腾讯同传

快速入门

腾讯云

【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。

快速入门

最近更新时间：2024-12-31 09:53:12

登录控制台

注册并通过实名认证后，登录腾讯云控制台。如果没有账号，请参考 账号注册教程 。

开通服务

1. 在腾讯云官网顶部导航产品下面，找到人工智能与机器学习，单击**腾讯同传**。
2. 进入腾讯同传 产品介绍 页，单击**立即使用**按钮，进入 概览页面，点击**开通同声传译服务**。
3. 阅读《服务等级协议》后勾选“我已阅读并同意《服务等级协议》”，然后单击**立即开通**。

计费说明

同声传译服务当前支持后付费和预付费模式，具体可以参考 计费概述 。

使用同传服务

通过 API 3.0 Explorer 在线调用

适用对象：开发初学者，有代码编写基础人员。
此方式能够实现在线调用、签名验证、SDK 代码生成和快速检索接口等能力。
在 API 概览 选择需要调用的接口，选择点击调试进入调试页面。并填写输入参数。输入参数在 API 3.0 Explorer 界面的“文档说明”选项卡中可以查看对应接口输入参数的具体含义。

> **说明：**
> 平台将对登录用户提供临时 Access Key，以便进行调试。

同传上传音频并查询结果

最近更新时间：2024-07-08 01:25:22

我的收藏

1. 接口描述

接口请求域名： tsi.tencentcloudapi.com 。

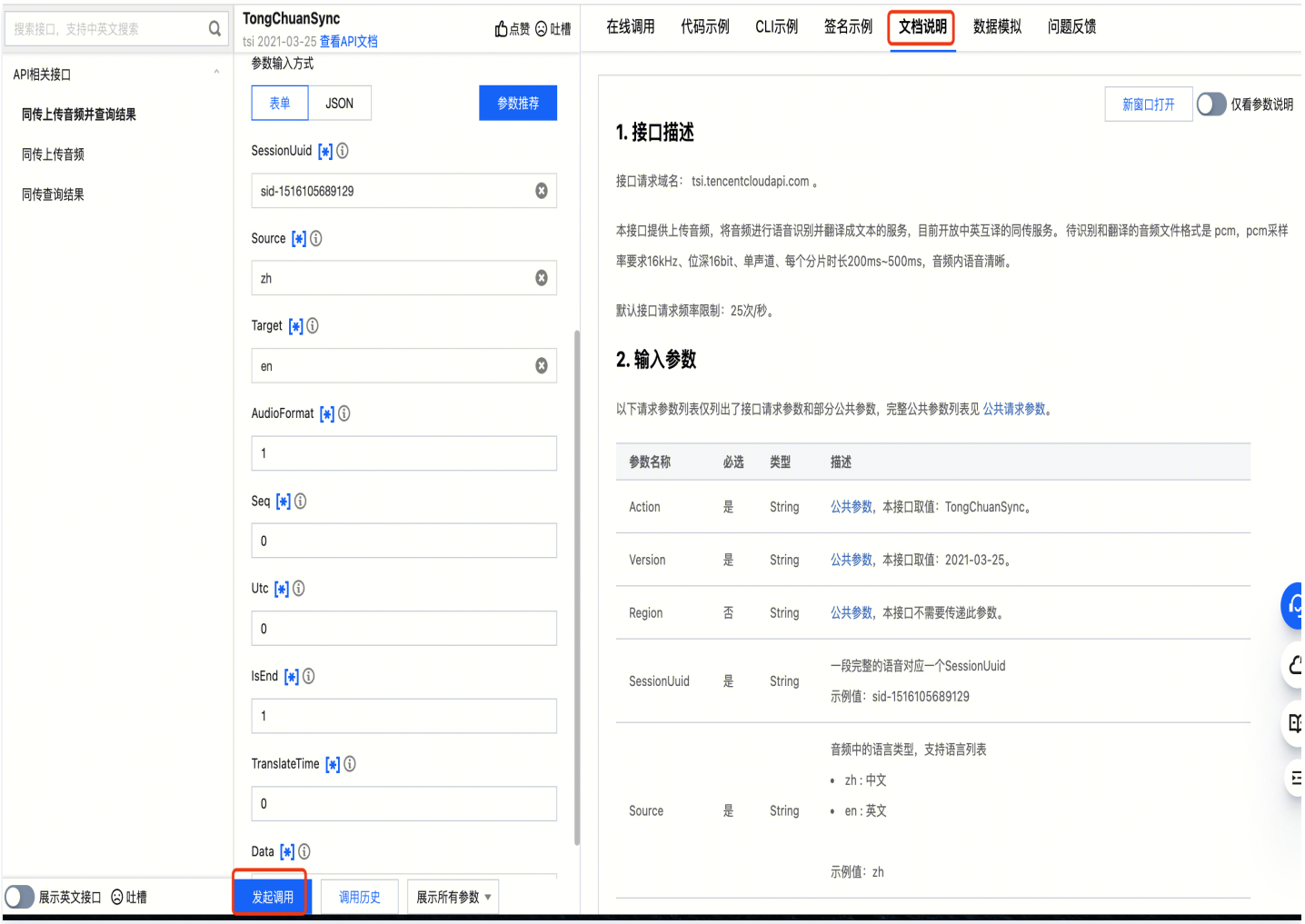
本接口提供上传音频，将音频进行语音识别并翻译成文本的服务，目前开放中英互译的同传服务。 待识别和翻译的音频文件格式是 pcm，pcm采样率要求16kHz、位深16bit、单声道、每个分片时长200ms~500ms，音频内语音清晰。

