

# 天翼云内容审核 用户操作手册

中国电信股份有限公司云计算分公司

# 目 录

|                       |           |
|-----------------------|-----------|
| <b>1 产品介绍 .....</b>   | <b>6</b>  |
| 1.1 产品简介 .....        | 6         |
| 1.2 产品功能 .....        | 6         |
| 1.3 产品特点 .....        | 7         |
| 1.4 使用场景 .....        | 7         |
| <b>2 定价.....</b>      | <b>10</b> |
| 内容审核产品定价说明 .....      | 10        |
| <b>3 快速入门.....</b>    | <b>11</b> |
| 3.1 创建内容审核应用 .....    | 11        |
| 3.1.1 准备工作.....       | 11        |
| 3.1.2 创建内容审核应用订单..... | 12        |
| 3.1.3 订单确认.....       | 12        |
| 3.2 内容审核控制台 .....     | 14        |
| <b>4 API 文档.....</b>  | <b>16</b> |
| 4.1 文本涉黄审核接口说明 .....  | 16        |
| 接口描述 .....            | 16        |

|                   |    |
|-------------------|----|
| 接口要求 .....        | 16 |
| 请求说明 .....        | 16 |
| 请求示例 .....        | 16 |
| 请求头说明 .....       | 16 |
| 请求参数 .....        | 17 |
| 请求示例代码 .....      | 17 |
| 返回说明 .....        | 18 |
| API 调用成功返回值 ..... | 18 |
| API 调用失败返回值 ..... | 19 |
| 返回示例 .....        | 19 |
| 错误码说明 .....       | 20 |
| 4.2 文本涉政审核 .....  | 21 |
| 接口描述 .....        | 21 |
| 接口要求 .....        | 21 |
| 请求说明 .....        | 21 |
| 请求示例 .....        | 21 |
| 请求头说明 .....       | 21 |
| 请求参数 .....        | 22 |

---

|                     |    |
|---------------------|----|
| 请求示例代码.....         | 22 |
| 返回说明 .....          | 23 |
| API 调用成功返回值 .....   | 23 |
| API 调用失败返回值 .....   | 24 |
| 返回示例.....           | 24 |
| 错误码说明 .....         | 24 |
| 3.3 图像涉黄审核接口说明..... | 25 |
| 接口描述 .....          | 25 |
| 接口要求 .....          | 25 |
| 请求说明 .....          | 26 |
| 请求示例.....           | 26 |
| 请求头说明 .....         | 26 |
| 请求参数.....           | 26 |
| 请求示例代码.....         | 26 |
| 返回说明 .....          | 28 |
| API 调用成功返回值 .....   | 28 |
| API 调用失败返回值 .....   | 29 |
| 返回示例.....           | 29 |

---

|                     |           |
|---------------------|-----------|
| 错误码说明 .....         | 30        |
| 3.4 图像涉暴恐审核接口 ..... | 31        |
| 接口描述 .....          | 31        |
| 接口要求 .....          | 31        |
| 请求说明 .....          | 31        |
| 请求示例 .....          | 31        |
| 请求头说明 .....         | 31        |
| 请求参数 .....          | 32        |
| 请求示例代码 .....        | 32        |
| 返回说明 .....          | 33        |
| API 调用成功返回值 .....   | 33        |
| API 调用失败返回值 .....   | 34        |
| 返回示例 .....          | 34        |
| 错误码说明 .....         | 35        |
| <b>5 常见问题 .....</b> | <b>37</b> |

#### 修订历史

| 版本号  | 日期      | 修订人 | 修订内容 | 修订原因 |
|------|---------|-----|------|------|
| V1.0 | 2020.11 |     | 初版   |      |
|      |         |     |      |      |
|      |         |     |      |      |
|      |         |     |      |      |
|      |         |     |      |      |
|      |         |     |      |      |

# 1 产品介绍

## 1.1 产品简介

内容审核是中国电信自主研发具有综合审核文本、图片能力的审核系统提供的能力之一，对网络流量用户生成文本、图片内容中的涉黄、涉政、涉暴恐等敏感内容进行智能审核，帮助用户降低业务违规风险。

天翼云用户通过订购文本内容审核产品，无需进行模型代码开发仅需 API 接口对接，即可轻松、高效、准确的实现对文本的内容审核，将有效减少网络、云存储中违规内容的生产、流通和消费。

