; charset=utf-8",
    "Host": host,
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region
}

# 发送请求
resp = requests.post(url=endpoint, data=payload, headers=headers)
return resp

# 计算签名摘要函数
def sign(key, msg):
    return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()

def image_detect(params):
    action = "TextModeration"
    resp = do_action(action, params)

    return resp.text

if __name__ == "__main__":

    params = []
    content = "" # 接口参数
    b4content = base64.b64encode(content.encode()).decode()
    p = {
        'BizType': 'default',
        'Content': b4content
    }
    params.append(p)
    executor = ThreadPoolExecutor(max_workers=1)
    resps = executor.map(image_detect, params)
    for resp in resps:
        print(resp)
```

## PHP

```
<?php
//简单的后端demo
// 接口入参
```

```
$Content='';
$BizType='';

$time=time();
$Nonce=rand();
$Action='TextModeration'; // 接口名称
$Version='2020-12-29';
$Region='ap-guangzhou';

//根据实际需求填写 SecretId 和 secretKey
$SecretId='';
$secretKey='';

//验签过程
$params["Nonce"] = $Nonce;
$params["Timestamp"] = $time;
$params["Region"] = $Region;
$params["SecretId"] = $SecretId;
$params["Action"] = $Action;
$params["Version"] = $Version;
$params["Content"] = $Content;
$params["BizType"] = $BizType;
ksort($params);
$signStr = "POSTtms.tencentcloudapi.com/?";
foreach ($params as $key => $value) {
    $signStr = $signStr . $key . "=" . $value . "&";
}
$signStr = substr($signStr, 0, -1);
$signature = base64_encode(hash_hmac("sha1", $signStr,$secretKey, true));
//$signature=urlencode($signature); //文档要求必须先urlencode

//POST数据
$post_data = array(
    'Action' => $Action,
    'Nonce' =>$Nonce,
    'SecretId'=>$SecretId,
    'Signature'=>$signature, //签名
    'Content' =>$Content,
    'BizType' =>$BizType,
    'Region' =>$Region,
    'Timestamp' =>$time,
    'Version' =>$Version,
);

//POST请求
$url='https://tms.tencentcloudapi.com/';
$query = http_build_query($post_data);
$options = array(
    'http' => array(
        'method' => 'POST',
        'header' => 'Content-type:application/x-www-form-urlencoded',
        'content' => $query,
        'timeout' => 15 * 60 // 超时时间(单位:s)
    )
);
$content = stream_context_create($options);
$result = file_get_contents($url,false,$content);

$data_arr['code']=1;
```

```
$data_arr['result']=$result;
$data_arr['signature']=$signature;

header('content-type:application/json');
echo json_encode($data_arr);
exit();
```

## 签名失败

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 <a href="#">控制台</a> 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）。

# SDK 接入指引

最近更新时间：2025-02-05 16:02:22

推荐您使用腾讯云 API 配套的 7 种常见的编程语言 SDK，已经封装了签名和请求过程，开发时只需关注产品提供的具体接口即可。

## API文档

参考 [文本内容安全](#) 接口文档，了解需要使用的参数。

## SDK使用说明

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，均已开源，能更方便的调用 API，支持 [Python](#)、[Java](#)、[PHP](#)、[Go](#)、[NodeJS](#)、[.NET](#)、[C++](#)。

各语言SDK使用说明可参考 [文档](#)，介绍依赖版本以及安装过程等。



The screenshot shows the 'SDK 中心' (SDK Center) page. On the left is a navigation menu with links for 'SDK 简介 3.0', 'PHP 3.0', 'Python 3.0', 'Java 3.0', 'Go 3.0', '.NET 3.0', 'Node.js 3.0', 'C++ 3.0', 'Ruby 3.0', and '联系我们 3.0'. The 'Python 3.0' link is highlighted with a red box. The main content area shows the 'Python' SDK page with the breadcrumb '文档中心 > SDK 中心 > Python'. The page title is 'Python' and the update time is '2023-01-11 20:20:38'. The '简介' (Introduction) section contains three bullet points: 1. Welcome to use the Tencent Cloud Developer Tool Kit (SDK) 3.0, which is a unified tool for the Cloud API 3.0 platform. 2. This article uses Python SDK 3.0 as an example to introduce how to use, debug, and integrate with Tencent Cloud APIs. 3. Currently supports Tencent Cloud products like CVM, VPC, and CBS, with more products to be supported in the future. The '依赖环境' (Dependencies) section lists: 1. Python 2.7, 3.6 to 3.9. 2. How to obtain a security credential (SecretId and SecretKey) from the 'API 密钥管理' page.

建议安装最新版本的SDK，即可调用接口，腾讯云提供了 [API Explorer](#) 可在线调用，并且可以快速生成 SDK 调用示例代码，以下Java、python、Golang示例均从 [API Explorer](#) 生成，其他语言可自行调试获取。

## 在线调用示例

### 代码生成示例

**Java**

```

import com.tencentcloudapi.common.Credential;
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;
import com.tencentcloudapi.common.exception.TencentCloudSDKException;
import com.tencentcloudapi.tms.v20201229.TmsClient;
import com.tencentcloudapi.tms.v20201229.models.*;

public class TextModeration
{
    public static void main(String [] args) {
        try{
            // 实例化一个认证对象，入参需要传入腾讯云账户 SecretId 和 SecretKey，此处还请注意密钥对的保密
            // 代码泄露可能会导致 SecretId 和 SecretKey 泄露，并威胁账号下所有资源的安全性。以下代码示例仅供参考，建
            // 议采用更安全的方式来使用密钥，请参见：https://cloud.tencent.com/document/product/1278/85305
            // 密钥可前往官网控制台 https://console.cloud.tencent.com/cam/capi 进行获取
            Credential cred = new Credential("SecretId", "SecretKey");

```

```
// 实例化一个http选项, 可选的, 没有特殊需求可以跳过
HttpProfile httpProfile = new HttpProfile();
httpProfile.setEndpoint("tms.tencentcloudapi.com");
// 实例化一个client选项, 可选的, 没有特殊需求可以跳过
ClientProfile clientProfile = new ClientProfile();
clientProfile.setHttpProfile(httpProfile);
// 实例化要请求产品的client对象, clientProfile是可选的
TmsClient client = new TmsClient(cred, "ap-guangzhou", clientProfile);
// 实例化一个请求对象, 每个接口都会对应一个request对象
TextModerationRequest req = new TextModerationRequest();
req.setContent("5Yqg5b6u5L+h");
req.setBizType("default");
// 返回的resp是一个TextModerationResponse的实例, 与请求对象对应
TextModerationResponse resp = client.TextModeration(req);
// 输出json格式的字符串回包
System.out.println(TextModerationResponse.toJsonString(resp));
} catch (TencentCloudSDKException e) {
    System.out.println(e.toString());
}
}
```

## Go

```
package main

import (
    "fmt"

    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/errors"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile"
    tms "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/tms/v20201229"
)

func main() {
    // 实例化一个认证对象, 入参需要传入腾讯云账户 SecretId 和 SecretKey, 此处还请注意密钥对的保密
    // 代码泄露可能会导致 SecretId 和 SecretKey 泄露, 并威胁账号下所有资源的安全性。以下代码示例仅供参考, 建议采用
    // 更安全的方式来使用密钥, 请参见: https://cloud.tencent.com/document/product/1278/85305
    // 密钥可前往官网控制台 https://console.cloud.tencent.com/cam/capi 进行获取
    credential := common.NewCredential(
        "SecretId",
        "SecretKey",
    )
    // 实例化一个client选项, 可选的, 没有特殊需求可以跳过
    cpf := profile.NewClientProfile()
    cpf.HttpProfile.Endpoint = "tms.tencentcloudapi.com"
    // 实例化要请求产品的client对象, clientProfile是可选的
    client, _ := tms.NewClient(credential, "ap-guangzhou", cpf)

    // 实例化一个请求对象, 每个接口都会对应一个request对象
    request := tms.NewTextModerationRequest()

    request.Content = common.StringPtr("5Yqg5b6u5L+h")
    request.BizType = common.StringPtr("default")

    // 返回的resp是一个TextModerationResponse的实例, 与请求对象对应
    response, err := client.TextModeration(request)
```

```
if _, ok := err.(*errors.TencentCloudSDKError); ok {
    fmt.Printf("An API error has returned: %s", err)
    return
}
if err != nil {
    panic(err)
}
// 输出json格式的字符串回包
fmt.Printf("%s", response.ToJsonString())
}
```

## Python

```
import json
from tencentcloud.common import credential
from tencentcloud.common.profile.client_profile import ClientProfile
from tencentcloud.common.profile.http_profile import HttpProfile
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudSDKException
from tencentcloud.tms.v20201229 import tms_client, models

try:
    # 实例化一个认证对象，入参需要传入腾讯云账户 SecretId 和 SecretKey，此处还请注意密钥对的保密
    # 代码泄露可能会导致 SecretId 和 SecretKey 泄露，并威胁账号下所有资源的安全性。以下代码示例仅供参考，建议采用更安全
    # 的方式来使用密钥，请参见：https://cloud.tencent.com/document/product/1278/85305
    # 密钥可前往官网控制台 https://console.cloud.tencent.com/cam/capi 进行获取
    cred = credential.Credential("SecretId", "SecretKey")
    # 实例化一个http选项，可选的，没有特殊需求可以跳过
    httpProfile = HttpProfile()
    httpProfile.endpoint = "tms.tencentcloudapi.com"

    # 实例化一个client选项，可选的，没有特殊需求可以跳过
    clientProfile = ClientProfile()
    clientProfile.httpProfile = httpProfile
    # 实例化要请求产品的client对象，clientProfile是可选的
    client = tms_client.TmsClient(cred, "ap-guangzhou", clientProfile)

    # 实例化一个请求对象，每个接口都会对应一个request对象
    req = models.TextModerationRequest()
    params = {
        "Content": "5Yqg5b6u5L+h",
        "BizType": "default"
    }
    req.from_json_string(json.dumps(params))

    # 返回的resp是一个TextModerationResponse的实例，与请求对象对应
    resp = client.TextModeration(req)
    # 输出json格式的字符串回包
    print(resp.to_json_string())

except TencentCloudSDKException as err:
    print(err)
```

## 热点问题

### SDK 是否支持设置超时时间？

SDK 具有默认的超时时间，除非必要，请勿更改默认设置。如果需要修改，可以 参考 [SDK仓库](#) 的详细版示例。

## 详细版

```
import com.tencentcloudapi.common.Credential;
import com.tencentcloudapi.common.exception.TencentCloudSDKException;
// 导入对应产品模块的client
import com.tencentcloudapi.cvm.v20170312.CvmClient;
// 导入要请求接口对应的request response类
import com.tencentcloudapi.cvm.v20170312.models.DescribeInstancesRequest;
import com.tencentcloudapi.cvm.v20170312.models.DescribeInstancesResponse;
import com.tencentcloudapi.cvm.v20170312.models.Filter;
//导入可选配置类
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;
import com.tencentcloudapi.common.profile.Language;

public class DescribeInstances {
    public static void main(String[] args) {
        try {
            // 实例化一个认证对象，入参需要传入腾讯云账户 SecretId, SecretKey。
            // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中，请参考本文凭证管理章节。
            // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
            // Credential cred = new Credential("SecretId", "SecretKey");
            Credential cred = new Credential(System.getenv("TENCENTCLOUD_SECRET_ID"), System.getenv("TENCENTCLOUD_SECRET_KEY"));

