

实时语音转写与同声传译API文档

一、语音识别优势

- 支持超过100个国家语言的实时同声传译，同步输出和客户母语对应的实时同传翻译文本和语音合成数据
- 业界领先的声纹识别，返回结果字段中精准识别说话人ID
- 庞大的声纹数据库，海量音视频文件中毫秒级返回说话人的片段音视频信息
- 业界领先的降噪算法，返回的音视频文件，可包含降噪后的清晰语音
- 业界领先的高精准中英文等语音识别算法

二、实时语音转写API文档

1、接口说明

实时语音转写（Real-time ASR）基于深度全序列卷积神经网络框架，通过 WebSocket 协议，建立应用与语言转写核心引擎的长连接，开发者可实现将连续的音频流内容，实时识别返回对应的文字流内容。
支持的音频格式：采样率为16K，采样深度为16bit的pcm_s16le单声道音频

实时语言转写基本兼容科大讯飞接口，同时做了大量简化。

线上环境单个用户最大并发控制在20个以内，如果需要更大并发，请向商务申请。

2、接口Demo

目前仅提供部分开发语言的demo，其他语言请参照下方接口文档进行开发。

3、接口参数规范

集成实时语音转写API时，需按照以下要求。

内容	说明
请求协议	wss
请求地址	wss://asr-prod.abcpn.com/v1/asr/ws?{请求参数} 注：服务器IP不固定，为保证您的接口稳定，请勿通过指定IP的方式调用接口，使用域名方式调用

内容	说明
接口鉴权	签名机制，详见 signa生成
响应格式	统一采用JSON格式
开发语言	任意，只要可以向笔声云服务发起WebSocket请求的均可
音频属性	采样率16k、位长16bit、单声道
音频格式	pcm
数据发送	建议音频流每200ms发送6400字节
语言种类	中文普通话、中英混合识别、英文；俄罗斯语、阿拉巴语、法语、德语等100多种语言的语音识别另行提供。

4、接口调用流程

实时语音转写接口调用包括两个阶段：握手阶段和实时通信阶段。

(1)、握手阶段

接口地址

```
wss://asr-prod.abcpn.com/v1/asr/ws?{请求参数}
```

参数格式

key1=value1&key2=value2... (key和value都需要进行urlencode)

参数说明

参数	类型	必须	说明	示例
appid	string	是	笔声开放平台应用ID	595f23df
ts	string	是	当前时间戳, 从1970年1月1日0点0分0秒开始到现在的秒数	1512041814
signa	string	是	加密数字签名 (基于HMACSHA1算法)	lrrzsJeOfk1NGfjHW6SkHUoN9CU=
source_lang	string	否	实时语音转写语种, 也就是源语言; 不传自动感知	语种类型: 中文、中英混合识别: cn; 英文: "en" 德文: "de" 法语: "fr" 西班牙语: "es" 意大利: "it" 俄罗斯: "ru" 日语 "ja" 韩国 "ko" 更多语言参考本文附录
punc	string	否	标点过滤控制, 默认返回标点, punc=0会过滤结果中的标点	1
speaker_number	string	否	发音人个数, 可选值: 0-10, 0表示盲分	默认: 2 (适用通话时两个人对话的场景)
scene	string	否	垂直领域个性化参数: 法院: court 教育: edu 金融: finance 医疗: medical 科技: tech 运营商: isp 政府: gov 电商: ecom 军事: mil 企业: com 生活: life 汽车: car	设置示例: scene="edu" 参数scene为非必须设置, 不设置参数默认为通用

参数	类型	必须	说明	示例
audio_sample_rate	string	否	音频采样率，有"8000", "16000", 分别代表采样率是8K和16K，默认是"16000"	"16000"
asr_type	string	否	识别结果输出类型，0，输出逐字和逐句结果；1，逐句输出；2，逐字输出。如输入错误或无输出，默认为0	"0"

(2)、实时变更同声传译参数

//传输该控制命令的时候，将下面的json数据编码成字符串传输（不是二进制数据）

```
{
  "translate": {
    "src_lang": "zh",
    "tgt_lang": "en",
    "enabled": 1
  },
  "punc": {
    "enabled": 1
  },
  "scene": "court"
}
```

实时变更参数说明：

- 语言参数
 - src_lang: 源语言，如"zh", "de"等，可以输入空格字符串
 - tgt_lang: 目标语言，如"de", "ja" 等
 - 常见翻译语种：控制把源语言转换成什么类型的语言；
 - 中文: cn
 - 英文: en
 - 日语: ja
 - 韩语: ko
 - 俄语: ru
 - 法语: fr
 - 西班牙语: es
 - 越南语: vi
- enabled: 是否打开同声传译，1表示打开，0表示关闭同声传译

(2) 、signa生成

a. python示例 ()

```
import hashlib
import hmac
import time
import base64

def get_signature_flytek(ts, app_id, app_secret):
    tt = (app_id + ts).encode('utf-8')
    md5 = hashlib.md5()
    md5.update(tt)
    baseString = md5.hexdigest()
    baseString = bytes(baseString, encoding='utf-8')

    apiKey = app_secret.encode('utf-8')
    signa = hmac.new(apiKey, baseString, hashlib.sha1).digest()
    signa = base64.b64encode(signa)
    signa = str(signa, 'utf-8')
    return signa
```

b. Java示例

- Java示例(具体参考github Java目录代码)

```
// 生成握手参数
public static String getHandShakeParams(String appId, String secretKey) {
    String ts = System.currentTimeMillis() / 1000 + "";
    String signa = "";
    try {
        signa = EncryptUtil.HmacSHA1Encrypt(EncryptUtil.MD5(appId + ts),
secretKey);
        return "?appid=" + appId + "&ts=" + ts + "&signa=" +
URLLEncoder.encode(signa, "UTF-8");
    } catch (Exception e) {
        e.printStackTrace();
    }

    return "";
}
```