## 1.2 产品功能

- 文本涉黄审核

对文本中文字内容进行检测分析，准确分析审核文本内容中不合规范的涉黄、低俗内容信息，降低违规信息传播风险，提升用户应用平台使用体验。

- 文本涉政审核

对文本中文字内容进行检测分析，准确分析审核文本内容中的涉政敏感、反动等不良信息，降低违规信息传播风险，提升用户应用平台使用体验。

- 图像涉黄审核

对图像内容的进行检测分析，准确对图像内容中涉黄信息进行识别并对涉黄程度量化，自动识别涉黄、低俗等内容，降低违规信息传播风险，提升用户应用平台使用体验。

- 图像涉暴恐审核

对图像内容的进行检测分析，准确对图像内容内容中涉暴信息进行识别，降低违规信息传播风险，提升用户应用平台使用体验。

## 1.3 产品特点

- 识别准确率高

使用深度学习与机器学习相结合的方式，基于大规模和多主题数据的模型训练，具备较高的识别率和较低的误判率。

- 单次审核多个对象

用户单次请求审核服务时，支持同时审核多张图片或者多条文本，一次性返回所有数据的审核结果。

- 覆盖多领域的内容审核

涵盖目前主要的违规非法类的信息审核，可持续扩展更多领域的审核能力。

- 快速响应

配备高配 GPU 和 CPU 等硬件来支撑能力，审核速度快，大幅度缩短了反馈时间，提升用户体验。

## 1.4 使用场景

- 资讯、文学等内容密集领域

通过使用文本内容审核服务，不使用人工对用户内容进行检查，不留存用户数据内容，对媒体资讯，文学发布网站等内容进行智能文本内容审核。

- 聊天、论坛内容等场景

通过使用文本内容审核服务，可对场景内的文本内容中涉黄、反动等文本内容，净化网络环境。

- 图像内容密集领域

通过使用图像内容审核服务，不使用人工对用户内容进行检查，不留存用户数据内容，保护用户的个人隐私，同时又能识别用户内容是否正常，做到合法合规运营。

- 网络资讯、媒体等场景

对基于网络传播发布的网络资讯、媒体发布内容中的图像资料进行自动化审核，识别其中的涉黄涉暴等违规内容，降低违规信息传播风险



# 2 定价

## 内容审核产品定价说明

目前天翼云内容审核产品根据接口调用次数的不同，分为五档资源包：180 万次/年、720 万次/年、3600 万次/年、1.8 亿次/年、3.6 亿次/年。用户可以根据自身需要购买多个资源包。各档位资源包的价格如下：

| 产品规格     | 价格（元/年） |
|----------|---------|
| 180万次/年  | 4000    |
| 720万次/年  | 14000   |
| 3600万次/年 | 63000   |
| 1.8亿次/年  | 225000  |
| 3.6亿次/年  | 360000  |

### 订购说明：

1、为满足客户不同业务使用量需求，每类 API 设置五档套餐，如：用户评估认为应用每年文本涉黄识别的文本数量大概为 180 万次，则可以购买一个 180 万次/年的 API 调用次数包，若大概为 360 万次，则可以购买两个 180 万次/年的 API 调用次数包；

2、每篇文本识别需调用一次 API，若用户采取批量上传的方式，则按照上传的文本数量计算 API 调用次数；

3. 用户预先购买 N 个固定额度的资源包，后续使用过程中产生的文本涉黄识别或文本涉政识别的次数从资源包中抵扣，有效期内未使用的资源不会流转至下一年。有效期内资源包中的次数用完之后，用户无法再调用 AI 产品 API。对于有效期内未使用的资源包剩余调用次数，本产品不支持退订。

# 3 快速入门

## 3.1 创建内容审核应用

### 3.1.1 准备工作

首先，用户使用天翼云网门户账号登录天翼云网门户后，点击菜单栏“产品”项，进入【大数据与AI】，即可看到内容审核产品相应服务。以“文本涉黄审核”为例，点击“文本涉黄审核”名称即可进入“文本涉黄审核”产品详情页。



点击【立即使用】按钮，进入文本内容审核创建页面，点击“新的应用”



### 3.1.2 创建内容审核应用订单

用户根据自身需要输入“应用名称”与“应用概述”。

应用名称：长度为 4-15 个字符。



新建应用

\* 应用名称: 文本内容审核应用场景

应用概述: 文本内容审核应用场景

取消 确定

用户创建完应用后，点击“确定”和“下一步”，进入应用订购页面

| 内容安全    | 人脸识别 | 安全生产  | 图像识别  | 印刷文字识别 | 自然语言处理 |
|---------|------|-------|-------|--------|--------|
| 图片鉴黄    |      |       |       |        |        |
| API     | 状态   | 调用量限制 | 购买次数包 | 领取免费额度 |        |
| 图片鉴黄    | 待付费  | —     | 购买    | 立即领取   |        |
| 图片涉暴恐识别 |      |       |       |        |        |
| API     | 状态   | 调用量限制 | 购买次数包 | 领取免费额度 |        |
| 图片涉暴恐识别 | 待付费  | —     | 购买    | 立即领取   |        |
| 文本涉黄审核  |      |       |       |        |        |
| API     | 状态   | 调用量限制 | 购买次数包 | 领取免费额度 |        |
| 文本涉黄审核  | 待付费  | —     | 购买    | 立即领取   |        |
| 文本涉政审核  |      |       |       |        |        |
| API     | 状态   | 调用量限制 | 购买次数包 | 领取免费额度 |        |
| 文本涉政审核  | 待付费  | —     | 购买    | 立即领取   |        |

### 3.1.3 订单确认

在“文本内容审核”列点击“购买”，进入产品规格及数量配置页面。

文本涉黄审核

选择配置

| 规格      | 公测截止时间              | 单价    | 购买数量             |
|---------|---------------------|-------|------------------|
| 180万次   | 2020-11-30 23:59:59 | ¥0.00 | <div>- 0 +</div> |
| 720万次   | 2020-11-30 23:59:59 | ¥0.00 | <div>- 1 +</div> |
| 3600万次  | 2020-11-30 23:59:59 | ¥0.00 | <div>- 0 +</div> |
| 18000万次 | 2020-11-30 23:59:59 | ¥0.00 | <div>- 1 +</div> |
| 36000万次 | 2020-11-30 23:59:59 | ¥0.00 | <div>- 0 +</div> |