            // 实例化一个http选项，可选的，没有特殊需求可以跳过
            HttpProfile httpProfile = new HttpProfile();
            // 从3.0.96版本开始，单独设置 HTTP 代理
            // httpProfile.setProxyHost("真实代理ip");
            // httpProfile.setProxyPort("真实代理端口");
            httpProfile.setReqMethod("GET"); // get请求(默认为post请求)
            httpProfile.setConnTimeout(30); // 请求连接超时时间，单位为秒(默认60秒)
            httpProfile.setWriteTimeout(30); // 设置写入超时时间，单位为秒(默认0秒)
            httpProfile.setReadTimeout(30); // 设置读取超时时间，单位为秒(默认0秒)
            httpProfile.setEndpoint("cvm.ap-shanghai.tencentcloudapi.com"); // 指定接入地域域名(默认就近)
```

### 多线程是否可以共用一个 Client?

SDK 目前不支持线程安全，因此需要在每个线程中创建一个 Client 来进行请求。

### 同时连接数量是否有限制?

目前没有限制，但是接口会限制每秒请求数（QPS）的并发数，文本接口默认为100QPS。建议客户端也控制并发数，因为太多线程可能超出用户机器的处理能力，导致连接失败。

# 接入后验证

最近更新时间：2025-02-05 16:02:22

当您通过上述步骤完成 API 接入后，可以通过如下方式验证文本内容检测服务是否接入成功。

## 接入成功

若 Response 回参没有“Error”字段且正常返回业务参数，则说明 TMS 接口接入成功，示例如下所示：

```
{
  "Response": {
    "DataId": "123",
    "Extra": "xx",
    "BizType": "0",
    "RiskDetails": [
      {
        "Level": 2,
        "Label": "RiskAccount"
      }
    ],
    "DetailResults": [
      {
        "LibName": "Porn",
        "Score": 72,
        "Label": "Porn",
        "LibId": "12",
        "Suggestion": "Review",
        "Keywords": [
          "色情"
        ],
        "LibType": 0
      },
      {
        "LibName": "Porn",
        "Score": 0,
        "Label": "",
        "LibId": "1",
        "Suggestion": "Block",
        "Keywords": [
          "色情"
        ],
        "LibType": 2
      }
    ],
    "Label": "Ad",
    "Score": 87,
    "RequestId": "x2123-123123-123",
    "Suggestion": "Block",
    "Keywords": [
      "加我好友，给你发优惠券"
    ]
  }
}
```

## 接入失败

若 Response 回参显示“Error”字段，则说明接入失败，以下为 signature 校验失败的示例：