- Java基础工具类

```
package com.abcpen.ai.rtasr.util;

import java.io.UnsupportedEncodingException;
import java.nio.charset.StandardCharsets;
import java.security.InvalidKeyException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
```

```

import java.security.SignatureException;

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;

import org.apache.commons.codec.binary.Base64;

public class EncryptUtil {

    /**
     * 加密数字签名（基于HMACSHA1算法）
     *
     * @param encryptText
     * @param encryptKey
     * @return
     * @throws SignatureException
     */
    public static String HmacSHA1Encrypt(String encryptText, String encryptKey)
    throws SignatureException {
        byte[] rawHmac = null;
        try {
            byte[] data = encryptKey.getBytes(StandardCharsets.UTF_8);
            SecretKeySpec secretKey = new SecretKeySpec(data, "HmacSHA1");
            Mac mac = Mac.getInstance("HmacSHA1");
            mac.init(secretKey);
            byte[] text = encryptText.getBytes(StandardCharsets.UTF_8);
            rawHmac = mac.doFinal(text);
        } catch (InvalidKeyException e) {
            throw new SignatureException("InvalidKeyException:" + e.getMessage());
        } catch (NoSuchAlgorithmException e) {
            throw new SignatureException("NoSuchAlgorithmException:" +
e.getMessage());
        }
        String oauth = new String(Base64.encodeBase64(rawHmac));

        return oauth;
    }

    public final static String MD5(String pstr) {
        char[] md5String = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a',
'b', 'c', 'd', 'e', 'f'};
        try {
            byte[] btInput = pstr.getBytes();
            MessageDigest mdInst = MessageDigest.getInstance("MD5");
            mdInst.update(btInput);
            byte[] md = mdInst.digest();
            int j = md.length;
            char[] str = new char[j * 2];
            int k = 0;
            for (int i = 0; i < j; i++) { // i = 0
                byte byte0 = md[i]; // 95
                str[k++] = md5String[byte0 >>> 4 & 0xf]; // 5
            }
        }
    }
}

```

```

        str[k++] = md5String[byte0 & 0xf]; // F
    }

    return new String(str);
} catch (Exception e) {
    return null;
}
}
}

```

- 加密规则说明

1.获取baseString，baseString由app_id和当前时间戳ts拼接而成，假如app_id为595f23df，ts为1512041814，则baseString为

```
595f23df1512041814
```

2.对baseString进行MD5，假如baseString为上一步生成的595f23df1512041814，MD5之后则为

```
0829d4012497c14a30e7e72aeebe565e
```

3.以app_secret为key对MD5之后的baseString进行HmacSHA1加密，然后再对加密后的字符串进行base64编码。

假如app_secret为d9f4aa7ea6d94faca62cd88a28fd5234，MD5之后的baseString为上一步生成的0829d4012497c14a30e7e72aeebe565e，则加密之后再进行base64编码得到的signa为

```
IrrzsJeOFk1NGfJHW6SkHUoN9CU=
```

备注：

- app_secret：接口密钥，在应用中添加实时语音转写服务时自动生成，调用方注意保管；
- signa的生成公式：HmacSHA1(MD5(app_id + ts), app_secret)，具体的生成方法参考本git实例代

####请求示例

```

wss://translate.abcpn.com/v1/asr/ws?
appid=595f23df&ts=1512041814&signa=IrrzsJeOFk1NGfJHW6SkHUoN9CU=&pd=edu

```

(3)、返回值

结果格式为json，字段说明如下：

参数	类型	说明
action	string	结果标识，result:结果，error:异常
code	string	结果码(具体见 错误码)
data	string	结果数据

参数	类型	说明
desc	string	描述
sid	string	会话ID

其中sid字段主要用于DEBUG追查问题，如果出现问题，可以提供sid帮助确认问题。

成功

```
{'action': 'result', 'code': '0', 'data': {}, 'desc': 'success', 'sid': '0', 'asr': '', 'translate': {}}
```

失败

```
{
  "action": "error",
  "code": "10110",
  "data": "",
  "desc": "invalid authorization|illegal signa",
  "sid": "0"
}
```