默认接口请求频率限制：25次/秒。

推荐使用 API Explorer

点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

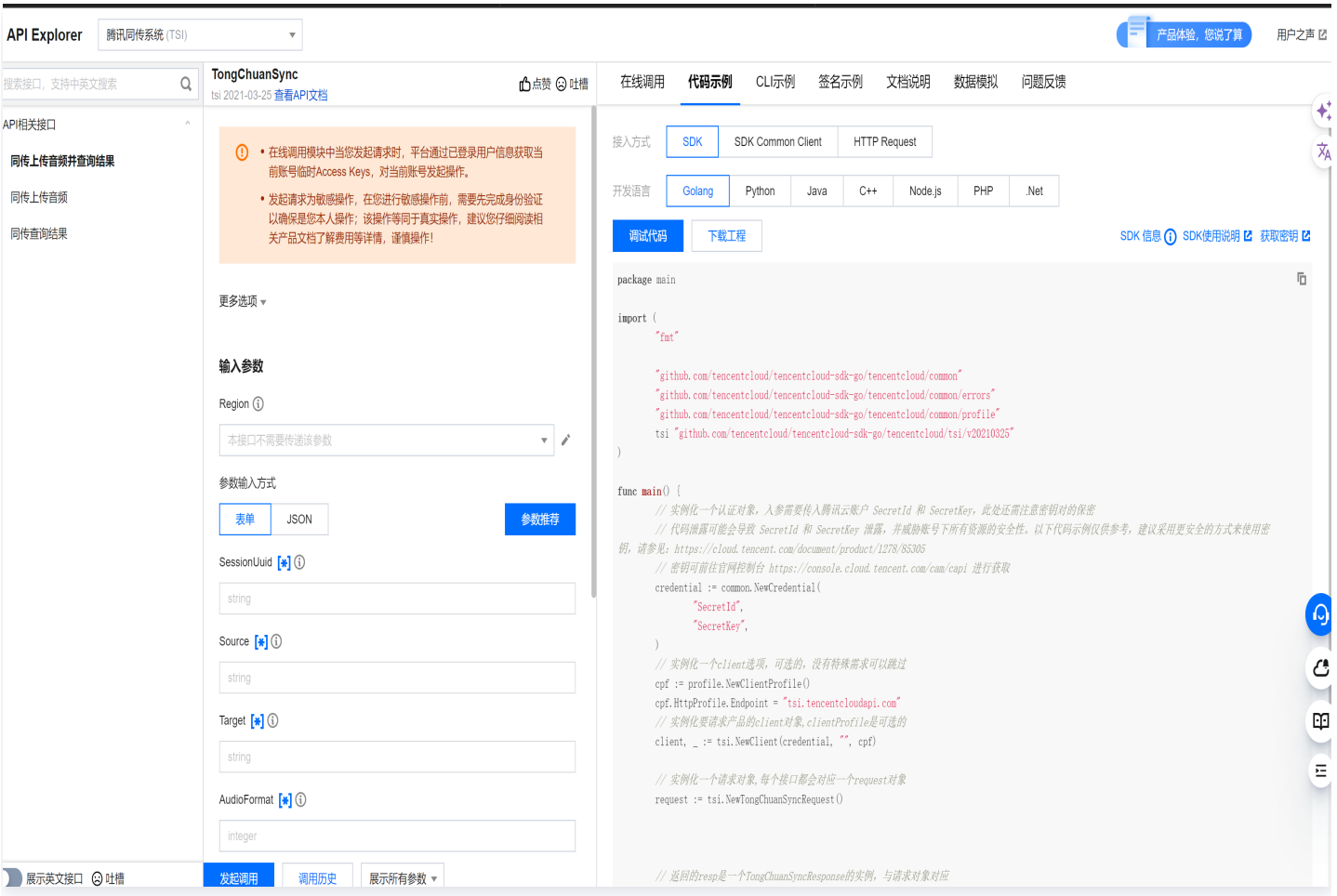


通过编写代码，调用 API 进行开发

选择需要调用的接口，在右侧选项卡中选择“代码示例”，选择接入方式和开发语言，并填写参数，就可以得到示例代码。

适用对象：开发工程师，熟悉代码编写人员。

说明：腾讯云已编写好的开发工具集（SDK），支持通过调用同传服务 API 开发功能。目前 SDK 已支持多种语言，包括 Python、Java、PHP、Go、Node.js、.Net 等，可在每个服务的文档中下载对应的 SDK。



下面以python代码为例，说明API本地调用的方法，识别传入的本地音视频文件。

python 版本 >= 3.6

```
# 按照需要的音视频处理包
pip install moviepy
pip install pydub
```

示例：

```
import hmac
import json
import sys
import time
import os
import hashlib
from datetime import datetime
if sys.version_info[0] <= 2:
    from httplib import HTTPSConnection
```

```
else:
    from http.client import HTTPSConnection


def sign(key, msg):
    return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()


def
get_TongChuan_result(SessionUuid,Source,Target,AudioFormat,Seq,Utc,IsEnd
,TranslateTime,Data):
    secret_id = "SecretId"
    secret_key = "SecretKey"
    token = ""

    service = "tsi"
    host = "tsi.tencentcloudapi.com"
    region = ""
    version = "2021-03-25"
    action = "TongChuanSync"
    payload = {
                "SessionUuid": SessionUuid,
                "Source": Source,
                "Target": Target,
                "AudioFormat": AudioFormat,
                "Seq": Seq,
                "Utc": Utc,
                "IsEnd": IsEnd,
                "TranslateTime": TranslateTime,
                "Data": Data
                }
    payload = json.dumps(payload)
    endpoint = "https://tsi.tencentcloudapi.com"