```
{
  "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"
  }
}
```

其中 Error 中的 Code 表示错误码，Message 表示错误的具体信息，详情请参见 [错误码](#)。

如您仍无法定位接入错误的原因，可以 [提交工单](#) 联系技术工程师，我们会7\*24小时响应和解决您的问题。

# 错误码

最近更新时间：2025-02-05 16:02:22

## 公共错误码

错误码	说明
ActionOffline	接口已下线，停止服务
AuthFailure.InvalidAuthorization	请求头部的 Authorization 不符合腾讯云标准
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）
AuthFailure.MFAFailure	<a href="#">MFA 错误</a>
AuthFailure.SecretIdNotFound	密钥不存在。请在控制台检查密钥是否已被删除或者禁用，如状态正常，请检查密钥是否填写正确，注意前后不得有空格
AuthFailure.SignatureExpire	签名过期。Timestamp 和服务器时间相差不得超过五分钟，请检查本地时间是否和标准时间同步
AuthFailure.SignatureFailure	签名错误。签名计算错误，请对照调用方式中的签名方法文档检查签名计算过程
AuthFailure.TokenFailure	token 错误
AuthFailure.UnauthorizedOperation	请求未授权。请参考 <a href="#">CAM 文档</a> 对鉴权的说明
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（包括参数格式、类型等错误）
InvalidParameterValue	参数取值错误
InvalidRequest	请求 body 的 multipart 格式错误
IpInBlacklist	IP 地址在黑名单中
IpNotInWhitelist	IP 地址不在白名单中
LimitExceeded	超过配额限制
MissingParameter	缺少参数
NoSuchProduct	产品不存在
NoSuchVersion	接口版本不存在
RequestLimitExceeded	请求的次数超过了频率限制
RequestLimitExceeded.IPLimitExceeded	IP 限频
RequestLimitExceeded.UinLimitExceeded	主账号限频
RequestSizeLimitExceeded	请求包超过限制大小
ResourceInUse	资源被占用
ResourceInsufficient	资源不足

ResourceNotFound	资源不存在
ResourceUnavailable	资源不可用
ResponseSizeLimitExceeded	返回包超过限制大小
ServiceUnavailable	当前服务暂时不可用
UnauthorizedOperation	未授权操作
UnknownParameter	未知参数错误，用户多传未定义的参数会导致错误
UnsupportedOperation	操作不支持
UnsupportedProtocol	http(s) 请求协议错误，只支持 GET 和 POST 请求
UnsupportedRegion	接口不支持所传地域

## 业务错误码

错误码	说明
InternalServerError.ErrTextTimeOut	请求超时
InvalidParameter.ErrAction	错误的 action
InvalidParameter.ErrTextContentLen	请求的文本长度过长
InvalidParameter.ErrTextContentType	文本类型错误，需要 base64 的文本
InvalidParameterValue.ErrTextContentLen	请求的文本长度超过限制
InvalidParameterValue.ErrTextContentType	请求的文本格式错误，需要 base64 编码格式的文本
UnauthorizedOperation.Unauthorized	未获取到接口授权

# 业务实践

## 文本 AI 生成识别

最近更新时间：2025-06-03 17:49:22

### 服务概述

文本内容安全提供了对 AI 生成内容的识别能力，本文将为您介绍**页面体验**和**API 调用**两种文本生成识别服务的使用方式。

#### 快捷入口：

- 页面体验：[内容安全控制台-生成识别](#)
- API 调用：[文本内容安全接入文档](#)

### 能力介绍

大模型技术的快速发展，给内容创作领域带来了更多的可能性，但同时也降低了创作门槛，带来了虚假信息、生成非法内容等问题。本产品通过文本在形式（如标点、特殊字符、Markdown 格式）和内容（如泛泛而谈、正确而无用的话）这些方面所呈现出的特征，鉴别文本是否为 AI 工具生成，覆盖 chatgpt4o、doubao、qwene、deepseek-r1 等数十种主流文本生成大模型，针对新闻、百科、问答、评论、小说内容主题，在实际应用中识别准确率较高。

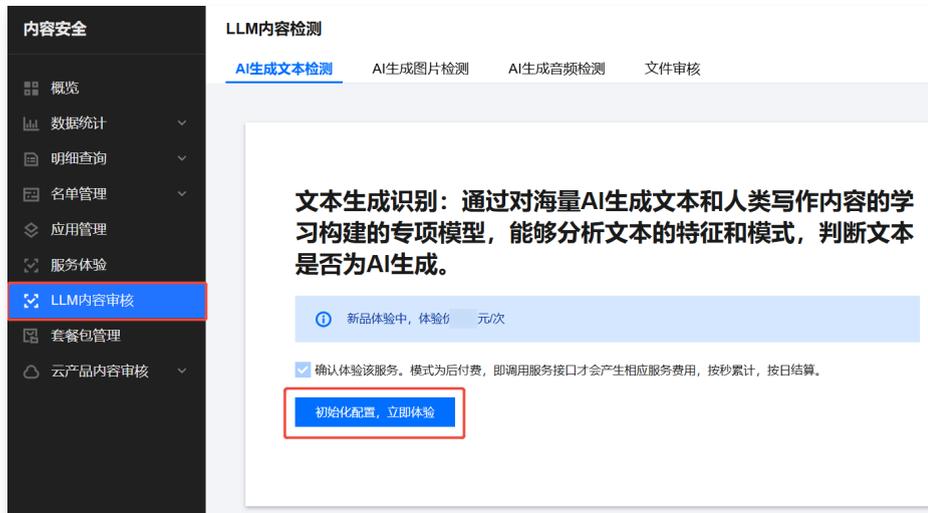
### 服务对接

#### 说明：

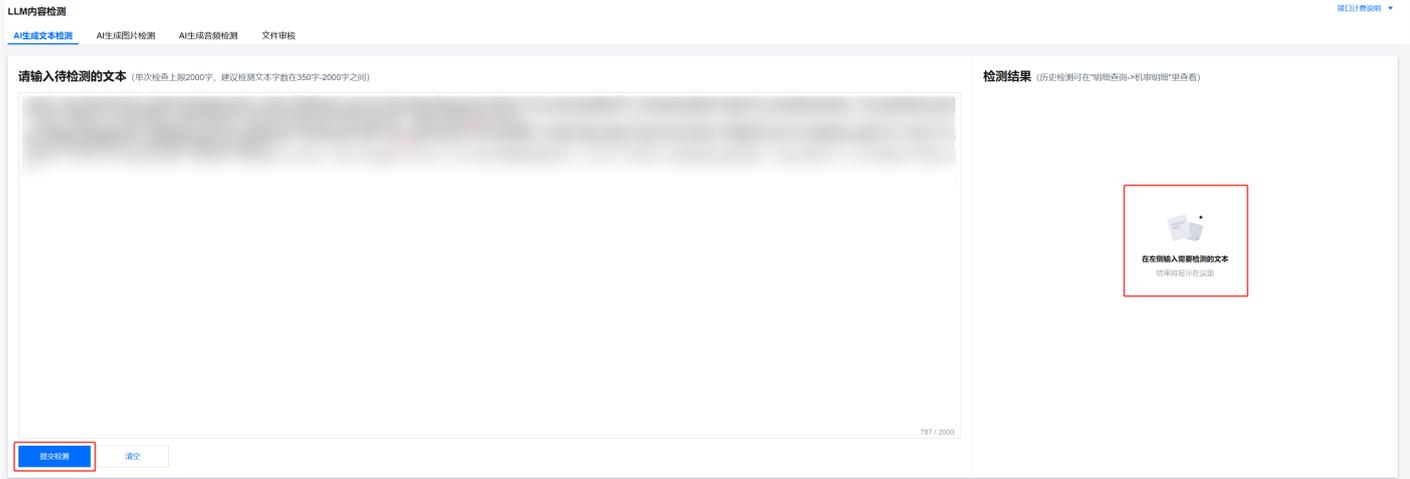
本服务的使用，需要您先登录腾讯云账号。若您未注册腾讯云账号，可参考 [账号注册教程](#) 先行注册。

### 方式一：通过控制台页面使用

1. 登录 [内容安全控制台](#)，单击 LLM 内容审核，跳转后单击**初始化配置**，立即体验。



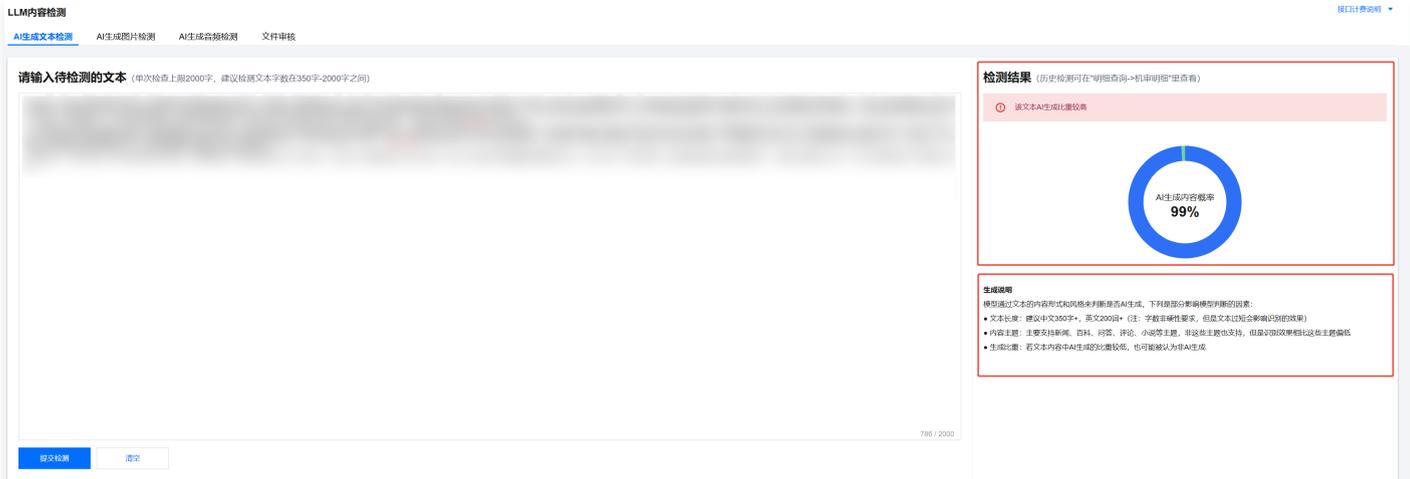
2. 在 AI 生成文本检测页面的指定区域中，域输入**待检测文本**。



**说明：**

AI 文本生成识别单次检查上限2000字，同时过短的文字也难以证明是否为 AI 生成，因此建议检测文本字数在350字-2000字之间。

3. 输入待检测文本后，单击**提交检测**，即可查看**检测结果与生成说明**。



**文本 AI 生成识别示例（您可复制这段文本，作为输入内容快速体验产品效果）：**

去年春天，我在社区医院偶然目睹了人工智能辅助诊断系统的首次试运行。记得那个下着细雨的午后，候诊厅的电子屏突然切换成蓝底白字的"AI读片系统上线"提示。放射科张主任戴着老花镜，手指在触控屏上迟疑滑动时的微妙表情，至今仍清晰印在我脑海里——那是一种混合着期待与不安的复杂神情。隔壁诊室传来实习生们的窃窃私语："听说这套系统在测试时把3%的陈旧性骨折误判成恶性肿瘤..."

这个场景让我想起大学导师常说的"技术驯化论"。任何新技术的落地都必然经历与人类工作模式的痛苦磨合期。就像去年市立图书馆引进的图书分拣机器人，最初两周总把哲学著作塞进少儿绘本区。

人工智能在医疗领域的应用正以前所未有的速度发展。基于深度学习的医学影像分析系统通过数百万张标注影像的训练，已能实现CT扫描94.7%的结节识别准确率。自然语言处理技术的进步使得电子病历自动生成系统可完整提取医患对话中的35个关键临床指标。值得关注的是，联邦学习框架的引入有效解决了医疗数据隐私问题，使跨机构模型训练成为可能。

在药物研发领域，生成式AI展现出巨大潜力。某程序成功预测了98.5%的人类蛋白质结构，极大缩短了新靶点发现周期。某公司2023年报告显示，其AI辅助设计的疫苗研发效率提升42%。与此同时，手术机器人市场年复合增长率达23.6%，某系统已完成超过千万例微创手术。

上周回访时，张主任的工作台已堆满各色标注便签。他指着屏幕上闪烁的置信度百分比苦笑："现在每个诊断都要和这个数字较劲。"窗台上新添的绿萝藤蔓缠绕着数据线生长，恰好构成个天然理线器。这颇具隐喻意味的画面提醒我们：当算法试图理解人体时，或许也需要保留些"不精确"的温柔。

**方式二：通过 API 接口调用**

若您需要和您的业务系统做对接，则可使用 API 接口的方式来调用我们的服务。

1. 开通服务：登录 [内容安全控制台](#)，单击 LLM 内容审核，跳转后单击初始化配置，立即体验。

2. 获取接口参数：单击“应用管理”，即可见如下图所示的“AI生成检测配套策略”，其中 Biztype 字段对应的值，即接口调用的必备传值。