(4)、实时通信阶段

握手成功后，进入实时通信阶段，此时客户端的主动操作有两种：上传数据和上传结束标识，被动操作有两种：接收转写结果和错误

(5)、上传数据

在实时转写过程中，客户端不断构造binary message发送到服务端，内容是音频的二进制数据。此操作的频率影响到文字结果展现的实时性。

注意：

1.建议音频流每200ms发送6400字节，发送过快可能导致引擎出错； 2.音频发送间隔超时时间为15秒，超时服务端报错并主动断开连接。

(6)、上传结束标志

音频数据上传完成后，客户端需发送一个特殊的binary message到服务端作为结束标识，内容是：

```
{"end" : true}
```

(7)、接收转写结果

交互过程中，服务端不断返回 text message（转写结果）到客户端。当所有结果发送完毕后，服务端断开连接，交互结束。

- 结果示例：


```
{'action': 'result', 'code': '0', 'data': {'cn': {'st': {'bg': '7688', 'ed': '8691', 'type': '1', 'rt': [{'w': '俄罗斯方块儿。', 'translate': {'english': 'Russian squares.'}], 'ws': [{'wb': '7688', 'we': '7796', 'cw': [{'w': '俄', 'wp': 1}]], {'wb': '7796', 'we': '7904', 'cw': [{'w': '罗', 'wp': 1}]], {'wb': '8135', 'we': '8296', 'cw': [{'w': '斯', 'wp': 1}]], {'wb': '8360', 'we': '8520', 'cw': [{'w': '方', 'wp': 1}]], {'wb': '8583', 'we': '8691', 'cw': [{'w': '块', 'wp': 1}]], {'wb': '8691', 'we': '8800', 'cw': [{'w': '儿', 'wp': 1}]]]]}}, 'seg_id': 4}, 'desc': 'success', 'sid': 'a0a44031-ac75-4929-9773-2a6bd46450e7', 'asr': '俄罗斯方块儿。', 'translate': {'english': 'Russian squares.'}}
```

- 转写结果字段说明如下：

字段	含义	描述
bg	句子在整段语音中的开始时间，单位毫秒(ms)	中间结果的bg为准确值
ed	句子在整段语音中的结束时间，单位毫秒(ms)	中间结果的ed为0
w	语音识别结果	这是实时识别的结果
asr	语音识别结果（为方便调用，我们同时将识别结果asr放在返回结构的最外层）	这是实时识别的结果
translate	同声传译的结果（为方便调用，我们同时将识别结果asr放在返回结构的最外层）	这是实时同声传译的结果
wp	词标识	n-普通词；s-顺滑词（语气词）；p-标点
wb	词在本句中的开始时间，单位是帧，1帧=10ms 即词在整段语音中的开始时间为(bg+wb*10)ms	
we	词在本句中的结束时间，单位是帧，1帧=10ms 即词在整段语音中的结束时间为(bg+we*10)ms	
type	结果类型标识	0-最终结果；1-中间结果
seg_id	转写结果序号	从0开始

(8)、接收错误信息

交互过程中，在服务端出现异常而中断服务时（如会话超时），会将异常信息以 text message 形式返回给客户端并关闭连接。

(9)、实时变更指令，实现语言的切换，或者场景的切换

变更源语言和目标语言, 和变更场景，统一使用下述指令，实时发送到现有的已有的websocket链接上（和语音数据的二进制数据不同）

```
{"config": { "lang": {"source_lang": "zh", "target_lang": "en"}, "scene": "law"}
```

- lang字典表示源语言和目标语言切换（有lang字典的时候，source_lang和target_lang必须同时存在），scene表示场景切换；
- lang和scene两者或的存在，也就是：
 - 有lang，无scene
 - 无lang，有scene
 - 有lang，有scene

5、白名单

在调用该业务接口时

- 若关闭IP白名单，接口认为IP不限，不会校验IP。
- 若打开IP白名单，则服务端会检查调用方IP是否在笔声开放平台配置的IP白名单中，对于没有配置到白名单中的IP发来的请求，服务端会拒绝服务。

IP白名单规则

- 不同Appid的不同服务都需要分别设置IP白名单；
- IP白名单需设置为外网IP，请勿设置局域网IP。
- 如果服务器返回结果如下所示(illegal client_ip)，则表示由于未配置IP白名单或配置有误，服务端拒绝服务。

```
{
  "action": "error",
  "code": "10105",
  "data": "",
  "desc": "illegal access|illegal client_ip: xx.xx.xx.xx",
  "sid": "rta..."
}
```

6、错误码

错误码	描述	说明	处理方式
0	success	成功	
-1	in progress	识别中	请继续重试
-2	audio encode error	音频编码错误	请编码成正确的格式，再提交请求
10105	illegal access	没有权限	检查apiKey，ip，ts等授权参数是否正确
10106	invalid parameter	无效参数	上传必要的参数，检查参数格式以及编码

错误码	描述	说明	处理方式
10107	illegal parameter	非法参数值	检查参数值是否超过范围或不符合要求
10109	audio url is not valid http(s) url	audio_url 不是http[s] 链接	长语音识别的时候，audio_url 必须是http[s]链接
10110	no license	无授权许可	检查参数值是否超过范围或不符合要求
10700	engine error	引擎错误	提供接口返回值，向服务提供商反馈
10701	Audio encode error, only support pcm, aac, mpeg2, opus and flac	音频编码错误	支持pcm, aac, mpeg2, opus 和 flac这几种编码，请选择其中一种
10702	Audio sample error, only support 8000、16000、44100 and 48000 Hz	音频采样率错误	支持 8000、16000、44100 和 48000 Hz，请选择其中一种
10202	websocket connect error	websocket 连接错误	检查网络是否正常
10204	websocket write error	服务端 websocket 写错误	检查网络是否正常，向服务提供商反馈
10205	websocket read error	服务端 websocket 读错误	检查网络是否正常，向服务提供商反馈
16003	basic component error	基础组件异常	重试或向服务提供商反馈
10800	over max connect limit	超过授权的连接数	确认连接数是否超过授权的连接数

7、常见问题

同声传译支持哪些国家的语言

我们支持如下国家的语言

中国
英语
德国
西班牙

俄语
韩国
法语
日本
葡萄牙
土耳其
磨光
加泰罗尼亚语
荷兰
阿拉伯语
瑞典
意大利
印度尼西亚
印度语
芬兰语
越南语
希伯来语
乌克兰
希腊
马来语
捷克
罗马尼亚语
丹麦语
匈牙利
挪威语
泰国
乌尔都语
克罗地亚语
保加利亚
立陶宛
拉丁语
毛利
马来语
威尔士
斯洛伐克语
泰卢固语
波斯
拉脱维亚语
孟加拉语
塞尔维亚
阿塞拜疆
斯洛文尼亚
卡纳达语
爱沙尼亚
马其顿
布雷顿
巴斯克语
冰岛
亚美尼亚
尼泊尔语
蒙古族
波斯尼亚