配置清单

| 规格      | 购买数量 | 价格 |
|---------|------|----|
| 720万次   | 1    | 0  |
| 18000万次 | 1    | 0  |

费用合计  
¥0元

配置费用 ¥ 0

参考价格，具体配置请以账单为准。 [了解计费详情](#)

下一步：确认配置

用户根据需要在要开通的资源包规格及数量，点击“下一步：确认配置”，进入配置确认界面。

配置

| 产品名称   | 计费方式 | 规格      | 公测截止时间              | 购买数量 | 单价    |
|--------|------|---------|---------------------|------|-------|
| 文本涉黄审核 | 次数包  | 720万次   | 2020-11-30 23:59:59 | 1    | ¥0.00 |
| 文本涉黄审核 | 次数包  | 18000万次 | 2020-11-30 23:59:59 | 1    | ¥0.00 |

协议

☐ 我已阅读并同意相关协议。 [中国电信天翼云公测服务协议（自然文本内容审核产品）](#)

配置费用 ¥ 0

参考价格，具体配置请以账单为准。 [了解计费详情](#)

上一步

立即购买

用户勾选服务协议，点击“立即购买”，进入订单支付页面。

订单管理 - 云订单 > 订单详情 > 订单支付

订单号: 20201109135454820808 产品类型: 包周期新购 创建时间: 2020-11-09 13:54:54 更新时间: 2020-11-09 13:54:54

| 产品   | 配置 | 订购数量 | 所属资源池 | 周期 | 金额 (元) |
|------|----|------|-------|----|--------|
| 内容审核 | -  | x 1  | —     | 1次 | 0.00   |
| 内容审核 | -  |      |       |    | 0.00   |

费用合计: 0.00元

郑重提示: 对云主机的使用请遵循国家相关法律法规之规定, 对于违反相关法律法规的行为, 服务商将关闭服务器, 并视情况决定是否关闭用户帐号, 停止所有服务, 不退余款。弹性IP不支持降级操作, 购买时请慎重选择。

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

2、云市场订单不能使用优惠券和代金券支付

3、订单不支持同时使用代金券和优惠券

支付方式 ☒ 预付费

账户余额支付 (元)

代金券支付 (元)

优惠券

订单费用: + 0.00元  
 账户余额支付: - 0.00元  
 代金券支付: - 0.00元  
 优惠券优惠: - 0.00元

需支付 **0.00元**

[立即支付](#)

用户确认订单费用是否准确, 费用无误点击“立即支付”。完成支付后进入“支付结果”界面。

订单管理 - 云订单 > 订单详情 > 支付结果

☒ 支付成功 [返回订单列表](#)

订单号: 20201109135454820808 产品类型: 包周期新购 创建时间: 2020-11-09 13:54:54 更新时间: 2020-11-09 13:56:08

| 产品   | 配置 | 订购数量 | 所属资源池 | 周期 | 金额 (元) |
|------|----|------|-------|----|--------|
| 内容审核 | -  | x 1  | —     | 1次 | 0.00   |
| 内容审核 | -  |      |       |    | 0.00   |

费用合计: 0.00元

订单费用: + 0.00元  
 账户余额支付: - 0.00元  
 代金券支付: - 0.00元  
**实际支付: 0.00元**

## 3.2 内容审核控制台

点击页面右上角“控制台”，进入控制中心。



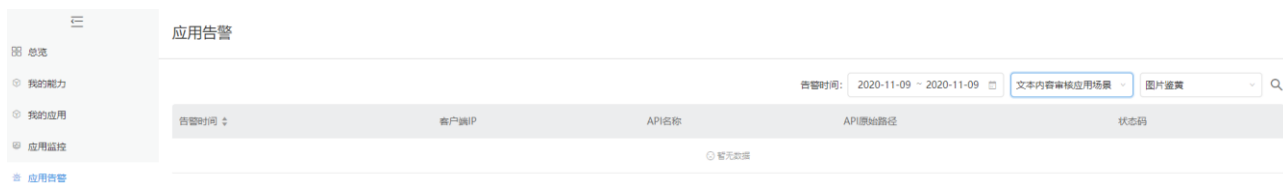
用户点击“我的应用”，可以查看已经开通的应用。



用户点击“应用监控”，可以查看所创建应用请求次数、响应时间与请求流量。



用户点击“应用告警”，可以了解应用使用过程中系统所反馈的异常情况。



# 4 API 文档

## 4.1 文本涉黄审核接口说明

### 接口描述

用于自动审核指定文本中是否存在涉黄和色情内容，并给出该文本的涉黄指数，表明文本内容的涉黄程度

### 接口要求

文本格式：

1. 单条文字数量不超过 1 万字符，批量请求单次不超过 50 篇
2. 仅支持中文文本，文本内容超过 50%非中文，返回评分 0.5

### 请求说明

#### 请求示例

HTTP 方法：POST

请求 URL：[https://ai.ctyun.cn/v1/aiop/api/\[系统生成的服务码\]/abc/api/v1/text\\_porn.json](https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/abc/api/v1/text_porn.json)

#### 请求头说明

| 参数        | 值(说明)            |
|-----------|------------------|
| Content-T | application/json |

|           |                                                                            |
|-----------|----------------------------------------------------------------------------|
| ype       |                                                                            |
| Sign      | MD5(AppKey + AppSecret + Timestamp + Path + Http Body),<br>32 位小写的 MD5 加密串 |
| Timestamp | 13 位时间戳                                                                    |
| AppKey    | 平台分配给应用独立的 AppKey                                                          |