3. 系统对接：请参见 [文本内容安全接入文档](#) 来完成系统接口的对接。其中，接口参数中 Biztype 的传值，即为上一步您在“应用管理”页面查看的 Biztype 值，Type 的传值为 TEXT\_AIGC。



# 第三方送审文本审核

最近更新时间：2025-02-05 16:02:22

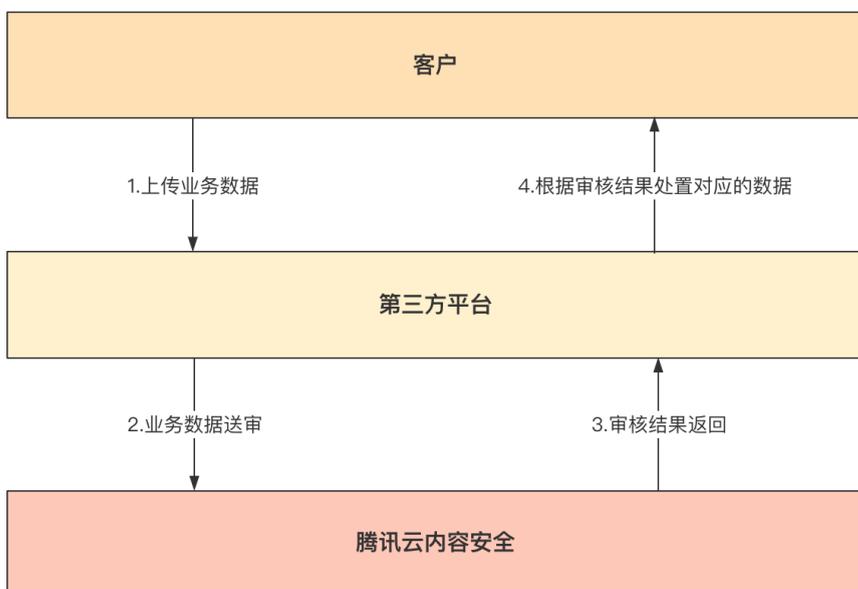
## 概述

内容安全分为 [文本内容安全](#)、[图片内容安全](#)、[音频内容安全](#)、[视频内容安全](#) 四个产品。

内容安全支持用户通过第三方平台接入内容安全服务，如您在第三方平台已有相应的业务数据信息，您可以选择由第三方平台协助您完成数据快速送审工作，同时内容安全产品团队会将审核结果返回至第三方平台，以便第三方平台进行相应配置，相关审核数据可以在内容安全控制台查看。

### 说明：

内容安全分为文本、图片、音频、视频四个场景，您可在 [步骤2：配置场景](#) 中，根据需要审核的类型，配置对应场景。



## 步骤1：新建应用

1. 登录 [内容安全控制台](#)，在概览页单击[应用管理](#) > [新增应用](#)，新增需要配置文本内容安全管理的应用。



2. 在应用管理页面，单击[新增应用](#)，输入应用名称，单击[保存](#)完成应用创建。

新建应用
✕

\*应用名称

仅支持数字、中英文、下划线，最多20个字符

保存
取消

## 步骤2：配置场景

1. 在应用管理页面，选择已创建的应用，单击**场景管理**。

应用名称	应用ID	关联场景数	token	创建时间	操作
测试	1		10	2023-07-06 15:06:03	<a href="#">识别统计</a> <a href="#">明细查询</a> <span style="border: 2px solid red; padding: 2px;">场景管理</span>

2. 在场景管理页面，单击**新建场景**，输入场景名称、选择行业分类、选择**文本内容安全**，单击**保存**即可完成场景创建。

**说明**

在审核服务管理中开启的审核服务可勾选，未开启的审核服务不可选。若想修改审核服务管理中的选项可单击**管理修改**。

新增场景
✕

\*场景名称

行业分类

\*关联审核服务 [管理](#)

文本内容安全

图片内容安全

音频内容安全

视频内容安全

视频流内容安全

音频流内容安全

保存
取消

参数名称	描述
场景名称	场景的文字描述，可使用中文、英文、数字及下划线组合，长度不超过20个字符。
行业分类	策略涉及的行业场景分类。
关联审核服务	选择场景需要关联的审核服务。

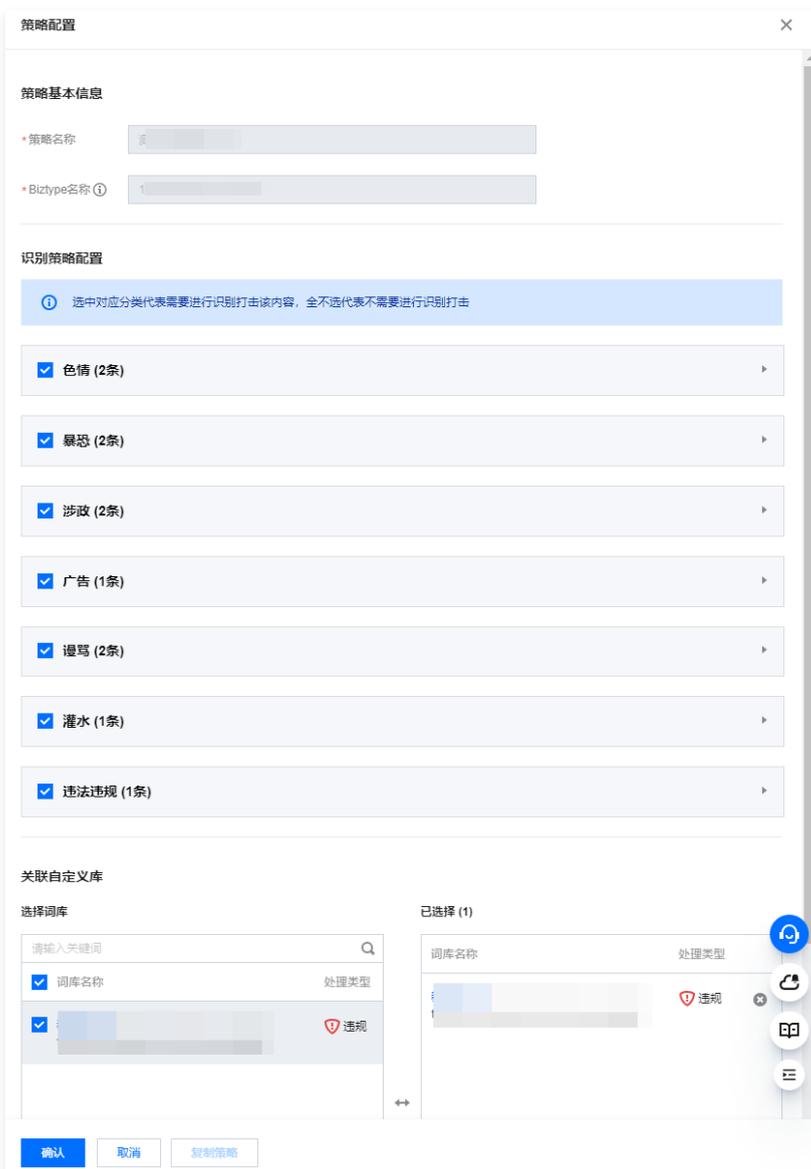
3. 场景创建完成，开始配置场景策略，单击**配置 > 文本内容安全**。



4. 在策略详情页面会显示当前系统默认策略，如需更改，单击编辑，即可修改当前策略。

**说明**

如未创建自定义词库，请参见 [自定义词库文档](#) 进行创建。



5. 完成后单击确认，完成策略配置。

### 步骤3: 获取 Token 和场景送审 URL

**说明**

- Token：用于调用审核服务时的身份验证。
- 场景送审 URL：内容安全审核服务的调用地址。

1. 在应用管理页面，选择目标应用，单击  复制送审 Token，并跳转回第三方平台进行配置。

**说明**

如您重置 Token，则在其他第三方渠道配置的原 Token 将会失效，需要您重新在第三方添加最新的 Token。



应用管理界面截图，显示应用列表。顶部有“新建应用”和“如何删除应用”按钮，以及搜索框。表格列包括：应用名称、应用ID、关联场景数、token、创建时间、操作。其中一行“测试”应用，其“token”列右侧有一个“重置”按钮，该按钮被红色方框高亮。

2. 在应用管理页面，选择已创建的应用，单击场景管理。



应用管理界面截图，显示应用列表。与上一图类似，但“操作”列中的“场景管理”按钮被红色方框高亮。

3. 在场景管理页面，选择目标场景，单击更多 > 复制送审 URL，并跳转回第三方平台进行配置。



场景管理界面截图，显示场景列表。顶部有“新建场景”和“如何删除场景”按钮，以及搜索框。表格列包括：场景名称、场景ID、所属行业、策略配置、Biztype、操作。其中一行“测试”场景，其“操作”列右侧有一个“更多”下拉菜单，展开后“复制送审URL”选项被红色方框高亮。

### 步骤4：跳转回第三方平台完成数据送审

在第三方平台配置内容安全审核服务的 Token 和场景送审 URL 后，请根据第三方平台后续的指引，完成数据送审。如有疑问，请 [联系我们](#)。

**说明**

用户可登录 [内容安全控制台](#)，查看送审的明细数据、统计数据，并进行策略配置等操作。

# 即时通信 IM 云端审核

最近更新时间：2025-02-05 16:02:22

即时通信 IM 云端审核功能，是天御内容安全产品，在云端对内容进行智能识别和拦截的服务实践。

- 支持对文本、图片内容同步审核，对音频、视频内容异步审核。
- 支持分别针对单聊、群聊、资料三大场景中产生的不同类型内容进行策略配置。
- 支持配置特定终端用户产生/接收的消息不送审，支持对返回审核结果进行配置。
- 支持查看审核用量抵扣明细、审核结果明细、审核识别统计等数据。
- 能够精准识别文本中出现可能令人反感、不安全或不适宜的内容，有效降低内容违规风险与有害信息识别成本。

## 步骤1：购买 IM 云端审核资源包

请前往 [即时通信 IM 购买页](#)，购买云端审核资源包。体验版、专业版、旗舰版应用均可购买，购买完成后即刻生效。

### ⚠ 注意：

开通服务后，默认为您开启图文审核场景，并配置默认审核策略（可识敏感事件、敏感人物、色情、辱骂、违禁等不安全内容），您可通过单击对应模块的编辑按钮，分别更改审核场景和审核策略。

## 步骤2：开启 IM 云端审核功能开关

1. 登录 [即时通信 IM 控制台](#)，在左侧导航栏中，选择内容审核 > 云端审核。
2. 在云端审核页面，单击页面右上角的 ，开启云端审核服务。

### ⚠ 注意：

如果您未提前购买云端审核资源包，在控制台直接开通云端审核服务后，产生的审核量将按照后付费计费规则扣除费用，请及时 [购买云端审核资源包](#)，计费规则详见 [计费说明](#)。

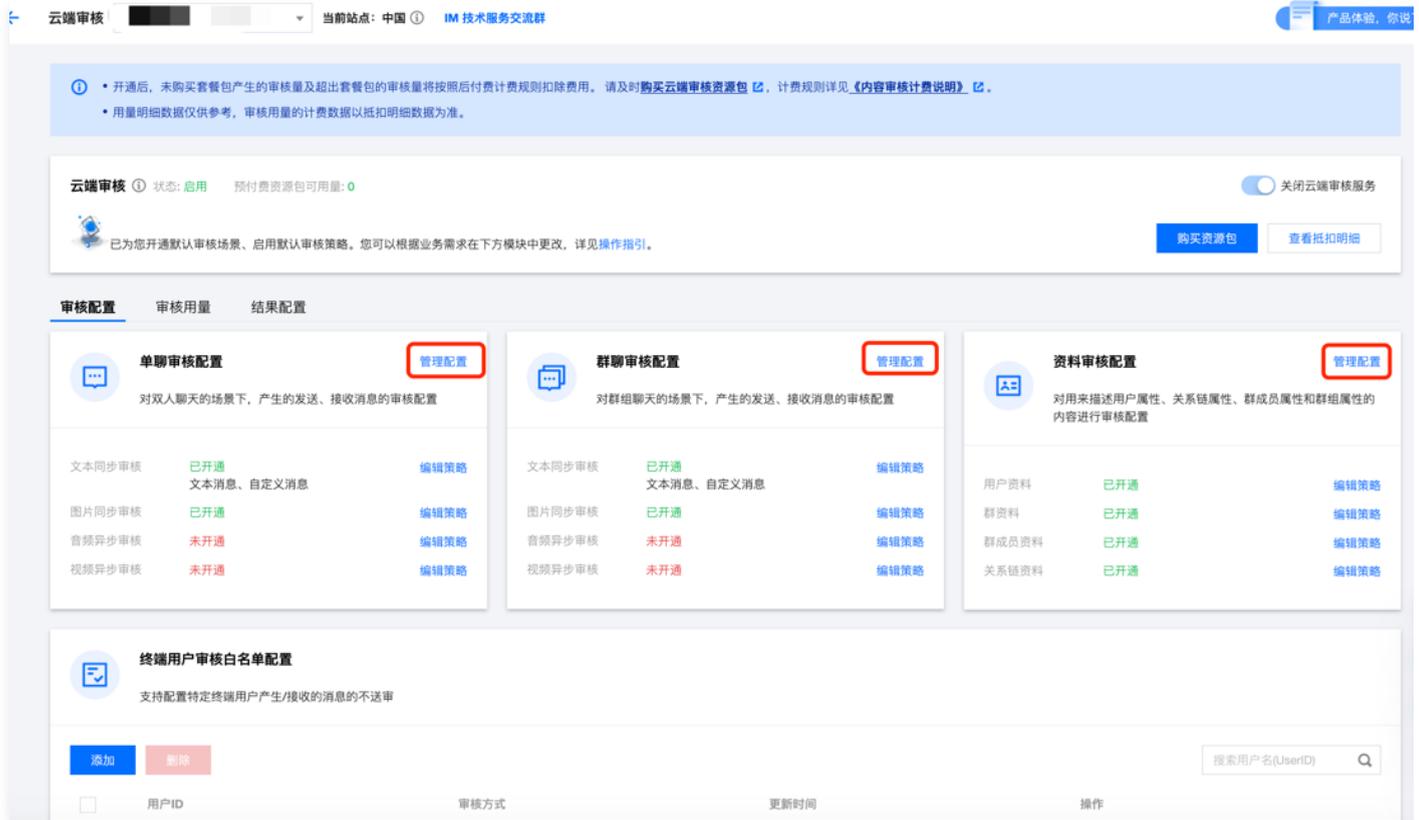


## 步骤3：自定义审核配置

### 审核场景配置

打开云端审核服务开关后，默认为您开启了单聊的文本和图片审核、群聊的文本和图片审核、用户资料/群资料/群成员资料/关系链资料的文本审核。

1. 如果需要开启音频/视频审核，或者关闭图文审核开关，请单击对应场景模块右上角的**管理配置**。



2. 在弹窗中，勾选或取消勾选对应审核项，单击**确定**即可。



**注意:**

单聊、群聊场景下，将视听/视频异步审核开启后，所有音频/视频都将正常送审并计费，但仅对存储在腾讯云即时通信 IM 的内容生效审核结果。如有自定义存储的音频/视频内容，请谨慎开启！如需审核自定义存储的音频/视频内容，请通过监听音频/视频审核结果的方式，进行拦截或撤回操作，详情请参见 [返回审核结果](#) 配置。

**审核策略配置**

每个审核场景及内容类型均已配置默认审核策略，默认审核策略包括敏感事件、敏感人物、色情、辱骂、违禁等风险的识别。如果需要编辑审核策略，请单击对应场景及内容类型右侧的**编辑策略**。

云端审核 当前站点: 中国 IM 技术服务交流群

开通后, 未购买套餐产生的审核量及超出套餐的审核量将按照后付费计费规则扣除费用。请及时购买云端审核资源包, 计费规则详见《内容审核计费说明》。  
用量明细数据仅供参考, 审核用量的计费数据以抵扣明细数据为准。

云端审核 状态: 启用 预付资源包可用量: 0 关闭云端审核服务

已为您开通默认审核场景、启用默认审核策略。您可以根据业务需求在下方模块中更改, 详见操作指引。

购买资源包 查看抵扣明细

**审核配置** 审核用量 结果配置

**单聊审核配置** 管理配置

对双人聊天的场景下, 产生的发送、接收消息的审核配置

文本同步审核	已开通	编辑策略
图片同步审核	已开通	编辑策略
音频异步审核	未开通	编辑策略
视频异步审核	未开通	编辑策略

**群聊审核配置** 管理配置

对群组聊天的场景下, 产生的发送、接收消息的审核配置

文本同步审核	已开通	编辑策略