    algorithm = "TC3-HMAC-SHA256"
    timestamp = int(time.time())
    date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")

    # ************* 步骤 1：拼接规范请求串 *************
    http_request_method = "POST"
    canonical_uri = "/"
    canonical_querystring = ""
    ct = "application/json; charset=utf-8"
```

```
    canonical_headers = "content-type:%s\nhost:%s\nx-tc-action:%s\n" %
(ct, host, action.lower())
    signed_headers = "content-type;host;x-tc-action"
    hashed_request_payload = hashlib.sha256(payload.encode("utf-
8")).hexdigest()
    canonical_request = (http_request_method + "\n" +
                         canonical_uri + "\n" +
                         canonical_querystring + "\n" +
                         canonical_headers + "\n" +
                         signed_headers + "\n" +
                         hashed_request_payload)

    # ************* 步骤 2：拼接待签名字符串 *************
    credential_scope = date + "/" + 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)

    # ************* 步骤 3：计算签名 *************
    secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
    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()

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

    # ************* 步骤 5：构造并发起请求 *************
    headers = {
        "Authorization": authorization,
        "Content-Type": "application/json; charset=utf-8",
        "Host": host,
        "X-TC-Action": action,
        "X-TC-Timestamp": timestamp,
        "X-TC-Version": version
```