### 请求参数

| 参数   | 是否必选 | 类型   | 可选值范围 | 说明                                      |
|------|------|------|-------|-----------------------------------------|
| data | true | list | -     | 请求参数放置于 http body 中，一段或者多段文本字符串构成的 list |

### 请求示例代码

#### Bash:

```
curl -X POST https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/abc/api/v1/text_porn.json -H "Content-Type: application/json" -H 'Sign: XXXXX' -H 'Timestamp: XXXXX' -H 'AppKey: XXXXX' --data '{"data": ["string1","string2",...]}'
```

#### Python:

```
# -*- coding: utf-8 -*-
import requests
import base64
import hashlib
import time
import json

'''
文本涉黄审核
'''
def md5(key):
    input_name = hashlib.md5()
    input_name.update(key.encode("utf-8"))
```

```

    return input_name.hexdigest()

request_url = "文本涉黄审核的 API URL"
params = {"data": ["string"]}
app_key = '平台分配的 APP Key'
app_secret = '平台分配的 APP Secret '
time_stamp = str(int(round(time.time() * 1000)))

headers = {
    'content-type': 'application/json',
    'Sign': md5(app_key + app_secret + time_stamp + '/v1/aiop/api/[系统生成的服务码]/abc/api/v1/text_porn.json' + json.dumps(params)),
    'Timestamp': time_stamp,
    'AppKey': app_key
}
response = requests.post(request_url, data=json.dumps(params), headers=headers)
if response:
    print(response.json())

```

## 返回说明

### API 调用成功返回值

| 返回字段                        | 是否<br>必选 | 类型     | 字段说明                                                                       |
|-----------------------------|----------|--------|----------------------------------------------------------------------------|
| code                        | true     | int    | 返回状态，返回 0 表示成功                                                             |
| message                     | true     | object | 请求信息结构体，message["success"]代表请求 list 中的成功数量，message["fail"]代表请求 list 中的失败数量 |
| result                      | true     | map    | 返回结果键值集合，键为场景名称"porn"；值为场景对应的文本结果，按照列表形式排列，每个元素为文本对应的鉴定结果键值对，元素排序按照传入顺序排列  |
| result['porn'][].label      | true     | int    | 文本的分类结果，0 表示正常，1 表示该场景下违规，2 表示人工审核                                         |
| result['porn'][].class_name | true     | string | 文本分类结果描述，为正常、违规或人工审核                                                       |

|                             |      |       |                                  |
|-----------------------------|------|-------|----------------------------------|
| result['porn'][].confidence | true | float | 文本违规的置信度，在 0 到 1 区间              |
| result['porn'][].err_code   | true | int   | 若该文本处理异常，则返回该文本的错误代码，参见本文档的错误码说明 |
| result['porn'][].err_msg    | true | int   | 若该文本处理异常，则返回该文本的错误消息             |

### API 调用失败返回值

| 返回字段    | 是否必选 | 类型     | 字段说明               |
|---------|------|--------|--------------------|
| code    | true | int    | 错误码，参见本文档错误码说明部分   |
| message | true | string | 返回对应的可读消息，值为 error |
| details | true | string | 错误消息               |

### 返回示例

```

HTTP/1.1 200 OK
Content-Type: application/json
{
  "code": 0,
  "message": {"success":3, "fail":0},
  "result":{
    "porn": [
      {
        "label": 0,
        "class_name": "正常",
        "confidence": 0.315978
      },
      {
        "label": 0,
        "class_name": "正常",
        "confidence": 0.000078003
      },
      {
        "label": 1,
        "class_name": "违规",
        "confidence": 0.985978
      }
    ]
  }
}

```

```

    }
  ]
}

```

## 错误码说明

| 错误代码 | 错误消息            | 错误可能原因                                   |
|------|-----------------|------------------------------------------|
| 4001 | 请求中未包含内容        | 请求的参数名不正确，参数为空、全为空格等转义符，或者参数格式、参数数据类型不正确 |
| 4007 | 请求 JSON 处理异常    | 在处理 JSON 数据的时候发生异常，可以尝试在本地进行 JSON 序列化操作  |
| 4010 | 文件大小不符合要求       | 文本字符超过了 9999 字                           |
| 4017 | 处理该请求超时         | 该请求内部处理超时，请稍后再试。                         |
| 4018 | 单次批量请求数量超过限制    | 该批次请求量超过限制，请降低单批次请求量                     |
| 4019 | 请求的 API 不存在     | 请求的 API 路径错误                             |
| 4020 | 不支持的请求方法        | 仅支持 HTTP POST 请求方法                       |
| 5000 | 服务内部错误          | 需要联系管理员处理                                |
| 5001 | 服务接口异常，请联系管理员   | 当前接口无法正常处理请求，需要联系管理员处理。                  |
| 5003 | 场景接口异常，具体参见返回消息 | 后端服务超时等                                  |

## 4.2 文本涉政审核

### 接口描述

用于自动审核指定文本中是否存在涉政敏感内容，并给出该文本的涉政指数，表明文本内容的涉政程度

### 接口要求

文本格式：

1. 单条文字数量不超过 1 万字符，批量请求单次不超过 50 篇
2. 仅支持中文文本，文本内容超过 50%非中文，返回评分 0.5

### 请求说明

#### 请求示例

HTTP 方法：POST

请求 URL： [https://ai.ctyun.cn/v1/aiop/api/\[系统生成的服务码\]/FileIdentity2/api/v1/text\\_politic.json](https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/FileIdentity2/api/v1/text_politic.json)