图片同步审核	已开通	编辑策略
音频异步审核	未开通	编辑策略
视频异步审核	未开通	编辑策略

**资料审核配置** 管理配置

对用来描述用户属性、关系链属性、群成员属性和群组属性的内容进行审核配置

用户资料	已开通	编辑策略
群资料	已开通	编辑策略
群成员资料	已开通	编辑策略
关系链资料	已开通	编辑策略

### 文本/图片策略配置

单击编辑, 修改通用识别策略配置与自定义库信息, 单击确认保存配置。

- 通用识别策略配置: 勾选需要识别的风险类型。
- 自定义库信息: 勾选需要关联的自定义词库 (最多可支持设置20个自定义词库)。

策略配置
✕

---

**策略基本信息**

\*策略名称

\*Biztype名称

---

**识别策略配置**

选中对应分类代表需要进行识别打击该内容，全不选代表不需要进行识别打击  
如需了解更多详细的策略配置信息，请通过企业微信联系文本内容安全策略运营人员

色情 (2条) ▶

暴恐 (2条) ▶

涉政 (2条) ▶

广告 (1条) ▶

谩骂 (2条) ▶

灌水 (1条) ▶

违法违规 (1条) ▶

---

**关联自定义库**

选择词库 已选择 (0)

确认
取消
复制策略

### 音频策略配置

单击编辑，修改通用识别策略配置、截帧策略配置与自定义库信息，单击确认保存配置。

- 通用识别策略配置：勾选需要识别的风险类型。
- 截帧策略配置：音频切片时长配置，支持滑动设定或填写整数，范围[1,60]。
- 自定义库信息：勾选需要关联的自定义词库（最多可支持设置20个自定义词库）。

### 策略配置

策略基本信息

策略名称

Biztype名称

识别策略配置

1. 选中对应该分类代表需要进行识别打击该文本内容，全不选代表不需要进行识别打击  
2. 点播音频文件与一句话音频通用识别策略配置  
如需了解更多详细的策略配置信息，请通过企业微信联系天御内容安全策略运营人员

色情 (3条)

暴恐 (2条)

涉政 (2条)

广告 (1条)

谩骂 (2条)

违法违规 (1条)

点播音频文件截帧配置

音频切片时长  15 30 45 60 - 30 +

审核结果回调配置  违规回调  全量回调

## 视频策略配置

单击编辑，修改通用识别策略配置、截帧策略配置与自定义库信息，单击确认保存配置。

- 通用识别策略配置：勾选需要识别的风险类型，包括对图片和音频的识别。
- 截帧策略配置：审核内容配置，支持仅审核视频画面、仅审核音频和全部审核；图片截帧间隔，支持滑动设定或填写整数，范围[1,60]；音频切片时长配置，支持滑动设定或填写整数，范围[1,60]。
- 自定义库信息：勾选需要关联的自定义词库（最多可支持设置20个自定义词库）。

### 策略基本信息

策略名称

Biztype名称

---

### 识别策略配置

选中对应分类代表需要进行识别打击该内容，全不选代表不需要进行识别打击  
如需了解更多详细的策略配置信息，请通过企业微信联系全国内容安全策略运营人员

图片识别     音频识别

- 色情 (4条)
- 暴恐 (9条)
- 涉政 (10条)
- 违法违规 (6条)
- 广告 (2条)
- 图片文本审核 (1条)
- 未成年人保护 (2条)

---

### 截帧策略配置

审核内容配置  全部审核     仅审核视频画面     仅审核音频

图片截帧间隔  (范围: 5 - 60)

音频切片时长  (范围: 15 - 60)

审核结果回调配置  违规回调     全量回调

---

### 关联自定义库

选择词库

词库名称	处理类型
已选择 (0)	

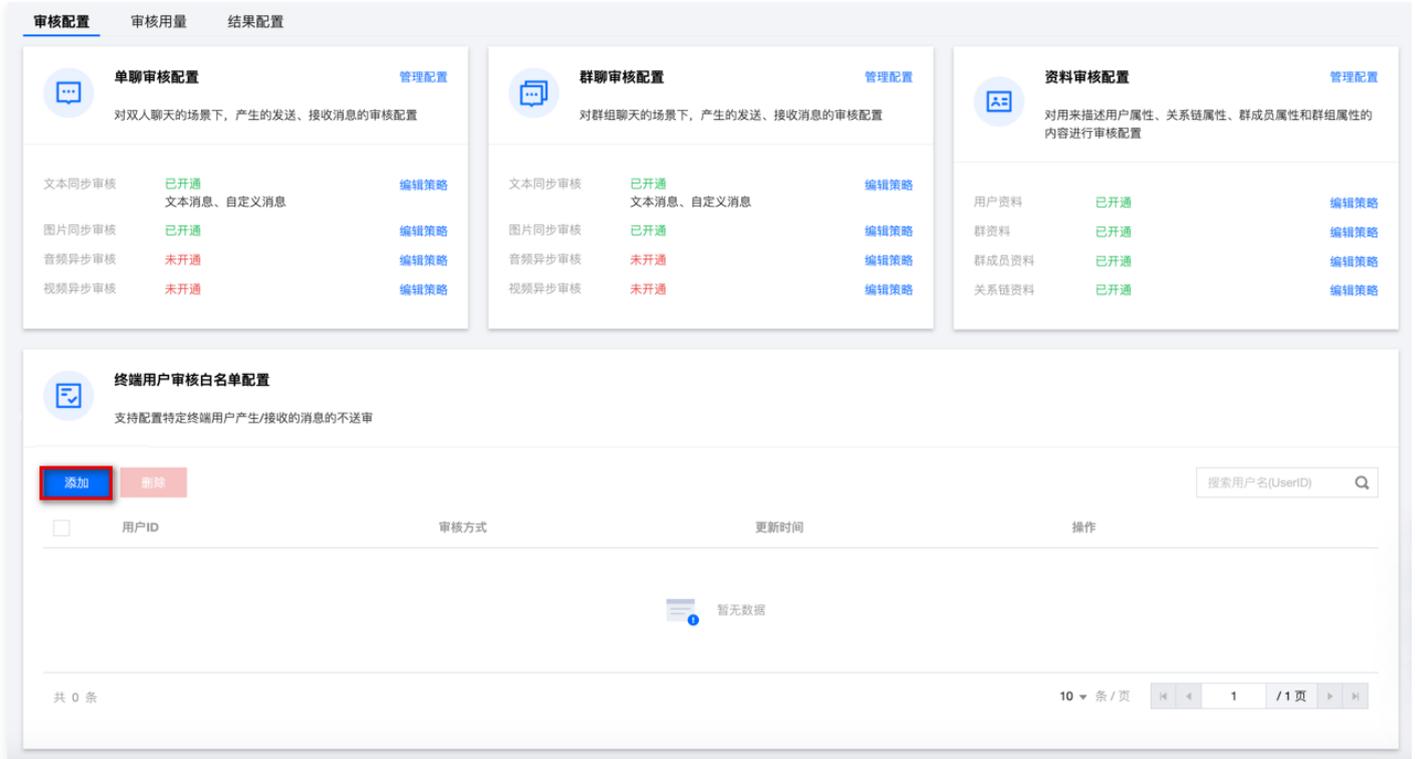
## 复制策略

图文音视频的编辑策略时，单击复制策略，可以将本策略复制到下拉选择的其他场景，单击确定。

## 终端用户审核白名单配置

如果您不需要审核特定终端用户发送/接收的消息内容，您可以通过终端用户审核白名单模块进行配置。具体操作如下：

1. 在云端审核页面的终端用户审核白名单配置模块，单击**添加**。



2. 在弹窗中，选择审核方式，输入终端用户 ID，单击**确定**。



3. 在终端用户审核白名单配置模块，支持对已添加的终端用户 ID 编辑不送审方式、删除/批量删除添加白名单记录。

## 步骤4：审核结果配置

### 返回审核结果

审核结果可以通过配置基础回调的方式，将结果转发给 App 后台。具体操作如下：

1. 返回审核结果开关默认关闭，如果需要打开，单击审核结果配置模块右上角的编辑，打开返回审核结果开关，单击确定。

云端审核 当前站点: 中国 IM 技术服务交流群 产品体验, 你说了算

① 开通后, 未购买套餐产生的审核量及超出套餐包的审核量将按照后付费计费规则扣除费用。请及时[购买云端审核资源包](#), 计费规则详见[《内容审核计费说明》](#)。  
• 用量明细数据仅供参考, 审核用量的计费数据以抵扣明细数据为准。

云端审核 ① 状态: 启用 预付费资源包可用量: 0 日结算后付费抵扣中, 请及时购买资源包  关闭云端审核服务

已为您开通默认审核场景、启用默认审核策略。您可以根据业务需求在下方模块中更改, 详见[操作指引](#)。 [购买资源包](#) [查看抵扣明细](#)

审核配置 审核用量 **结果配置**

**审核结果配置** [编辑](#)

返回审核结果 **已关闭**  
审核结果将发送至您在基础配置中配置的 URL 地址, 请您开启此功能前, 确保已配置基础回调。

下发拦截错误码 **已开启**  
用于配置消息发送方是否可收到消息被拦截的错误码提示。

2. 内容回调结果将发送至您在基础配置中配置的 URL 地址, 如果您在开启返回审核结果开关前未配置基础回调 URL, 请在打开开关时弹出的弹窗中填写基础回调 URL, 详情请参见 [配置指引文档](#)。

请配置基础回调

内容回调结果将发送至您在基础配置中配置的 URL 地址, 您需要在开启前, 完成下方的基础回调 URL 填写。详见 [配置指引](#)。

回调URL \*

确定

3. 开关开启后, 内容回调的结果将通过基础回调抄送到 App 后台, 支持拦截、所有结果的配置选项。

#### 注意

App 后台在收到回调请求之后, 务必校验请求 URL 中的参数 SDKAppID 是否是自己的 SDKAppID。

## 下发拦截错误码

针对图文同步审核, 默认下发拦截错误码, 即消息发送方会收到消息被拦截的错误码提示。

如果需要发送方对被消息拦截无感知, 即发送方不需要收到消息下发失败的提示, 可通过关闭“下发拦截错误码”开关实现。具体操作如下:  
在审核结果配置模块, 单击右上角的编辑, 关闭“下发拦截错误码”开关, 单击确定。

← 云端审核 当前站点: 中国 IM 技术服务交流群 产品体验, 你说了算

① 开通后, 未购买套餐包产生的审核量及超出套餐包的审核量将按照后付费计费规则扣除费用。请及时[购买云端审核资源包](#), 计费规则详见[《内容审核计费说明》](#)。  
 用量明细数据仅供参考, 审核用量的计费数据以抵扣明细数据为准。

云端审核 ① 状态: 启用 预付费资源包可用量: 0 日结算后付费抵扣中, 请及时购买资源包  关闭云端审核服务

已为您开通默认审核场景、启用默认审核策略。您可以根据业务需求在下方模块中更改, 详见[操作指引](#)。 [购买资源包](#) [查看抵扣明细](#)

审核配置 审核用量 **结果配置**

**审核结果配置** [编辑](#)

返回审核结果 **已关闭**  
 审核结果将发送至您在基础配置中配置的 URL 地址, 请您开启此功能前, 确保已配置基础回调。

下发拦截错误码 **已开启**  
 用于配置消息发送方是否可收到消息被拦截的错误码提示。

**注意**

为方便您回溯消息未成功发送的原因, 您需要在关闭“下发拦截错误码”开关前, 开启 [返回审核结果](#) 开关, 接收审核结果。

## 步骤5: 查看审核明细和抵扣明细

### 查看审核明细

在云端审核页面的审核用量中, 可查看今日审核用量数据。

**注意**

用量明细数据仅供参考, 审核用量的计费数据以抵扣明细数据为准。

← 云端审核 14r 当前站点: 中国 IM 技术服务交流群 产品体验, 你说了算

① 开通后, 未购买套餐包产生的审核量及超出套餐包的审核量将按照后付费计费规则扣除费用。请及时[购买云端审核资源包](#), 计费规则详见[《内容审核计费说明》](#)。  