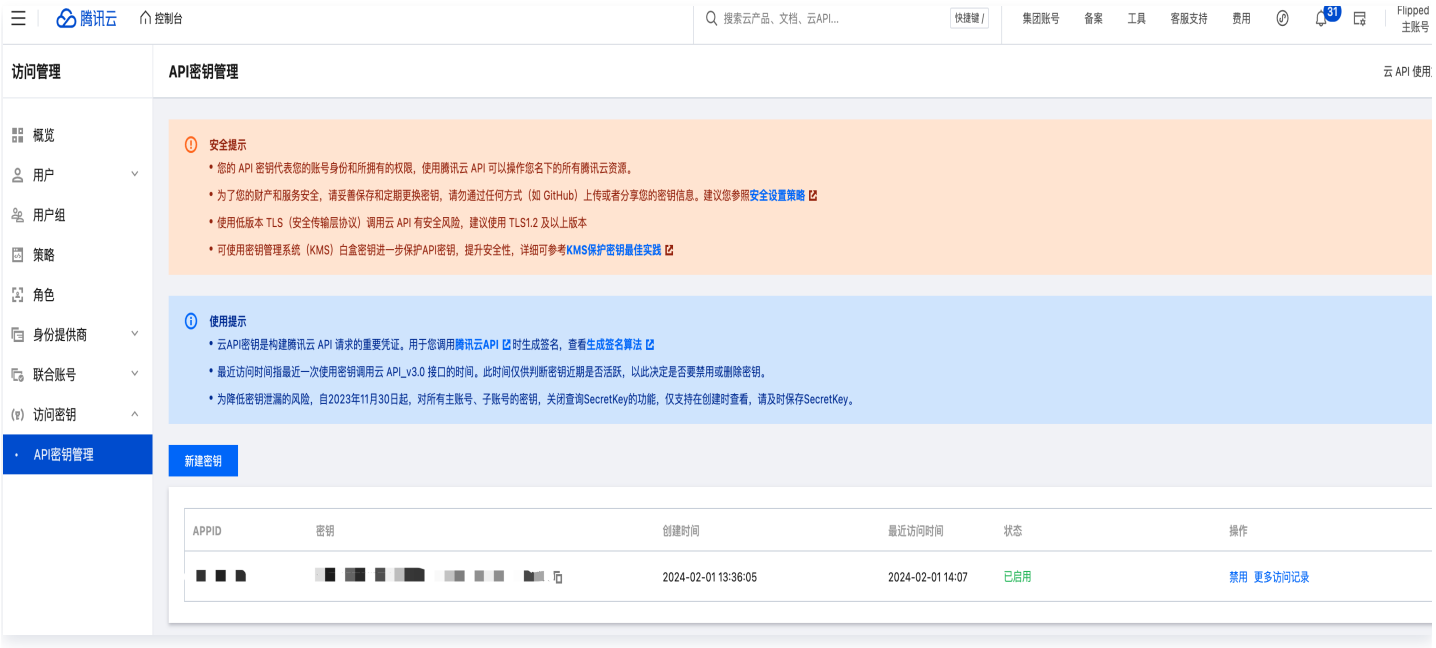
```
    }
    if region:
        headers["X-TC-Region"] = region
    if token:
        headers["X-TC-Token"] = token

    req = HTTPSConnection(host)
    req.request("POST", "/", headers=headers, body=payload.encode("utf-
8"))
    resp = req.getresponse()
    resp = json.loads(resp.read())
    return resp


def make_chunks(audio_segment, chunk_length_ms):
    """将音频分割成指定长度的片段"""
    return [audio_segment[i:i + chunk_length_ms] for i in range(0,
len(audio_segment), chunk_length_ms)]


def deal_audio(video_path, Source, Target):
    '''
        video_path: 视频文件路径
        Source: 待识别的语言
        Target: 翻译的语言
    '''
    from moviepy.editor import VideoFileClip
    from pydub import AudioSegment
    import io,base64
    # 将音视频转换为pcm并提交给腾讯云进行语音识别
    # 待识别和翻译的音频文件格式是 pcm，pcm采样率要求16kHz、位深16bit、单声道、每
个分片时长200ms~500ms
    SessionUuid = os.path.basename(video_path) + "_" +
str(int(time.time()))
    AudioFormat = 1
    TranslateTime = 1
    IsEnd = 0
    video = VideoFileClip(video_path)
    audio = video.audio
    # 将音频写入临时文件
    temp_audio_path = "temp_audio.wav"
    audio.write_audiofile(temp_audio_path, codec='pcm_s16le',
ffmpeg_params=["-ac", "1", "-ar", "16000"])
```

```
    # 使用pydub加载处理后的音频
    sound = AudioSegment.from_file(temp_audio_path, format="wav",
frame_rate=16000, channels=1, sample_width=2)
    # 分片处理
    chunk_length_ms = 500  # 可以调整为200~500ms之间
    chunks = make_chunks(sound, chunk_length_ms)
    # 处理每个音频片段
    result_list = dict()
    for i, chunk in enumerate(chunks):
        # 编码为base64
        chunk_buffer = io.BytesIO()