哈萨克语
阿尔巴尼亚
斯瓦希里语
加利西亚
马拉地语
旁遮普语
辛哈拉语
赫默语
约鲁巴语
索马里
南非
奥克西坦
格鲁吉亚
白俄罗斯
塔吉克语
辛迪希
古贾拉蒂语
阿姆哈拉语
伊迪什语
老挝
乌兹别克语
法罗埃
海地克里奥尔语
普什图语
土库曼
尼诺斯克
马耳他语
梵语
卢森堡语
缅甸
藏语
塔加洛语
马达加斯加
阿萨姆语
鞑靼语
夏威夷
林加拉语
豪萨
巴什基尔
爪哇语
圣代
缅甸语
巴伦西亚
佛兰芒语
海地
莱泽布格什
普什图语
旁遮普语
摩尔达维亚
摩尔多瓦
僧伽罗语
卡斯蒂利亚语

实时语音转写支持什么平台？

答：实时转写只支持webapi接口，开放平台“实时语音转写”需要WebSocket接入，针对是有编程基础的开发者用户。如果您是个人用户，不想通过编程方式直接实现语音转写功能，可以去笔声官网，了解语音转写功能的更多详情。

实时语音转写支持什么语言？

答：中文普通话、中英混合识别、英文；中英文之外的语音识别请联系商务。

支持的音频是什么格式？

答：采样率为16K，采样深度为16bit的pcm_s16le音频

实时语音转写支持的音频时长有什么限制？

答：实时语音转写可以实时识别持续的音频流，结果是实时返回，音频流长度理论上不做限制，典型的应用场景是大会或者直播的实时字幕。

实时语音转写的分片时长200ms是什么意思？

答：可以理解为上传的间隔为200ms，建议音频流每200ms向服务器发送6400字节，发过快可能导致引擎出错，音频发送间隔超时时间为15s，超时服务端报错并主动断开连接。

实时语音转写支不支持离线？

答：这个问题有点矛盾；我们有离线识别的api，请使用离线asr api完成请求服务。

实时语音转写如果一次连接使用时长超出了剩余时长怎么办？

答：首先为了使业务使用不受影响，如果在连接期间使用时长超出，转写功能并不会立刻停止。本次连接断开后时长可能会出现为负数的情况，请在使用过程中关注时长剩余情况并及时购买时长。