 用量明细数据仅供参考, 审核用量的计费数据以抵扣明细数据为准。

云端审核 ① 状态: 启用 预付费资源包可用量: 0  关闭云端审核服务

已为您开通默认审核场景、启用默认审核策略。您可以根据业务需求在下方模块中更改, 详见[操作指引](#)。 [购买资源包](#) [查看抵扣明细](#)

审核配置 **审核用量** 结果配置

今日审核用量 更新时间 2022-12-26 21:03:27 [查看审核明细](#) [查看识别统计](#)

<p>文本审核</p> <p>0 次</p>	<p>图片审核</p> <p>0 次</p>	<p>音频审核</p> <p>0 次</p>	<p>视频审核</p> <p>0 次</p>
------------------------	------------------------	------------------------	------------------------

• 单击模块右上角的查看审核明细，可查看历史审核的明细数据。

即时通讯 返回旧版 接口计费说明

识别统计 **明细查询** 场景管理 自定义库名单

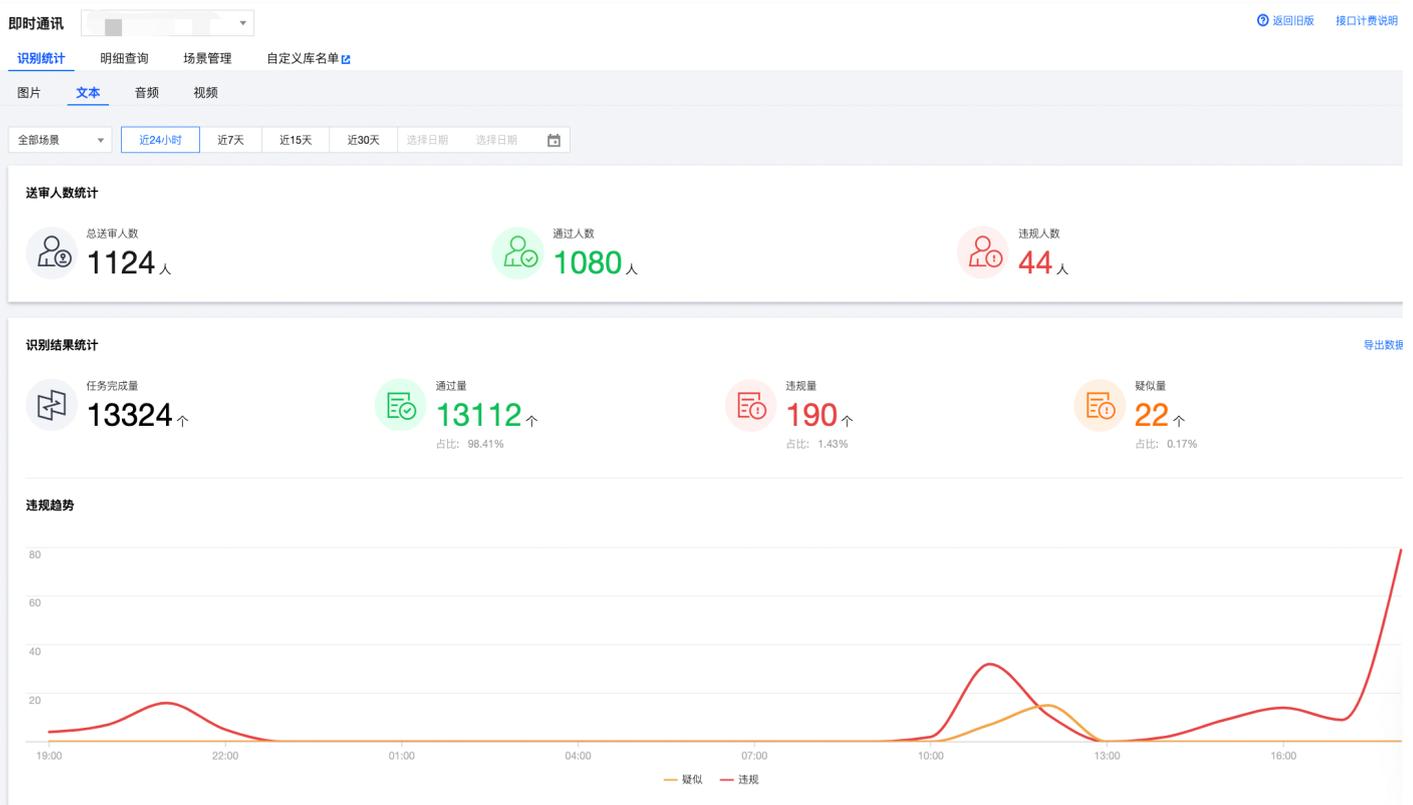
图片 **文本** 视频 音频

全部场景 今天 近7天 近15天 近30天 2022-12-28 00:00:00 ~ 2022-12-28 18:02:13 全部识别类型 全部处理建议

文本内容	关键字	文本ID	场景名称	识别类型	处理建议	创建时间	操作
Adq	-	c54	关系链	正常	通过	2022-12-28 18:01:44	<a href="#">查看策略</a>
AddSr	-	c54	关系链	正常	通过	2022-12-28 18:01:43	<a href="#">查看策略</a>
AddS	-	c54	关系链	正常	通过	2022-12-28 18:01:43	<a href="#">查看策略</a>
不会回消息的机器人C	-	c54	用户资料	正常	通过	2022-12-28 18:01:43	<a href="#">查看策略</a>
没回信息就是在放羊，一直没回就是羊丢了	-	c54	用户资料	正常	通过	2022-12-28 18:01:43	<a href="#">查看策略</a>
不会回消息的机器人B	-	c54	用户资料	正常	通过	2022-12-28 18:01:43	<a href="#">查看策略</a>
去宇宙摘星星啦，不方便回消息哦	-	c54	用户资料	正常	通过	2022-12-28 18:01:43	<a href="#">查看策略</a>
["link*":h	-	c54	群聊	广告	违规	2022-12-28 18:01:26	<a href="#">查看策略</a>
["link*	-	c54	群聊	广告	违规	2022-12-28 18:01:25	<a href="#">查看策略</a>
["busine	-	5	群聊	广告	违规	2022-12-28 18:01:07	<a href="#">查看策略</a>

共 11403 条 10 条 / 页 1 / 1141 页

• 单击模块右上角的查看识别统计，可查看历史审核的统计数据。



## 查看抵扣明细

1. 在云端审核页面，单击查看抵扣明细。

云端审核 当前站点: 中国 IM 技术服务交流群 产品体验, 你说了算

• 开通后, 未购买套餐包产生的审核量及超出套餐包的审核量将按照后付费计费规则扣除费用。请及时[购买云端审核资源包](#), 计费规则详见[《内容审核计费说明》](#)。  
• 用量明细数据仅供参考, 审核用量的计费数据以抵扣明细数据为准。

云端审核 状态: 启用 预付费资源包可用用量: 0 关闭云端审核服务

已为您开通默认审核场景、启用默认审核策略。您可以根据业务需求在下方模块中更改, 详见[操作指引](#)。

购买资源包 查看抵扣明细

2. 在抵扣明细页面，支持筛选、查看特定日期内审核产生用量的具体抵扣数据（每小时更新）。

抵扣明细

用量明细数据仅供参考, 审核用量的计费数据以此页面抵扣明细数据为准

7天 14天 30天 2022-12-15 00:00:00 ~ 2022-12-28 23:59:59

抵扣时间	抵扣方式	资源包	审核类型	审核用量	抵扣比例	实际抵扣量
2022-12-28 15:00:00	后付费	-	图片	4	1 : 0.65	2.6
2022-12-28 15:00:00	后付费	-	文本	2607	1 : 1	2607
2022-12-28 14:00:00	后付费	-	视频	0.5	1 : 56	28
2022-12-28 14:00:00	后付费	-	图片	12	1 : 0.65	7.8
2022-12-28 14:00:00	后付费	-	文本	1369	1 : 1	1369
2022-12-28 13:00:00	后付费	-	图片	5	1 : 0.65	3.25
2022-12-28 13:00:00	后付费	-	文本	327	1 : 1	327
2022-12-28 12:00:00	后付费	-	文本	236	1 : 1	236
2022-12-28 11:00:00	后付费	-	图片	14	1 : 0.65	9.1
2022-12-28 11:00:00	后付费	-	文本	861	1 : 1	861

共 33 条 10 条 / 页 1 / 4 页

3. 如果您已购买预付费资源包，可在资源包管理页面查看资源包可用量、有效期等信息。

← 抵扣明细
当前站点: 中国  ⓘ  [IM 技术服务交流群](#)
产品体验, 你说了算

抵扣明细    **资源包管理**

购买资源包

**有效**
无效

资源包 ID	套餐包类型	剩余量	总量	创建时间	到期时间
<div style="display: flex; align-items: center; justify-content: center;"> <span>暂无数据</span> </div>					

共 0 条
10 条 / 页

⏪
⏩
1
/ 1 页
⏪
⏩

# 论坛评论区文本内容审核

最近更新时间：2025-02-05 16:02:22

可以直接调用**文本内容检测 (TextModeration)** 接口，对论坛评论区内容（例如购物网站评论、社区回帖、视频评论区等）进行识别。

## 说明：

- 请在调用前确保目前账号至少拥有**文本内容安全服务**的访问权限，有关权限配置的相关信息，敬请参阅 [CAM 授权指引](#) 文档。
- 若无法访问文本内容安全服务，则请开通服务/检查计费信息（主账号），或向管理员或主账号申请相应权限（子账号/协作者账号）。

## 步骤一：配置任务策略（可选）

建议您使用配置任务策略，可根据业务需求配置识别策略，用于个性化服务体验。

## 说明：

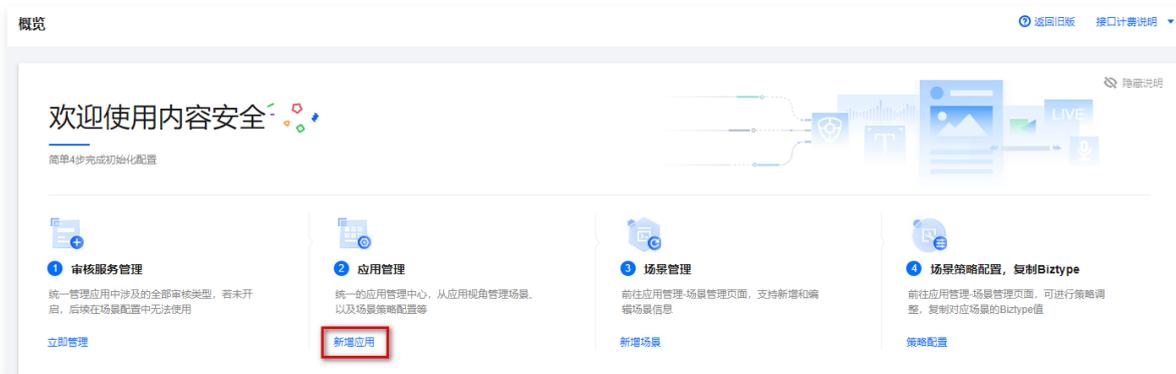
- 腾讯云内容安全服务已预设默认策略，如使用默认策略，可略过此步骤。
- 默认策略为天御多行业模型沉淀的策略配置，适用于大部分的内容安全需求。

## 前提条件

- 已获取腾讯云主账号或子账号。
- 登录账号已拥有文本内容安全服务的读写权限。
- 登录账号已开通文本内容安全服务。

## 操作步骤

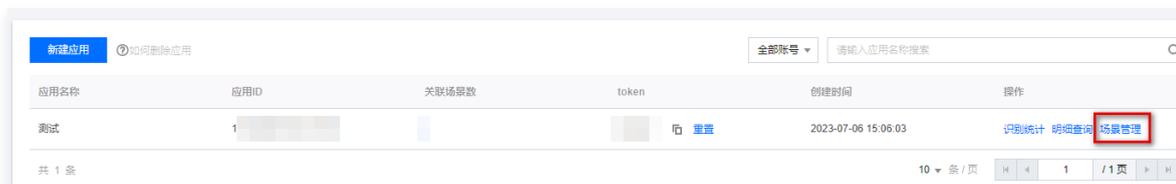
1. 登录 [内容安全控制台](#)，在概览页单击**应用管理 > 新增应用**，新增需要配置文本内容安全管理的应用。



2. 在应用管理页面，单击新增应用，输入应用名称，单击**保存**完成应用创建。



3. 在应用管理页面，选择刚创建的应用，单击**场景管理**。



4. 在场景管理页面，单击**新建场景**，输入场景名称、选择行业分类、选择文本内容安全，单击**保存**即可完成场景创建。

新增场景
✕

\* 场景名称

行业分类

\* 关联审核服务 [管理](#)

文本内容安全

图片内容安全

音频内容安全

视频内容安全

视频流内容安全

音频流内容安全

保存
取消

参数名称	描述
场景名称	场景的文字描述，可使用中文、英文、数字及下划线组合，长度不超过20个字符
行业分类	策略涉及的行业场景分类
关联审核服务	选择场景需要关联的审核服务

5. 场景创建完成，开始配置场景策略，单击**配置** > **文本内容安全**。

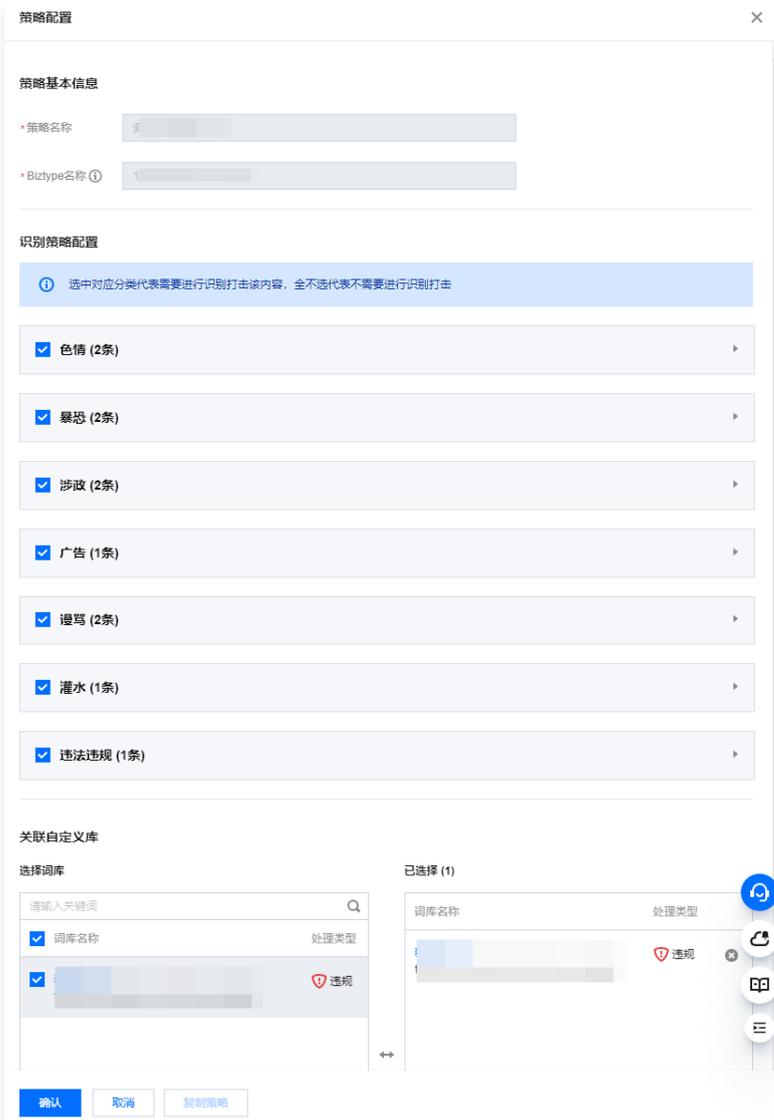
场景管理
返回旧版 [接口计费说明](#)

