

推必安内容审核系统开发者手册

版本：V2.0.6

更新时间：2023年03月21日

统一推送工委会

目录

1. 概述

1.1 申请流程

2. 服务端接口文档

2.1 概述

2.2 接口调用凭证

2.2.1 请求地址

2.2.2 请求参数

2.2.3 签名生成方法

2.2.4 返回值

2.2.5 errCode

2.2.6 请求示例

2.2.7 返回示例

2.3 文本审核

2.3.1 请求地址

2.3.2 请求头参数

2.3.3 请求参数

2.3.4 返回值

2.3.5 errCode

2.3.6 请求示例

2.3.7 返回示例

1. 概述

文本检测主要提供高效过滤色情、广告、涉政、暴恐等多类垃圾文字及敏感词、违禁变种等文本审核检测能力。

1.1 申请流程

- 登录推必安官网(<https://tuibian.mobileservice.cn>)并申请服务；
- 通过推必安服务申请后，可获取相应接口文档、调用凭证，或者验签SDK资料包和证书等。

2. 服务端接口文档

2.1 概述

给开发者提供了服务端使用的 HTTPS API 接口。

对于文档中提供的 POST 请求, body 中的参数应以 JSON 字符串形式写入

当以 JSON 形式传递数据时候, 需要设置请求头 `"Content-Type": "application/json"`

2.2 接口调用凭证

2.2.1 请求地址

POST `https://push.tuibianchina.com/v1/token`

2.2.2 请求参数

字段	数据类型	说明
accessKeyId	string	开发者 accessKeyId, 申请通过后可在 [推必安官网-开发者信息] 页获取
signatureMethod	string	签名方法, 目前支持"HMAC-SHA256"
signatureNonce	string	随机字符串, 36个字符以内。每次请求时无论成功失败, 必须重新生成
timestamp	string	秒级时间戳, 字符串格式, 与服务器时间相差必须在5分钟以内。如: 1659943094
signature	string	签名, 签名方式请见下方

2.2.3 签名生成方法

1. 申请通过后, 在推必安官网开发者中心可获取 accessKeyId 及 accessKeySecret。
2. 将 accessKeyId、signatureMethod、signatureNonce、timestamp 四个字段按照顺序使用符号 `&` 进行拼接。
3. 使用 accessKeySecret 作为 key, 以 HMAC-SHA256 方法对第2步生成的字符串进行签名
4. 对第3步得到的结果进行base64处理, 即得到最终的signature。

例如以下伪代码:

```
const accessKeyId = "a064325ab3d64c5a98562e891c316ab6"
const accessKeySecret = "598661b6aa874a9994f9498f8e448130"
const signatureMethod = "HMAC-SHA256"
const signatureNonce = "93fea3da-7214-43c1-96ea-e37c88cd3e17"

timestamp = time.Now().Timestamp()
println(timestamp)

// 按顺序拼接字段
signedText = accessKeyId + "&" + signatureMethod + "&" +signatureNonce + "&" +
timestamp
```

```

println(signedText)

rawSign = signWithHmacSha256(signedText, accessKeySecret)

signature = base64Encode(rawSign)

println(signature)

// output:
1660200714

a064325ab3d64c5a98562e891c316ab6&HMAC-SHA256&93fea3da-7214-43c1-96ea-
e37c88cd3e17&1660200714

gYIXvSsBw9RjXH78bf/FZhpGTiQLC3UP9WhwMoJqHLc=

```

2.2.4 返回值

当服务器端正确返回时，会返回如下信息：

字段	数据类型	说明
code	int	code 200 请求成功，非 200 请求错误
msg	string	请求成功/失败提示信息
data	JSONObject	返回数据
ts	int	时间戳
requestId	string	请求 ID

data :

字段	数据类型	说明
accessToken	string	调用文本审核接口时的凭证
expiresIn	int	accessToken 有效时间，单位：秒

2.2.5 errCode

当服务器端无法正确返回时，会返回如下信息：

字段	数据类型	说明
code	int	错误码
msg	string	错误信息

具体对应关系为:

响应码	说明
40009	accessKeyId 错误
400010	签名错误
400011	accessToken 生成失败

2.2.6 请求示例

POST 请求 body

```
{
  "accessKeyId": "9ea93d3b80704168a3061xxxxxxxx",
  "signatureMethod": "HMAC-SHA256",
  "signatureNonce": "d04b5238A34e7c4d",
  "timestamp": "1660200514",
  "signature": "NJiXVZt2EnVs7k3nWKk8OqWolYehqigkA6uwM+yXqmU="
}
```

2.2.7 返回示例

正常返回

```
{
  "code": 200,
  "ts": 1659421654502,
  "requestId": "9221dbbaxxxxx",
  "msg": "ok",
  "data": {
    "accessToken":
    "IDp7WVJyJVgOUQwNbDkPAX1bCkd7LWAxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "expiresIn": 604800
  }
}
```

错误返回

```
{
  "code": 40010,
  "ts": 1659421878827,
  "requestId": "b86baf0f-9fa4-45fb-846f-eb5567dca28a",
  "msg": "INVALID SIGNATURE"
}
```

2.3 文本审核

2.3.1 请求地址

POST `https://push.tuibianchina.com/v1/audit`

2.3.2 请求头参数

字段	数据类型	说明
Audit-Access-Token	string	接口调用凭证，预授权接口返回

2.3.3 请求参数

字段	数据类型	说明
pkg	string	应用包名，可以在 [推必安官网-开发者信息] 页申请
content	string	审核文本，长度最大限制为 10000

2.3.4 返回值

当服务器端无法正确返回时，会返回如下信息：

字段	数据类型	说明
code	int	code 200 请求成功，非 200 请求错误
msg	string	本次审核结果信息：Pass, Review, Block (Pass 通过；Review 通过，建议人工审核；Block 拦截)
data	JSONObject	返回审核数据
sign	string	本次审核的签名

data :

字段	数据类型	说明
auditId	string	本次审核的唯一标识 Id
auditEngine	string	审核引擎标识符
labels	array	风险详情，无风险信息列表为空

labels:

字段	数据类型	说明
level	string	风险等级: 1, 2 为高风险等级; 3, 4 为低风险等级
reason	string	拦截原因: POLITIC (涉政), RIOT (暴恐), PORN (涉黄), CONTRABAND (违法), VULGAR (低俗)
words	string	拦截文本

2.3.5 errCode

当服务器端无法正确返回时, 会返回如下信息:

字段	数据类型	说明
code	int	错误码
msg	string	错误信息

具体对应关系为:

响应码	说明
200	审核通过
202	审核文本存在疑似敏感词 (按不通过处理)
40000	服务调用失败, 具体信息请看 msg
40001	凭证错误
40002	请求参数错误
40003	审核文本长度超过限制
40004	审核不通过

2.3.6 请求示例

```
{
  "pkg": "com.test",
  "content": "我要测试相关违禁词,我草"
}
```

2.3.7 返回示例

正常返回

```
{
  "code": 40004,
  "msg": "Block",
  "sign": "MEUCIQC40lsywwqGK/DI19y6XM9KcoMWVoToJblMZzj1fbjEdQIg*****",
  "data": {
    "auditId": "44e3d69684a348c99cbc68xxxxx",
    "auditEngine": "zx",
    "labels": [
      {
        "level": 1,
        "reason": "VULGAR",
        "words": "我草"
      }
    ],
    "remark": ""
  }
}
```

错误返回

```
{
  "code": 40000,
  "msg": "AUDIT TEXT EMPTY"
}
```