附录：一次完整的同声传译交互过程

```
{'action': 'started', 'code': '0', 'data': {}, 'desc': 'success', 'sid': '0', 'asr': '', 'translate': {}}
{'action': 'result', 'code': '0', 'data': {'seg_id': 1}, 'desc': 'success', 'sid': '0', 'asr': '', 'translate': {}}
{'action': 'result', 'code': '0', 'data': {'cn': {'st': {'bg': '860', 'ed': '1028', 'type': '1', 'rt': [{'w': '英特', 'translate': {'english': 'Int.'}, 'ws': [{'wb': '860', 'we': '1020', 'cw': [{'w': '英', 'wp': 1}]}], {'wb': '1028', 'we': '1188', 'cw': [{'w': '特', 'wp': 1}]}]}]}, 'seg_id': 2}, 'desc': 'success', 'sid': 'cae8bd0f-c5c1-41a5-aae1-db655ff606ee', 'asr': '英特', 'translate': {'english': 'Int.'}}
{'action': 'result', 'code': '0', 'data': {'cn': {'st': {'bg': '860', 'ed': '1136', 'type': '1', 'rt': [{'w': '英特尔', 'translate': {'english': 'Intel.'}, 'ws': [{'wb': '860', 'we': '1020', 'cw': [{'w': '英', 'wp': 1}]}], {'wb': '1028', 'we': '1136', 'cw': [{'w': '特', 'wp': 1}]}], {'wb': '1136', 'we': '1244', 'cw': [{'w': '尔', 'wp': 1}]}]}]}, 'seg_id': 3}, 'desc': 'success', 'sid': 'cae8bd0f-c5c1-41a5-aae1-db655ff606ee', 'asr': '英特尔', 'translate': {'english': 'Intel.'}}
```

```
{'action': 'result', 'code': '0', 'data': {'cn': {'st': {'bg': '860', 'ed': '1531',  
'type': '1', 'rt': [{'w': '英特尔是', 'translate': {'english': 'Intel is...'}], 'ws':  
[{'wb': '860', 'we': '1020', 'cw': [{'w': '英', 'wp': 1}]}], {'wb': '1028', 'we':  
'1136', 'cw': [{'w': '特', 'wp': 1}]}], {'wb': '1136', 'we': '1244', 'cw': [{'w':  
'尔', 'wp': 1}]}], {'wb': '1531', 'we': '1691', 'cw': [{'w': '是', 'wp': 1}]}]}]},  
'seg_id': 4}, 'desc': 'success', 'sid': 'cae8bd0f-c5c1-41a5-aae1-db655ff606ee',  
'asr': '英特尔是', 'translate': {'english': 'Intel is...'}}  
{'action': 'result', 'code': '0', 'data': {'cn': {'st': {'bg': '860', 'ed': '2203',  
'type': '1', 'rt': [{'w': '英特尔是当之无愧', 'translate': {'english': 'Intel deserves  
it.'}], 'ws': [{'wb': '860', 'we': '1020', 'cw': [{'w': '英', 'wp': 1}]}], {'wb':  
'1028', 'we': '1136', 'cw': [{'w': '特', 'wp': 1}]}], {'wb': '1136', 'we': '1244',  
'cw': [{'w': '尔', 'wp': 1}]}], {'wb': '1531', 'we': '1691', 'cw': [{'w': '是', 'wp':  
1}]}], {'wb': '1700', 'we': '1860', 'cw': [{'w': '当', 'wp': 1}]}], {'wb': '1868',  
'we': '2028', 'cw': [{'w': '之', 'wp': 1}]}], {'wb': '2036', 'we': '2196', 'cw':  
[{'w': '无', 'wp': 1}]}], {'wb': '2203', 'we': '2363', 'cw': [{'w': '愧', 'wp':  
1}]}]}]}, 'seg_id': 5}, 'desc': 'success', 'sid': 'cae8bd0f-c5c1-41a5-aae1-  
db655ff606ee', 'asr': '英特尔是当之无愧', 'translate': {'english': 'Intel deserves  
it.'}}  
{'action': 'result', 'code': '0', 'data': {'cn': {'st': {'bg': '860', 'ed': '2428',  
'type': '1', 'rt': [{'w': '英特尔是当之无愧的', 'translate': {'english': 'Intel  
deserves it.'}], 'ws': [{'wb': '860', 'we': '1020', 'cw': [{'w': '英', 'wp': 1}]}],  
{'wb': '1028', 'we': '1136', 'cw': [{'w': '特', 'wp': 1}]}], {'wb': '1136', 'we':  
'1244', 'cw': [{'w': '尔', 'wp': 1}]}], {'wb': '1531', 'we': '1691', 'cw': [{'w':  
'是', 'wp': 1}]}], {'wb': '1700', 'we': '1860', 'cw': [{'w': '当', 'wp': 1}]}], {'wb':  
'1868', 'we': '2028', 'cw': [{'w': '之', 'wp': 1}]}], {'wb': '2036', 'we': '2196',  
'cw': [{'w': '无', 'wp': 1}]}], {'wb': '2203', 'we': '2363', 'cw': [{'w': '愧', 'wp':  
1}]}], {'wb': '2428', 'we': '2587', 'cw': [{'w': '的', 'wp': 1}]}]}]}, 'seg_id': 6},  
'desc': 'success', 'sid': 'cae8bd0f-c5c1-41a5-aae1-db655ff606ee', 'asr': '英特尔是当之  
无愧 的', 'translate': {'english': 'Intel deserves it.'}}  
{'action': 'result', 'code': '0', 'data': {'cn': {'st': {'bg': '860', 'ed': '2987',  
'type': '1', 'rt': [{'w': '英特尔是当之无愧的第一代', 'translate': {'english': 'Intel is  
the first generation to be deserved.'}], 'ws': [{'wb': '860', 'we': '1020', 'cw':  
[{'w': '英', 'wp': 1}]}], {'wb': '1028', 'we': '1136', 'cw': [{'w': '特', 'wp': 1}]}],  
{'wb': '1136', 'we': '1244', 'cw': [{'w': '尔', 'wp': 1}]}], {'wb': '1531', 'we':  