新建场景 [如何删除场景](#)

场景名称	场景ID	所属行业	策略配置	操作
测试	1		<div style="border: 2px solid red; padding: 2px; display: inline-block;"> <span style="color: green;">•</span> 文本内容安全                             </div>	<a href="#">编辑场景</a> <a href="#">更多</a>
测试			<span style="color: green;">•</span> 视频内容安全 <span style="color: green;">•</span> 图片内容安全	<a href="#">编辑场景</a> <a href="#">更多</a>

共 2 条
10 条 / 页

6. 在策略详情页面会显示当前系统默认策略，如需更改，单击**编辑**，即可修改当前策略。



参数名称	描述
识别策略配置	根据业务需求，选择是否需要识别不同类型的识别内容。
关联自定义库	选择是否需要使用自定义词库用于内容识别，如暂无自定义库，可跳过，或保存当前策略后前往 <a href="#">步骤二：配置自定义词库</a> 。

7. 完成后单击**确认**，完成策略配置。

## 步骤二：配置自定义词库（可选）

自定义词库用于添加个性化的识别词库，识别图片上是否含有违规文本。

### 说明

如无需配置自定义词库，可略过此步骤。

### 前提条件

- 已获取腾讯云主账号或子账号。
- 登录账号已拥有文本内容安全服务的读写权限。
- 登录账号已开通文本内容安全服务。

### 操作步骤

1. 登录 [内容安全控制台](#)，在左侧导航栏中，选择**名单管理 > 关键词名单 > 自定义名单**。

2. 在自定义名单页面，单击**新建词库**，填写文本库名称。

**新建词库** ✕

\* 文本库名称

\* 处理建议  违规  疑似  放过

\* 匹配模式  模糊匹配 (?)  精确匹配 (?)

确定
取消

参数名称	描述
文本库名称	词库的文字描述，可使用中文、英文、数字及下划线组合，长度不超过32个字符。
处理建议	可选择违规、疑似、放过。 <ul style="list-style-type: none"> <li>违规：确认为不良信息。</li> <li>疑似：可能为不良信息，需要人工识别。</li> <li>放过：业务方认为的正常信息。</li> </ul>
匹配模式	可选择精确匹配或模糊匹配。 <ul style="list-style-type: none"> <li>精确匹配：对输入文本进行匹配，匹配对象需完全一致。</li> <li>模糊匹配：可检测变体后的输入词，支持拆分字、形似字、音似字、简繁体、大小写、大写数字等形式的相似词进行匹配。</li> </ul>

3. 单击**确定**，即可创建自定义词库。

4. 在自定义词库页面下方的列表中，将显示刚创建的词库。说明自定义词库的不同颜色代表不同的屏蔽逻辑，红色代表违规，橙色代表疑似。

**说明**  
自定义词库的不同颜色代表不同的屏蔽逻辑，红色代表违规，橙色代表疑似，绿色代表通过。

关键词名单 🔙 返回旧版 接口计费说明

预设名单 自定义名单

(?) 对于名单内容的操作成功后，会在 10 分钟内生效

新建词库

词库名称	处理建议	匹配模式	最近修改时间	操作
test	疑似	精确匹配	2023-06-20 17:49:41	关联场景 编辑 删除
测试词库0601	违规	精确匹配	2023-06-01 14:23:47	关联场景 编辑 删除
测试词库	违规	精确匹配	2023-02-13 11:33:06	关联场景 编辑 删除

5. 在自定义名单页面，选择需要批量导入关键词的词库，单击**管理**。

新建词库

词库名称	处理建议	匹配模式	最近修改时间	操作
test	违规	模糊匹配	2023-07-07 10:32:39	关联场景 <span style="border: 1px solid #00aaff; padding: 2px 5px;">管理</span> 删除

6. 在编辑词库页面，单击**添加关键词**，输入关键词，单击**确定**，即可保存关键词至当前词库。

添加关键词
✕

识别类型 请选择识别类型

关键词类型  普通词  组合词  英文

添加关键词 多个以回车换行分隔，一次提交不超过2000个

关键词备注 请输入关键词的备注说明，最多可输入100个字

0 / 100

1. 每个关键词以换行来确定，单个关键词长度20个汉字以内  
 2. 支持以复制粘贴的方式批量导入，单次最多添加关键词2000个，词与词之间用换行符分隔  
 3. 关键词的添加个数上限为10000个  
 4. 如需添加组合词，选择关键词类型：组合词，使用&符号对两个或者多个词进行连接，添加格式为“关键词1&关键词2...”，其中关键词只支持纯中文，这些关键词同时出现时则会命中

确定
取消

参数名称	描述
处理建议	识别模型对应的违规类型。
关键词	<ul style="list-style-type: none"> <li>单个关键词以换行来确定，长度在20个汉字或字母以内。</li> <li>支持以复制粘贴的方式批量导入，单次最多添加关键词2000个，词与词之间用换行符分隔。</li> <li>关键词添加个数上限为10000个。</li> <li>如需添加组合词，选择关键词类型：组合词，使用&amp;符号对两个或者多个词进行连接，添加格式为“关键词1&amp;关键词2...”，其中关键词只支持纯中文，这些关键词同时出现时则会命中。</li> </ul>

### 步骤三：创建任务及获取识别结果

在完成以上步骤后，可以调用**文本内容检测（TextModeration）**接口创建评论内容识别任务，具体方法如下：

- 确保您的待识别文本内容符合接口传入的 [文件格式要求](#)。
- 参考接口的 [接口文档说明](#)，填入相应的输入参数。
- 若任务创建成功，接口会返回详细识别结果，请参考 [文本内容识别示例](#) 进一步了解返回参数示例；若任务创建失败，接口会返回错误码，请参考 [业务错误码](#) 和 [公共错误码](#) 进行问题排查。

#### 📌 说明

在进行服务接入时，可以使用 [API Explorer](#) 工具进行在线调试。

# 权限管理

## 访问控制和权限管理

最近更新时间：2025-02-05 16:02:22

### 概述

对于腾讯文本内容安全（Text Moderation System，TMS）资源，不同企业之间或同企业多团队之间，需要对不同的团队或人员配置不同的访问权限。您可通过访问管理（Cloud Access Management，CAM）对子账号设置不同的操作权限，使得不同团队或人员能够相互协作。首先，我们需要先了解几个关键概念：主账号、子账号（用户）和用户组。CAM 的相关术语、配置详细描述请参见访问管理的 [词汇表](#)。

### 主账号

主账号又被称为开发商。用户申请腾讯云账号时，系统会创建一个用于登录腾讯云服务的主账号身份。主账号是腾讯云资源使用计量计费的基本主体。主账号默认拥有其名下所拥有的资源的完全访问权限，包括访问账单信息，修改用户密码，创建用户和用户组以及访问其他云服务资源等。默认情况下，资源只能被主账号所访问，任何其他用户访问都需要获得主账号的授权。

### 子账号（用户）和用户组

子账号是由主账号创建的实体，有确定的身份 ID 和身份凭证，拥有登录腾讯云控制台的权限。

子账号默认不拥有资源，必须由所属主账号进行授权。

一个主账号可以创建多个子账号（用户）。

一个子账号可以归属于多个主账号，分别协助多个主账号管理各自的云资源，但同一时刻，一个子账号只能登录到一个主账号下，管理该主账号的云资源。

子账号可以通过控制台切换开发商（主账号），从一个主账号切换到另外一个主账号。

子账号登录控制台时，会自动切换到默认主账号上，并拥有默认主账号所授予的访问权限。

切换开发商之后，子账号会拥有切换到的主账号授权的访问权限，而切换前的主账号授予的访问权限会立即失效。

用户组是多个相同职能的用户（子账号）的集合。您可以根据业务需求创建不同的用户组，为用户组关联适当的策略，以分配不同权限。

### 操作步骤

授权子账号访问 TMS 分为三个步骤：创建子账号、对子账号授予权限、子账号访问 TMS 资源。

#### 步骤1：创建子账号

在 CAM 控制台可创建子账号，并配置授予子账号的访问权限。具体操作如下所示：