        chunk.export(chunk_buffer, format="wav")
        base64_data =
base64.b64encode(chunk_buffer.getvalue()).decode('utf-8')
        Seq = i
        if i == len(chunks) - 1:
            IsEnd = 1
        # start_time为当前uinx时间
        start_time = int(time.time())
        result = get_TongChuan_result(SessionUuid, Source, Target,
AudioFormat, Seq, start_time, IsEnd, TranslateTime, base64_data)
        if "Error" in result["Response"]:
            print(result["Response"]["Error"]["Message"])
            continue
        res_list = result["Response"]["List"]
        if len(res_list) > 0:
            for index,j in enumerate(res_list):
                SeId = j["SeId"]
                if SeId in result_list.keys():
                    continue
                # 展示同传结果
                SourceText = j["SourceText"]
                TargetText = j["TargetText"]
                print(SourceText)  # ASR结果
                print(TargetText)  # 翻译结果
                if j["IsEnd"] == True:
                    SeId = j["SeId"]
                    if SeId not in result_list.keys():
                        result_list[SeId] =
{"SourceText":SourceText,"TargetText":TargetText}
        time.sleep(0.5)   # 发包间隔和包体时长保持一致
    os.remove(temp_audio_path)
    return result_list
```

```
if __name__ == "__main__":
    # 本地文件路径
    path = "test.mp4"
    # 待识别和翻译的语言
    Source = "zh"
    Target = "en"
    result_list = deal_audio(path, Source, Target)
    # print(result_list)
    # 将结果写入文件
    with open("test.json","w",encoding="utf-8") as f:
        json.dump(result_list,f,ensure_ascii=False)
```