'1691', 'cw': [{'w': '是', 'wp': 1}]}], {'wb': '1700', 'we': '1860', 'cw': [{'w':  
'当', 'wp': 1}]}], {'wb': '1868', 'we': '2028', 'cw': [{'w': '之', 'wp': 1}]}], {'wb':  
'2036', 'we': '2196', 'cw': [{'w': '无', 'wp': 1}]}], {'wb': '2203', 'we': '2363',  
'cw': [{'w': '愧', 'wp': 1}]}], {'wb': '2428', 'we': '2587', 'cw': [{'w': '的', 'wp':  
1}]}], {'wb': '2596', 'we': '2732', 'cw': [{'w': '第', 'wp': 1}]}], {'wb': '2732',  
'we': '2867', 'cw': [{'w': '一', 'wp': 1}]}], {'wb': '2987', 'we': '3147', 'cw':  
[{'w': '代', 'wp': 1}]}]}]}, 'seg_id': 7}, 'desc': 'success', 'sid': 'cae8bd0f-  
c5c1-41a5-aae1-db655ff606ee', 'asr': '英特尔是当之 无愧的第一代', 'translate':  
{'english': 'Intel is the first generation to be deserved.'}}
```

```
{'action': 'result', 'code': '0', 'data': {'cn': {'st': {'bg': '860', 'ed': '3212',  
'type': '1', 'rt': [{'w': '英特尔是当之无愧的第一代C', 'translate': {'english': 'Intel  
is the first generation C to be deserved.'}], 'ws': [{'wb': '860', 'we': '1020',  
'cw': [{'w': '英', 'wp': 1}]}], {'wb': '1028', 'we': '1136', 'cw': [{'w': '特', 'wp':  
1}]}], {'wb': '1136', 'we': '1244', 'cw': [{'w': '尔', 'wp': 1}]}], {'wb': '1531',  
'we': '1691', 'cw': [{'w': '是', 'wp': 1}]}], {'wb': '1700', 'we': '1860', 'cw':  
[{'w': '当', 'wp': 1}]}], {'wb': '1868', 'we': '2028', 'cw': [{'w': '之', 'wp': 1}]}],  
{'wb': '2036', 'we': '2196', 'cw': [{'w': '无', 'wp': 1}]}], {'wb': '2203', 'we':  
'2363', 'cw': [{'w': '愧', 'wp': 1}]}], {'wb': '2428', 'we': '2587', 'cw': [{'w':  
'的', 'wp': 1}]}], {'wb': '2596', 'we': '2732', 'cw': [{'w': '第', 'wp': 1}]}], {'wb':  
'2732', 'we': '2867', 'cw': [{'w': '一', 'wp': 1}]}], {'wb': '2987', 'we': '3147',  
'cw': [{'w': '代', 'wp': 1}]}], {'wb': '3212', 'we': '3372', 'cw': [{'w': 'C', 'wp':  
1}]}]}], 'seg_id': 8}, 'desc': 'success', 'sid': 'cae8bd0f-c5c1-41a5-aae1-  
db655ff606ee', 'asr': '英特尔是当之无愧的第一代C', 'translate': {'english': 'Intel is  
the first generation C to be deserved.'}}  
{'action': 'result', 'code': '0', 'data': {'cn': {'st': {'bg': '860', 'ed': '3827',  
'type': '1', 'rt': [{'w': '英特尔是当之无愧的第一代CPU制造', 'translate': {'english':  
'Intel was made by the first generation of well-deserved CPUs.'}], 'ws': [{'wb':  
'860', 'we': '1020', 'cw': [{'w': '英', 'wp': 1}]}], {'wb': '1028', 'we': '1136',  
'cw': [{'w': '特', 'wp': 1}]}], {'wb': '1136', 'we': '1244', 'cw': [{'w': '尔', 'wp':  
1}]}], {'wb': '1531', 'we': '1691', 'cw': [{'w': '是', 'wp': 1}]}], {'wb': '1700',  
'we': '1860', 'cw': [{'w': '当', 'wp': 1}]}], {'wb': '1868', 'we': '2028', 'cw':  
[{'w': '之', 'wp': 1}]}], {'wb': '2036', 'we': '2196', 'cw': [{'w': '无', 'wp': 1}]}],  
{'wb': '2203', 'we': '2363', 'cw': [{'w': '愧', 'wp': 1}]}], {'wb': '2428', 'we':  
'2587', 'cw': [{'w': '的', 'wp': 1}]}], {'wb': '2596', 'we': '2732', 'cw': [{'w':  
'第', 'wp': 1}]}], {'wb': '2732', 'we': '2867', 'cw': [{'w': '一', 'wp': 1}]}], {'wb':  
'2987', 'we': '3147', 'cw': [{'w': '代', 'wp': 1}]}], {'wb': '3436', 'we': '3596',  
'cw': [{'w': 'CPU', 'wp': 1}]}], {'wb': '3660', 'we': '3820', 'cw': [{'w': '制',  
'wp': 1}]}], {'wb': '3827', 'we': '3987', 'cw': [{'w': '造', 'wp': 1}]}]}], 'seg_id': 9}, 'desc': 'success', 'sid': 'cae8bd0f-c5c1-41a5-aae1-db655ff606ee',  
'asr': '英特尔是当之无愧的第一代CPU制造', 'translate': {'english': 'Intel was made by the  
first generation of well-deserved CPUs.'}}  
{'action': 'result', 'code': '0', 'data': {'cn': {'st': {'bg': '860', 'ed': '4052',  
'type': '1', 'rt': [{'w': '英特尔是当之无愧的第一代CPU制造商', 'translate': {'english':  
'Intel was the first well-deserved CPU manufacturer.'}], 'ws': [{'wb': '860', 'we':  
'1020', 'cw': [{'w': '英', 'wp': 1}]}], {'wb': '1028', 'we': '1136', 'cw': [{'w':  
'特', 'wp': 1}]}], {'wb': '1136', 'we': '1244', 'cw': [{'w': '尔', 'wp': 1}]}], {'wb':  
'1531', 'we': '1691', 'cw': [{'w': '是', 'wp': 1}]}], {'wb': '1700', 'we': '1860',  
'cw': [{'w': '当', 'wp': 1}]}], {'wb': '1868', 'we': '2028', 'cw': [{'w': '之', 'wp':  