1. 使用主账号登录 [CAM 控制台](#)。
2. 选择 **用户** > **用户列表** > **新建用户**，进入新建用户页面。
3. 选择 **自定义创建**，选择 **可访问资源并接收消息类型**，单击 **下一步**。
4. 按照要求填写用户相关信息。
  - **设置用户信息**：输入子用户名称，例如 Sub\_user。输入子用户的邮箱，您需要为子用户添加邮箱来获取由腾讯云发出的绑定微信的邮件。
  - **访问方式**：选择编程访问和腾讯云控制台访问。其他配置可按需选择。
5. 填写用户信息完毕后，单击 **下一步**，进行身份验证。
6. 身份验证完毕，设置子用户权限。根据系统提供的策略选择，策略配置说明可参考 [步骤2](#)。
7. 设置用户标签，该项为可选项，可按需设置，单击 **下一步**。
8. 确认配置信息无误后，单击 **完成**即可创建子账号。

#### 步骤2：授予子账号控制台和 API 资源访问权限

根据 [步骤1](#) 设置的访问方式：编程访问和腾讯云控制台访问。

##### 编程访问

当使用子账号通过编程（例如 API、SDK 和工具等）访问 TMS 资源时需要先获取主账号的 APPID、子账号的 SecretId 和 SecretKey 信息。您可以在访问管理控制台生成子账号的 SecretId 和 SecretKey。

1. 使用主账号登录 [CAM 控制台](#)。
2. 选择 **用户列表**，进入用户列表页面。
3. 单击 **子账号用户名称**，进入子账号信息详情页。

4. 单击 **API 密钥** 页签，并单击 **新建密钥** 为该子账号创建 **SecretId** 和 **SecretKey**。

5. 按照 **步骤1** 授权 TMS 的 CAM 策略 **QcloudTMSFullAccess** 后，至此您就可以通过子账号的 **SecretId** 和 **SecretKey** 访问 TMS 资源。

## 腾讯云控制台

子用户被授予权限后，可在 **子用户登录界面** 输入主账号 ID、子用户名和子用户密码登录控制台，并在 **云产品** 中选择单击 **内容安全**，即可进入控制台。登录控制必须需授权以下 CAM：

- 图片策略：QcloudIMSReadOnlyAccess
- 文本策略：QcloudTMSReadOnlyAccess（如需使用 **服务体验**，需授权QcloudTMSFullAccess）
- 音频策略：QcloudIAMSReadOnlyAccess
- 视频策略：QcloudVMReadOnlyAccess
- CMS 策略：QcloudCMSFullAccess

## 子账号查询其他子用户数据

主账号可以查看所有子用户的调用数据，而子账号只能查看自己账号的调用记录。如果想让子账号也能够访问其他用户的信息，可以通过授权 **ListUsers** 权限实现。当前功能不支持角色 SSO 登录方式。

1. 在 **访问管理** > **策略** 页面，单击 **新建自定义策略**，创建一个自定义策略配置 **ListUsers** 接口权限并关联对应用户。



2. 在新的窗口中，选择 **按策略生成器创建**。



3. 在编辑策略页面，服务选择 **访问管理 (cam)**，操作选择 **ListUsers**，资源选择 **全部**，单击 **下一步**。

1 编辑策略 > 2 关联用户/用户组/角色

可视化策略生成器 JSON

访问管理 (1 个操作)

效果 (Effect)	<input checked="" type="radio"/> 允许 <input type="radio"/> 拒绝
服务 (Service)	访问管理 (cam)
操作 (Action)	读操作 编辑 ListUsers 拉取子用户列表
资源 (Resource) 收起	<input checked="" type="radio"/> 全部资源
条件 (Condition)	<input type="checkbox"/> 来源 IP ⓘ 添加其他条件

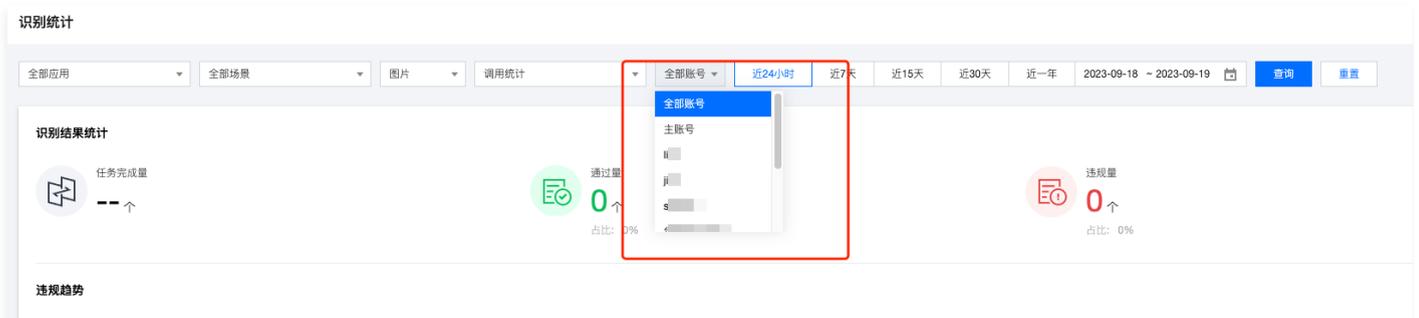
+ 添加权限

下一步 字符数: 168 (最多6144)

4. 在关联用户/用户组/角色页面，输入策略名称和描述，并且授予需要相应权限的用户和用户组，单击完成。



5. 授权后，子账号登录控制台后，对应的多个页面的界面查询框部分有用户选择框。并且默认是当前子账户，可以手动选择全部账户，主账户和其他账户，数据权限和主账号权限一致。



## 热点问题

### 腾讯云账号的 UIN 和 APPID 是什么？子账号是什么？

1. UIN 即账号 ID，是唯一的且不能修改，APPID 是腾讯云账号的 APPID，是与账号 ID 有唯一对应关系的应用 ID，部分腾讯云产品会使用此 APPID。可在 [腾讯云控制台](#) 的账号信息中心查看 UIN 和 APPID。
2. 子账号是由主账号创建的实体，有确定的身份 ID 和身份凭证，拥有登录腾讯云控制台的权限。子账号默认不拥有资源，必须由所属主账号进行授权。一个主账号可以创建多个子账号。子账号身份 ID 和凭证可以在 [腾讯云控制台](#) 查看。

账号信息

**基本信息**

账号ID ID

APPID ID

**账号权限**

账号身份 子用户

所属主账号 (昵称: )

**登录方式**

微信 支持微信扫码授权登录

修改

**主账号身份信息**

认证状态 已认证 实名认证

所属行业 :

### 如何查看腾讯云后台子账号?

可以通过账号登录 [访问管理](#) > [用户列表](#) 查看已创建的子账号信息。

#### 用户列表

**如何查看更多信息?**  
访问管理对您的敏感信息进行安全升级保护，您可以点击列表中左侧下拉按钮 **▶** 查看用户的身份安全状态、已加入组以及消息订阅等更多

新建用户
更多操作 ▼

	用户名称 <span style="font-size: small;">↕</span>	用户类型 <span style="font-size: small;">▼</span>
▶	子	主账号
▶	u	子用户
▶	n	子用户
▶	y	子用户

### 如何获取 API 密钥?

可以通过主账号登录 [访问管理](#) > [API 密钥管理](#) 查看已创建的子账号信息。

APPID	密钥	创建时间	最近访问时间	状态	
1	SecretId: AI SecretKey: ***** <span style="font-size: small;">显示</span>	201	21	20	已启用
1	SecretId: AI SecretKey: ***** <span style="font-size: small;">显示</span>	202	12	-	已启用

**说明:**

- AppID 和 SecretId 不是一样的。
- 在 云 API 密钥管理页面可以新建 SecretId, 每个账号最多可以创建两个 SecretId, 没有有效期。
- 在 API 密钥管理页面获得的密钥, 适用腾讯云所有的 API 调用使用。

### 访问管理 CAM 是什么?

是指帮助客户安全管理对腾讯云产品和资源的访问，默认主账号拥有腾讯云所有权限，客户需要用子账号来调用服务，就需要主账号来给予子账号授权对应的CAM权限。否则会返回如下错误：

```
{
  "Response":{
    "Error":{
      "Code":"AuthFailure.UnauthorizedOperation",
      "Message":"You are not authorized to perform this operation. Check your CAM policies, and ensure that you are using the correct access keys. [[request id:ba58220a-0710-4fb4-b7b0-11d4f5a04f94]you are not authorized to perform operation (tms:TextModeration)\nresource (*) has no permission\n]"
    },
    "RequestId":"ba58220a-0710-4fb4-b7b0-11d4f5a04f94"
  }
}
```