#### 请求头说明

| 参数           | 值(说明)                                                                      |
|--------------|----------------------------------------------------------------------------|
| Content-Type | application/json                                                           |
| Sign         | MD5(AppKey + AppSecret + Timestamp + Path + Http Body),<br>32 位小写的 MD5 加密串 |
| Timestamp    | 13 位时间戳                                                                    |
| AppKey       | 平台分配给应用独立的 AppKey                                                          |

## 请求参数

| 参数   | 是否必选 | 类型   | 可选值范围 | 说明                                       |
|------|------|------|-------|------------------------------------------|
| data | true | list | -     | 请求参数放置于 http body 中，一段或者多段文本字符串构成的 list。 |

## 请求示例代码

### Bash:

```
curl -X POST https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/FileIdentity2/api/v1/text_politic.json -H 'Content-Type: application/json' -H 'Sign: XXXXX' -H 'Timestamp: XXXXX' -H 'AppKey: XXXXX' --data '{"data": ["string1", "string2", ...]}'
```

### Python:

```
# -*- coding: utf-8 -*-
import requests
import base64
import hashlib
import time
import json

'''
文本涉政审核
'''
def md5(key):
    input_name = hashlib.md5()
    input_name.update(key.encode("utf-8"))
    return input_name.hexdigest()

request_url = "文本涉政审核的 API URL"
params = {"data": ["string"]}
app_key = '平台分配的 APP Key'
app_secret = '平台分配的 APP Secret '
time_stamp = str(int(round(time.time() * 1000)))

headers = {
    'content-type': 'application/json',
    'Sign': md5(app_key + app_secret + time_stamp + '/v1/aiop/api/[系统生成的服务码]/FileIdentity2/api/v1/text_politic.json' + json.dumps(params)),
    'Timestamp': time_stamp,
    'AppKey': app_key
```

```

}
response = requests.post(request_url, data=json.dumps(params), headers=headers)
if response:
    print(response.json())

```

## 返回说明

### API 调用成功返回值

| 返回字段                           | 是否<br>必选 | 类<br>型 | 字段说明                                                                         |
|--------------------------------|----------|--------|------------------------------------------------------------------------------|
| code                           | true     | int    | 返回状态，返回 0 表示成功                                                               |
| message                        | true     | object | 请求信息结构体，message["success"]代表请求 list 中的成功数量，message["fail"]代表请求 list 中的失败数量   |
| result                         | true     | map    | 返回结果键值集合，键为场景名称"politic"；值为场景对应的文本结果，按照列表形式排列，每个元素为文本对应的鉴定结果键值对，元素排序按照传入顺序排列 |
| result['politic'][].label      | true     | int    | 文本的分类结果，0 表示正常，1 表示该场景下违规，2 表示人工审核                                           |
| result['politic'][].class_name | true     | string | 文本分类结果描述，为正常、违规或人工审核                                                         |
| result['politic'][].confidence | true     | float  | 文本违规的置信度，在 0 到 1 区间                                                          |
| result['politic'][].err_code   | true     | int    | 若该文本处理异常，则返回该文本的错误代码                                                         |
| result['politic'][].err_msg    | true     | int    | 若该文本处理异常，则返回该文本的错误消息                                                         |

## API 调用失败返回值

| 返回字段    | 是否<br>必选 | 类型     | 字段说明                |
|---------|----------|--------|---------------------|
| code    | true     | int    | 错误码, 参见本文档错误码说明部分   |
| message | true     | string | 返回对应的可读消息, 值为 error |
| details | true     | string | 错误消息                |