1}]}], {'wb': '2036', 'we': '2196', 'cw': [{'w': '无', 'wp': 1}]}], {'wb': '2203',  
'we': '2363', 'cw': [{'w': '愧', 'wp': 1}]}], {'wb': '2428', 'we': '2587', 'cw':  
[{'w': '的', 'wp': 1}]}], {'wb': '2596', 'we': '2732', 'cw': [{'w': '第', 'wp': 1}]}],  
{'wb': '2732', 'we': '2867', 'cw': [{'w': '一', 'wp': 1}]}], {'wb': '2987', 'we':  
'3147', 'cw': [{'w': '代', 'wp': 1}]}], {'wb': '3436', 'we': '3596', 'cw': [{'w':  
'CPU', 'wp': 1}]}], {'wb': '3660', 'we': '3820', 'cw': [{'w': '制', 'wp': 1}]}],  
{'wb': '3827', 'we': '3987', 'cw': [{'w': '造', 'wp': 1}]}], {'wb': '4052', 'we':  
'4212', 'cw': [{'w': '商', 'wp': 1}]}]}], 'seg_id': 10}, 'desc': 'success', 'sid':  
'cae8bd0f-c5c1-41a5-aae1-db655ff606ee', 'asr': '英特尔是当之无愧的第一代CPU制造商',  
'translate': {'english': 'Intel was the first well-deserved CPU manufacturer.'}}
```



```
{'action': 'result', 'code': '0', 'data': {'cn': {'st': {'bg': '860', 'ed': '4052',  
'type': '1', 'rt': [{'w': '英特尔是当之无愧的第一代CPU制造商', 'translate': {'english':  
'Intel was the first well-deserved CPU manufacturer.'}], 'ws': [{'wb': '860', 'we':  
'1020', 'cw': [{'w': '英', 'wp': 1}]}], {'wb': '1028', 'we': '1136', 'cw': [{'w':  
'特', 'wp': 1}]}], {'wb': '1136', 'we': '1244', 'cw': [{'w': '尔', 'wp': 1}]}], {'wb':  
'1531', 'we': '1691', 'cw': [{'w': '是', 'wp': 1}]}], {'wb': '1700', 'we': '1860',  
'cw': [{'w': '当', 'wp': 1}]}], {'wb': '1868', 'we': '2028', 'cw': [{'w': '之', 'wp':  
1}]}], {'wb': '2036', 'we': '2196', 'cw': [{'w': '无', 'wp': 1}]}], {'wb': '2203',  
'we': '2363', 'cw': [{'w': '愧', 'wp': 1}]}], {'wb': '2428', 'we': '2587', 'cw':  
[{'w': '的', 'wp': 1}]}], {'wb': '2596', 'we': '2732', 'cw': [{'w': '第', 'wp': 1}]}],  
{'wb': '2732', 'we': '2867', 'cw': [{'w': '一', 'wp': 1}]}], {'wb': '2987', 'we':  
'3147', 'cw': [{'w': '代', 'wp': 1}]}], {'wb': '3436', 'we': '3596', 'cw': [{'w':  
'CPU', 'wp': 1}]}], {'wb': '3660', 'we': '3820', 'cw': [{'w': '制', 'wp': 1}]}],  
{'wb': '3827', 'we': '3987', 'cw': [{'w': '造', 'wp': 1}]}], {'wb': '4052', 'we':  
'4212', 'cw': [{'w': '商', 'wp': 1}]}]]]]], 'seg_id': 11, 'desc': 'success', 'sid':  
'cae8bd0f-c5c1-41a5-aae1-db655ff606ee', 'asr': '英特尔是当之无愧的第一代CPU制造商',  
'translate': {'english': 'Intel was the first well-deserved CPU manufacturer.'}}  
{'action': 'result', 'code': '0', 'data': {'cn': {'st': {'bg': '860', 'ed': '4947',  
'type': '1', 'rt': [{'w': '英特尔是当之无愧的第一代CPU制造商创造了', 'translate':  
{'english': 'Intel was created by a well-deserved first generation of CPU  
manufacturers.'}], 'ws': [{'wb': '860', 'we': '1020', 'cw': [{'w': '英', 'wp': 1}]}],  
{'wb': '1028', 'we': '1136', 'cw': [{'w': '特', 'wp': 1}]}], {'wb': '1136', 'we':  
'1244', 'cw': [{'w': '尔', 'wp': 1}]}], {'wb': '1531', 'we': '1691', 'cw': [{'w':  
'是', 'wp': 1}]}], {'wb': '1700', 'we': '1860', 'cw': [{'w': '当', 'wp': 1}]}], {'wb':  
'1868', 'we': '2028', 'cw': [{'w': '之', 'wp': 1}]}], {'wb': '2036', 'we': '2196',  
'cw': [{'w': '无', 'wp': 1}]}], {'wb': '2203', 'we': '2363', 'cw': [{'w': '愧', 'wp':  
1}]}], {'wb': '2428', 'we': '2587', 'cw': [{'w': '的', 'wp': 1}]}], {'wb': '2596',  
'we': '2732', 'cw': [{'w': '第', 'wp': 1}]}], {'wb': '2732', 'we': '2867', 'cw':  
[{'w': '一', 'wp': 1}]}], {'wb': '2987', 'we': '3147', 'cw': [{'w': '代', 'wp': 1}]}],  
{'wb': '3436', 'we': '3596', 'cw': [{'w': 'CPU', 'wp': 1}]}], {'wb': '3660', 'we':  
'3820', 'cw': [{'w': '制', 'wp': 1}]}], {'wb': '3827', 'we': '3987', 'cw': [{'w':  
'造', 'wp': 1}]}], {'wb': '4052', 'we': '4212', 'cw': [{'w': '商', 'wp': 1}]}], {'wb':  
'4612', 'we': '4772', 'cw': [{'w': '创', 'wp': 1}]}], {'wb': '4780', 'we': '4940',  
'cw': [{'w': '造', 'wp': 1}]}], {'wb': '4947', 'we': '5107', 'cw': [{'w': '了', 'wp':  
1}]}]]]]]]], 'seg_id': 12, 'desc': 'success', 'sid': 'cae8bd0f-c5c1-41a5-aae1-  
db655ff606ee', 'asr': '英特尔是当之无愧的第一代CPU制造商创造了', 'translate': {'english':  
'Intel was created by a well-deserved first generation of CPU manufacturers.'}}
```

```