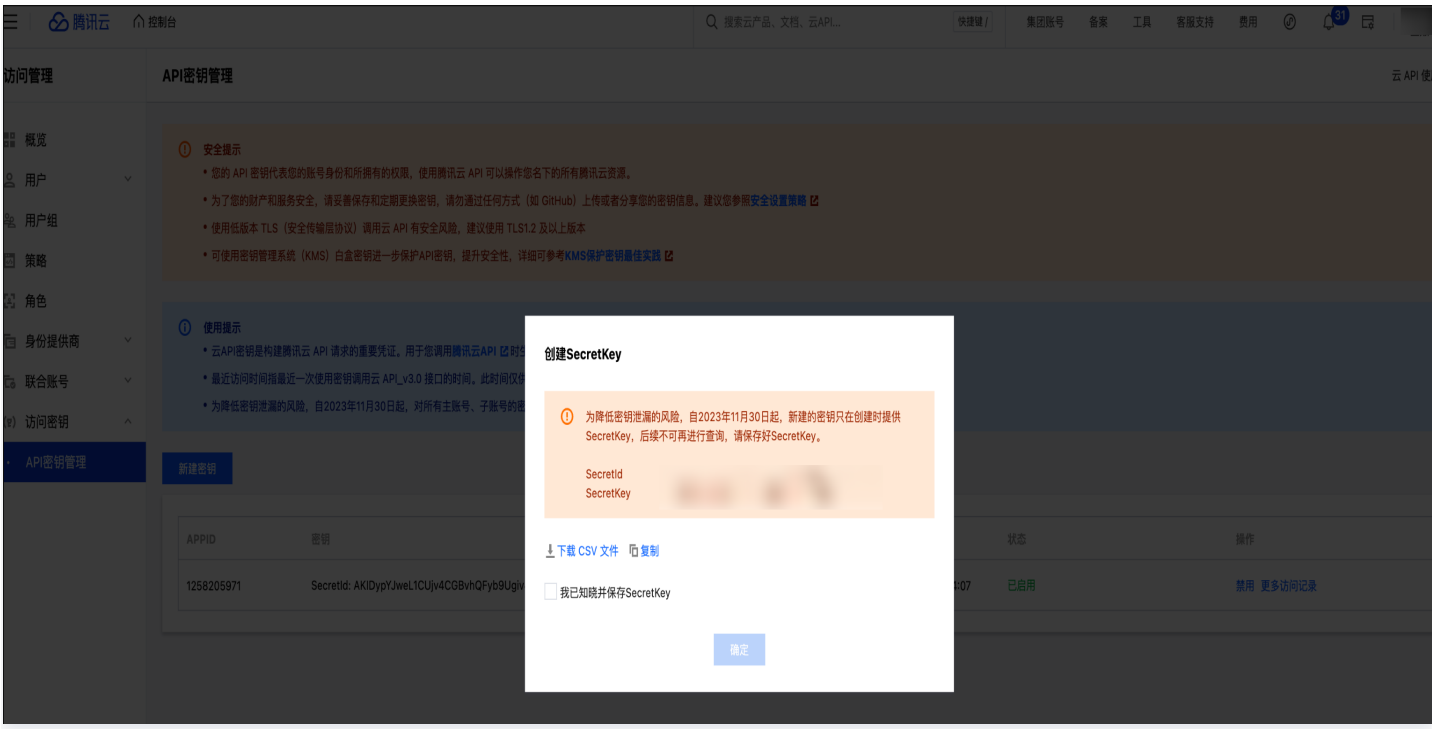
查看密钥

前往官网控制台 腾讯云控制台 API 密钥管理 获取密钥。



单击**新建密钥**按钮，弹窗查看自己的 Secretid 和 Secretkey，可单击**下载 CSV 文件**保存至本地。

> **注意：**
> 为降低密钥泄漏的风险，自2023年11月30日起，新建的密钥只在创建时提供 SecretKey，后续不可再进行查询，请保存好 SecretKey。



查看用量

登录 腾讯同传控制台，选择**用量统计**菜单，找到**同声传译**，查看服务的使用情况。