## 返回示例

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "code": 0,
  "message": {"success":3, "fail":0},
  "result":{
    "politic": [
      {
        "label": 0,
        "class_name": "正常",
        "confidence": 0.315978
      },
      {
        "label": 0,
        "class_name": "正常",
        "confidence": 0.0000078003
      },
      {
        "label": 1,
        "class_name": "违规",
        "confidence": 0.985978
      }
    ]
  }
}
```

## 错误码说明

| 错误<br>代码 | 错误消息     | 错误可能原因                             |
|----------|----------|------------------------------------|
| 4001     | 请求中未包含内容 | 请求的参数名不正确, 参数为空、全为空格等转义符, 或者参数格式、参 |

|      |                 |                                         |
|------|-----------------|-----------------------------------------|
|      |                 | 数数据类型不正确                                |
| 4007 | 请求 JSON 处理异常    | 在处理 JSON 数据的时候发生异常，可以尝试在本地进行 JSON 序列化操作 |
| 4010 | 文件大小不符合要求       | 文本字符超过了 9999 字                          |
| 4017 | 处理该请求超时         | 该请求内部处理超时，请稍后再试。                        |
| 4018 | 单次批量请求数量超过限制    | 该批次请求量超过限制，请降低单批次请求量                    |
| 4019 | 请求的 API 不存在     | 请求的 API 路径错误                            |
| 4020 | 不支持的请求方法        | 仅支持 HTTP POST 请求方法                      |
| 5000 | 服务内部错误          | 需要联系管理员处理                               |
| 5001 | 服务接口异常，请联系管理员   | 当前接口无法正常处理请求，需要联系管理员处理。                 |
| 5003 | 场景接口异常，具体参见返回消息 | 后端服务超时等                                 |

### 3.3 图像涉黄审核接口说明

#### 接口描述

用于自动审核指定图片中是否存在色情内容，并给出该图片的涉黄指数，表明图片内容的涉黄程度

#### 接口要求

图片格式：

1. 图片单张大小不超过 10MB，批量请求单次不超过 50 张

2. 图片像素尺寸大于 32x32 , 小于 5000x5000
3. 图片格式支持 png、jpg、gif、jpeg、bmp、heic 格式

## 请求说明

### 请求示例

HTTP 方法: POST

请求 URL: [https://ai.ctyun.cn/v1/aiop/api/\[系统生成的服务码\]/PictureIdentity/api/v1/image\\_porn.json](https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/PictureIdentity/api/v1/image_porn.json)

### 请求头说明

| 参数           | 值(说明)                                                                      |
|--------------|----------------------------------------------------------------------------|
| Content-Type | application/json                                                           |
| Sign         | MD5(AppKey + AppSecret + Timestamp + Path + Http Body),<br>32 位小写的 MD5 加密串 |
| Timestamp    | 13 位时间戳                                                                    |
| AppKey       | 平台分配给应用独立的 AppKey                                                          |

### 请求参数

| 参数   | 是否必选 | 类型   | 可选值范围 | 说明                                                                        |
|------|------|------|-------|---------------------------------------------------------------------------|
| data | true | list | -     | 请求参数放置于 http body 中, 一张或多张图像 Base64 编码数据构成的 list。注意: 图片需要使用常规 base64 编码方式 |

### 请求示例代码

**Bash:**

```
curl -X POST https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/PictureIdentity/api/v1/image_p
orn.json -H "Content-Type: application/json" -H 'Sign: XXXXX' -H 'Timestamp: XXXXX' -H 'AppKe
y: XXXXX' --data '{"data": ["AAAAAAAAAA..."]}'
```

### Python:

```
# -*- coding: utf-8 -*-
import requests
import base64
import hashlib
import time
import json

'''
图片鉴黄
'''

def md5(key):
    input_name = hashlib.md5()
    input_name.update(key.encode("utf-8"))
    return input_name.hexdigest()

request_url = "图片鉴黄的 API URL"

# 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read()).decode()

params = {"data": [img]}
app_key = '平台分配的 APP Key'
app_secret = '平台分配的 APP Secret '
time_stamp = str(int(round(time.time() * 1000)))

headers = {
    'content-type': 'application/json',
    'Sign': md5(app_key + app_secret + time_stamp + '/v1/aiop/api/[系统生成的服务码]/PictureIdentit
y/api/v1/image_porn.json' + json.dumps(params)),
```

```

    'Timestamp': time_stamp,
    'AppKey': app_key
}
response = requests.post(request_url, data=json.dumps(params), headers=headers)
if response:
    print(response.json())

```

## 返回说明

### API 调用成功返回值

| 返回字段                        | 是否<br>必选 | 类型     | 字段说明                                                                       |
|-----------------------------|----------|--------|----------------------------------------------------------------------------|
| code                        | true     | int    | 返回状态，返回 0 表示成功                                                             |
| message                     | true     | object | 请求信息结构体，message["success"]代表请求 list 中的成功数量，message["fail"]代表请求 list 中的失败数量 |
| result                      | true     | object | 返回结果键值集合，键为场景名称"porn"；值为场景对应的图片结果，按照列表形式排列，每个元素为图片对应的鉴定结果键值对，元素排序按照传入顺序排列  |
| result['porn'][].label      | true     | int    | 图片的分类结果，0 表示正常，1 表示该场景下违规，2 表示人工审核                                         |
| result['porn'][].class_name | true     | string | 图片分类结果描述，为正常、违规或人工审核                                                       |
| result['porn'][].confidence | true     | float  | 图片违规的置信度，在 0 到 1 区间                                                        |
| result['porn'][].err_code   | true     | int    | 若该张图片处理异常，则返回该图片的错误代码，具体参见本文档错误码说明部分                                       |

|                                       |      |     |                       |
|---------------------------------------|------|-----|-----------------------|
| <code>result['porn'][].err_msg</code> | true | int | 若该张图片处理异常，则返回该图片的错误消息 |
|---------------------------------------|------|-----|-----------------------|

### API 调用失败返回值

| 返回字段                 | 是否必选 | 类型     | 字段说明               |
|----------------------|------|--------|--------------------|
| <code>code</code>    | true | int    | 错误码，参见本文档错误码说明部分   |
| <code>message</code> | true | string | 返回对应的可读消息，值为 error |
| <code>details</code> | true | string | 错误消息               |

### 返回示例

HTTP/1.1 200 OK

Content-Type: application/json

```
{
  "code": 0,
  "message": {"success": 3, "fail": 0},
  "result": {
    "porn": [
      {
        "label": 0,
        "class_name": "正常",
        "confidence": 0.315978
      },
      {
        "label": 0,
        "class_name": "正常",
        "confidence": 0.000078003
      },
      {
        "label": 1,
        "class_name": "违规",

```