{'action': 'result', 'code': '0', 'data': {'cn': {'st': {'bg': '860', 'ed': '5284',  
'type': '1', 'rt': [{'w': '英特尔是当之无愧的第一代CPU制造商创造了无数', 'translate':  
{'english': 'Intel was the first well-deserved CPU manufacturer to create so  
many.'}], 'ws': [{'wb': '860', 'we': '1020', 'cw': [{'w': '英', 'wp': 1}]], {'wb':  
'1028', 'we': '1136', 'cw': [{'w': '特', 'wp': 1}]], {'wb': '1136', 'we': '1244',  
'cw': [{'w': '尔', 'wp': 1}]], {'wb': '1531', 'we': '1691', 'cw': [{'w': '是', 'wp':  
1}]], {'wb': '1700', 'we': '1860', 'cw': [{'w': '当', 'wp': 1}]], {'wb': '1868',  
'we': '2028', 'cw': [{'w': '之', 'wp': 1}]], {'wb': '2036', 'we': '2196', 'cw':  
[{'w': '无', 'wp': 1}]], {'wb': '2203', 'we': '2363', 'cw': [{'w': '愧', 'wp': 1}]],  
{'wb': '2428', 'we': '2587', 'cw': [{'w': '的', 'wp': 1}]], {'wb': '2596', 'we':  
'2732', 'cw': [{'w': '第', 'wp': 1}]], {'wb': '2732', 'we': '2867', 'cw': [{'w':  
'一', 'wp': 1}]], {'wb': '2987', 'we': '3147', 'cw': [{'w': '代', 'wp': 1}]], {'wb':  
'3436', 'we': '3596', 'cw': [{'w': 'CPU', 'wp': 1}]], {'wb': '3660', 'we': '3820',  
'cw': [{'w': '制', 'wp': 1}]], {'wb': '3827', 'we': '3987', 'cw': [{'w': '造', 'wp':  
1}]], {'wb': '4052', 'we': '4212', 'cw': [{'w': '商', 'wp': 1}]], {'wb': '4612',  
'we': '4772', 'cw': [{'w': '创', 'wp': 1}]], {'wb': '4780', 'we': '4940', 'cw':  
[{'w': '造', 'wp': 1}]], {'wb': '4947', 'we': '5107', 'cw': [{'w': '了', 'wp': 1}]],  
{'wb': '5116', 'we': '5275', 'cw': [{'w': '无', 'wp': 1}]], {'wb': '5284', 'we':  
'5444', 'cw': [{'w': '数', 'wp': 1}]]]]]], 'seg_id': 13, 'desc': 'success', 'sid':  
'cae8bd0f-c5c1-41a5-aae1-db655ff606ee', 'asr': '英特尔是当之无愧的第一代CPU制造商创造了无  
数', 'translate': {'english': 'Intel was the first well-deserved CPU manufacturer to  
create so many.'}}
```

```
{'action': 'result', 'code': '0', 'data': {'cn': {'st': {'bg': '860', 'ed': '6124',  
'type': '1', 'rt': [{'w': '英特尔是当之无愧的第一代CPU制造商创造了无数的世界纪录。',  
'translate': {'english': 'Intel was the first generation of well-deserved CPU  
manufacturers to create countless world records.'}], 'ws': [{'wb': '860', 'we':  
'1020', 'cw': [{'w': '英', 'wp': 1}]], {'wb': '1028', 'we': '1136', 'cw': [{'w':  
'特', 'wp': 1}]], {'wb': '1136', 'we': '1244', 'cw': [{'w': '尔', 'wp': 1}]], {'wb':  
'1531', 'we': '1691', 'cw': [{'w': '是', 'wp': 1}]], {'wb': '1700', 'we': '1860',  
'cw': [{'w': '当', 'wp': 1}]], {'wb': '1868', 'we': '2028', 'cw': [{'w': '之', 'wp':  
1}]], {'wb': '2036', 'we': '2196', 'cw': [{'w': '无', 'wp': 1}]], {'wb': '2203',  
'we': '2363', 'cw': [{'w': '愧', 'wp': 1}]], {'wb': '2428', 'we': '2587', 'cw':  
[{'w': '的', 'wp': 1}]], {'wb': '2596', 'we': '2732', 'cw': [{'w': '第', 'wp': 1}]],  
{'wb': '2732', 'we': '2867', 'cw': [{'w': '一', 'wp': 1}]], {'wb': '2987', 'we':  
'3147', 'cw': [{'w': '代', 'wp': 1}]], {'wb': '3436', 'we': '3596', 'cw': [{'w':  
'CPU', 'wp': 1}]], {'wb': '3660', 'we': '3820', 'cw': [{'w': '制', 'wp': 1}]],  
{'wb': '3827', 'we': '3987', 'cw': [{'w': '造', 'wp': 1}]], {'wb': '4052', 'we':  
'4212', 'cw': [{'w': '商', 'wp': 1}]], {'wb': '4612', 'we': '4772', 'cw': [{'w':  
'创', 'wp': 1}]], {'wb': '4780', 'we': '4940', 'cw': [{'w': '造', 'wp': 1}]], {'wb':  
'4947', 'we': '5107', 'cw': [{'w': '了', 'wp': 1}]], {'wb': '5116', 'we': '5275',  
'cw': [{'w': '无', 'wp': 1}]], {'wb': '5284', 'we': '5444', 'cw': [{'w': '数', 'wp':  
1}]], {'wb': '5452', 'we': '5612', 'cw': [{'w': '的', 'wp': 1}]], {'wb': '5620',  
'we': '5780', 'cw': [{'w': '世', 'wp': 1}]], {'wb': '5787', 'we': '5947', 'cw':  
[{'w': '界', 'wp': 1}]], {'wb': '5955', 'we': '6115', 'cw': [{'w': '纪', 'wp': 1}]],  
{'wb': '6124', 'we': '6284', 'cw': [{'w': '录', 'wp': 1}]]]]]], 'seg_id': 14,  
'desc': 'success', 'sid': 'cae8bd0f-c5c1-41a5-aae1-db655ff606ee', 'asr': '英特尔是当之  
无愧的第一代CPU制造商创造了无数的世界纪录。', 'translate': {'english': 'Intel was the first  
generation of well-deserved CPU manufacturers to create countless world records.'}}
```