```

    "confidence": 0.985978
  }
]
}
}

```

## 错误码说明

| 错误代码 | 错误消息          | 错误可能原因                                          |
|------|---------------|-------------------------------------------------|
| 4001 | 请求中未包含内容      | 请求的参数名不正确，参数为空、全为空格等转义符，或者参数格式、参数数据类型不正确        |
| 4002 | 请求文件格式不支持     | 请求的数据格式该接口暂不支持                                  |
| 4006 | base64 数据处理异常 | 客户端传递的 base64 格式在处理时发生异常，可以尝试在本地转换 base64 是否有异常 |
| 4007 | 请求 JSON 处理异常  | 在处理 JSON 数据的时候发生异常，可以尝试在本地进行 JSON 序列化操作         |
| 4008 | 图片分辨率不符合要求    | 该图片的分辨率过大或过小，目前系统可接受的尺寸在 32*32 到 5000 * 5000 之间 |
| 4009 | 图片文件大小不符合要求   | 该文件大小不符合要求，静态图片要求小于 10M，动态图片要求小于 50M            |
| 4017 | 处理该请求超时       | 该请求内部处理超时，请稍后再试。                                |
| 4018 | 单次批量请求数量超过限制  | 该批次请求量超过限制，请降低单批次请求量                            |
| 4019 | 请求的 API 不存在   | 请求的 API 路径错误                                    |
| 4020 | 不支持的请求方法      | 仅支持 HTTP POST 请求方法                              |

|      |                 |                         |
|------|-----------------|-------------------------|
| 5000 | 服务内部错误          | 需要联系管理员处理               |
| 5001 | 服务接口异常，请联系管理员   | 当前接口无法正常处理请求，需要联系管理员处理。 |
| 5003 | 场景接口异常，具体参见返回消息 | 后端服务超时等                 |

## 3.4 图像涉暴恐审核接口

### 接口描述

用于自动审核指定图片中是否存在涉暴涉恐内容，并给出该图片的涉暴涉恐指数，表明图片内容的涉暴涉恐程度

### 接口要求

图片格式：

1. 图片单张大小不超过 10MB，批量请求单次不超过 50 张
2. 图片像素尺寸大于 32x32，小于 5000x5000
3. 图片格式支持 png、jpg、gif、jpeg、bmp、heic 格式

### 请求说明

#### 请求示例

HTTP 方法：POST

请求 URL：[https://ai.ctyun.cn/v1/aiop/api/\[系统生成的服务码\]/ViolencePhoto/api/v1/image\\_violence.json](https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/ViolencePhoto/api/v1/image_violence.json)

#### 请求头说明

| 参数 | 值(说明) |
|----|-------|
|----|-------|

|              |                                                                            |
|--------------|----------------------------------------------------------------------------|
| Content-Type | application/json                                                           |
| Sign         | MD5(AppKey + AppSecret + Timestamp + Path + Http Body),<br>32 位小写的 MD5 加密串 |
| Timestamp    | 13 位时间戳                                                                    |
| AppKey       | 平台分配给应用独立的 AppKey                                                          |

### 请求参数

| 参数   | 是否必选 | 类型   | 可选值范围 | 说明                                                                      |
|------|------|------|-------|-------------------------------------------------------------------------|
| data | true | list | -     | 请求参数放置于 http body 中，一张或多张图像 Base64 编码数据构成的 list，注意：图片需要使用常规 base64 编码方式 |

### 请求示例代码

#### Bash:

```
curl -X POST https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/ViolencePhoto/api/v1/image_violence.json -H "Content-Type: application/json" -H 'Sign: XXXXX' -H 'Timestamp: XXXXX' -H 'AppKey: XXXXX' --data '{"data": ["AAAAAAAAA...."]}'
```

#### Python:

```
# -*- coding: utf-8 -*-
import requests
import base64
import hashlib
import time
import json

'''
图片涉暴恐识别
'''
```

```
def md5(key):
    input_name = hashlib.md5()
    input_name.update(key.encode("utf-8"))
    return input_name.hexdigest()

request_url = "图片涉暴恐识别的 API URL"

# 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read()).decode()

params = {"data": [img]}
app_key = '平台分配的 APP Key'
app_secret = '平台分配的 APP Secret '
time_stamp = str(int(round(time.time() * 1000)))

headers = {
    'content-type': 'application/json',
    'Sign': md5(app_key + app_secret + time_stamp + '/v1/aiop/api/[系统生成的服务码]/ViolencePh
oto/api/v1/image_violence.json' + json.dumps(params)),
    'Timestamp': time_stamp,
    'AppKey': app_key
}
response = requests.post(request_url, data=json.dumps(params), headers=headers)
if response:
    print(response.json())
```

## 返回说明

### API 调用成功返回值

| 返回字段    | 是否<br>必选 | 类<br>型     | 字段说明                                                                                    |
|---------|----------|------------|-----------------------------------------------------------------------------------------|
| code    | true     | int        | 返回状态，返回 0 表示成功                                                                          |
| message | true     | obj<br>ect | 请求信息结构体，message["suc<br>cess"]代表请求 list 中的成功数<br>量，message["fail"]代表请求 list<br>中的失败数量   |
| result  | true     | ma<br>p    | 返回结果键值集合，键为场景<br>名称"violence"；值为场景对应<br>的图片结果，按照列表形式排<br>列，每个元素为图片对应的鉴<br>定结果键值对，元素排序按照 |

|                                               |      |        | 传入顺序排列                             |
|-----------------------------------------------|------|--------|------------------------------------|
| <code>result['violence'][][label]</code>      | true | int    | 图片的分类结果，0 表示正常，1 表示该场景下违规，2 表示人工审核 |
| <code>result['violence'][][class_name]</code> | true | string | 图片分类结果描述，为正常、违规或人工审核               |
| <code>result['violence'][][confidence]</code> | true | float  | 图片违规的置信度，在 0 到 1 区间                |
| <code>result['violence'][][err_code]</code>   | true | int    | 若该张图片处理异常，则返回该图片的错误代码              |
| <code>result['violence'][][err_msg]</code>    | true | int    | 若该张图片处理异常，则返回该图片的错误消息              |

### API 调用失败返回值

| 返回字段                 | 是否必选 | 类型     | 字段说明               |
|----------------------|------|--------|--------------------|
| <code>code</code>    | true | int    | 错误码，参见本文档错误码说明部分   |
| <code>message</code> | true | string | 返回对应的可读消息，值为 error |
| <code>details</code> | true | string | 错误消息               |

### 返回示例

```

HTTP/1.1 200 OK
Content-Type: application/json
{
  "code": 0,
  "message": {"success": 3, "fail": 0},
  "result": {
    "violence": [
      {
        "label": 0,

```

```

        "class_name": "正常",
        "confidence": 0.315978
    },
    {
        "label": 0,
        "class_name": "正常",
        "confidence": 0.0000078003
    },
    {
        "label": 1,
        "class_name": "违规",
        "confidence": 0.985978
    }
]
}

```

## 错误码说明

| 错误代码 | 错误消息          | 错误可能原因                                          |
|------|---------------|-------------------------------------------------|
| 4001 | 请求中未包含内容      | 请求的参数名不正确，参数为空、全为空格等转义符，或者参数格式、参数数据类型不正确        |
| 4002 | 请求文件格式不支持     | 请求的数据格式该接口暂不支持                                  |
| 4006 | base64 数据处理异常 | 客户端传递的 base64 格式在处理时发生异常，可以尝试在本地转换 base64 是否有异常 |
| 4007 | 请求 JSON 处理异常  | 在处理 JSON 数据的时候发生异常，可以尝试在本地进行 JSON 序列化操作         |
| 4008 | 图片分辨率不符合要求    | 该图片的分辨率过大或过小，目前系统可接受的尺寸在 32*32 到 5000 * 5000 之间 |
| 4009 | 图片文件大小不符合要求   | 该文件大小不符合要求，静态图片要求小于 10M，动态图片要求小于 50M            |
| 4017 | 处理该请求超时       | 该请求内部处理超时，请稍后再试。                                |

|      |                 |                         |
|------|-----------------|-------------------------|
| 4018 | 单次批量请求数量超过限制    | 该批次请求量超过限制，请降低单批次请求量    |
| 4019 | 请求的 API 不存在     | 请求的 API 路径错误            |
| 4020 | 不支持的请求方法        | 仅支持 HTTP POST 请求方法      |
| 5000 | 服务内部错误          | 需要联系管理员处理               |
| 5001 | 服务接口异常，请联系管理员   | 当前接口无法正常处理请求，需要联系管理员处理。 |
| 5003 | 场景接口异常，具体参见返回消息 | 后端服务超时等                 |

# 5 常见问题

1. Q: 文本内容审核包含哪些 API?

A: 包括 2 个 API: 涉黄 API 涉政 API。

2. Q: 文本内容审核的 API, 主要支持哪些文本格式?

A: 'doc', 'xism', 'lrc', 'xlsb', 'ssa', 'csv', 'xltx', 'et', 'docx', 'pdf', 'xml', 'srt', 'xlsx', 'ass', 'xltm', 'pptx', 'xls', 'txt', 'idx'。

3. Q: 文本内容审核对于文本内容个格式有什么要求?

A: (1) 单条文字数量不超过 1 万字符, 批量请求单次不超过 50 篇; (2) 仅支持中文文本, 文本内容超过 50%非中文, 返回评分 0.5。

4. Q: 文本内容审核是 HTTP GET 请求还是 HTTP POST 请求?

A: HTTP POST 请求。具体可以参看 API 文档。

5. Q: 请求时图片应当放置于 http 中的哪个位置?

A: 请求图片应当放置于 http 的 body 中, 不能放置于 query 或者 header 中。

6. Q: 图像内容审核对于图片有何要求?

A: (1) 图片单张大小不超过 10MB, 批量请求单次不超过 50 张; (2) 图片像素尺寸大于 32x32, 小于 5000x5000; (3) 图片格式支持 png、jpg、gif、jpeg、bmp、heic 格式。

7. Q: 在哪里可以查询到请求 URL 中的“系统生成的服务码”? 比如文本涉黄审核接口请求 URL:

[https://ai.ctyun.cn/v1/aiop/api/\[系统生成的服务码\]/abc/api/v1/text\\_porn.json](https://ai.ctyun.cn/v1/aiop/api/[系统生成的服务码]/abc/api/v1/text_porn.json)

A: “系统生成的服务码”可在相应接口的产品详情页中获得, 以文本涉黄审核接口为例。登录“文本涉黄审核”产品详情页, 查看“能力详情”版块信息。



点击“能力详情”版块下的“API 接口”。调用地址中红框标出的内容即为“系统生成的服务码”。

