
语音识别云平台 HTTP 接口文档



二零一八年八月

目录

一. 文档概述	3
二. 系统架构	3
三. 用户使用流程	4
四. 协议具体交互描述	5
五. CURL DEMO 例子.....	6
六. 技术支持	8

一.文档概述

文档简介：天聪语音识别通过 REST API 的方式给开发者提供一个通用的 HTTP 接口。

支持语种：普通话。

适用范围：任意操作系统，任意编程语言，只要能对语音识别服务器发起 http 请求，均可使用本接口。

语音格式：wav（不压缩，pcm 编码）格式，采样率 8000 或 16000，16bit 采样精度的单声道语音。

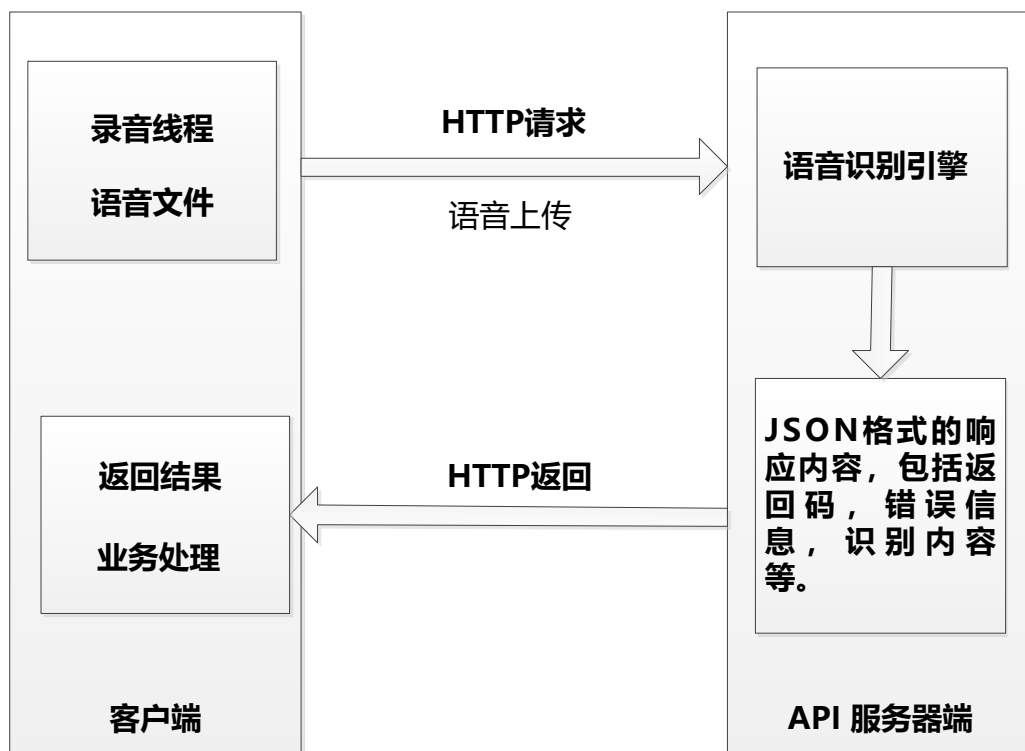
语音要求：需要上传完整的语音文件，且语音文件时长不超过 60s。

响应时间：在 E3-1220 V6 3G 主频 32G 的 win2008 64 位机器上测试，语音 4s 左右,内容 10 个字左右的语音识别时间 300 多毫秒左右。

私有化部署：系统支持 windows 64 位或者 Linux centos 6.X 64 位操作系统。支持私有云集群化部署。

版本号：V2.5.

二.系统架构



说明:

系统通过 http 的协议来进行调用。客户端采用 http post,发 post 请求到服务器,然后获取服务器的响应,根据响应的代码,判断操作是否成功。客户端负责语音的采集,并将采集后的语音上传到服务端,由服务端进行语音识别,并将结果返回到客户端。

三.用户使用流程

说明: 先申请 userid 和 token, 然后可以测试.

测试 token 可以使用您的手机号码, userid 可以使用手机号码.

假如是 8k 的语音, 使用: 您的手机号码-8K 作为 userid

服务器地址: SERVER_ADDR = "http://125.77.202.194:3998"

四.协议具体交互描述

使用 POST 方法

注册 URL /dotcasr

完整的 URL:

<http://125.77.202.194:3998/dotcasr>

传送字节流

必选字段:

userid: 用户名称,可使用用户手机号码

token:系统分配

file: 文件标识

语音缓冲(可以合并传,也可分段传,但不能有间隔符):

buffer1(录音缓冲区) +

buffer2(录音缓冲区) +

buffer3(录音缓冲区) +

...

bufferN(录音缓冲区)

识别成功服务器返回:

```
{"result": "语音识别识别内容文本", "errCode": "0"}
```

识别结果是 utf-8 编码。

识别失败服务器返回:

```
类似下面这个: {"result": "", "errCode": "-1"}
```

识别失败,会返回 errCode!=0

错误代码及常见误原因:

SUCCESS_ASR = 0 // 识别成功

FAILED_ASR = 1 // 识别错误

FAILED_UND = 2 // 理解错误

VOICE_TOO_SHORT=-1 //语音太短

VOICE_ASR_FAIL =-2 //识别过程错误.

```
ERROR_USER_ILLEGAL = 2012 // 用户名不合法
ERROR_APP_TOKEN = 2018 // 权限不合法
```

五.CURL demo 例子

```
#include <windows.h>
#include <stdio.h>
#include <time.h>
#include <stdio.h>
#include <fcntl.h>
#include <io.h>
#include <sys/timeb.h>
#include <string>
#include <process.h>
#include "curl/curl.h"
#include "curl/easy.h"

static size_t asrwritefunc(void *ptr, size_t size, size_t nmemb, int*index)
{
    int count=0;
    char result[2048];
    size_t result_len = size * nmemb;
    result[0]=0;
    if(result_len>=2000) result_len=2000;
    if(result_len>0)
    {
        memcpy(result, ptr, result_len);
        result[result_len] = '\0';
        printf("result=%s", result);
    }
    else
        printf("result_len=%d", result_len);
    return result_len;
}

int yc_asr(int index,char *audiodata,int content_len)
{
    static int boot;
    static char userid[64], token[28];
    static char ycasr_url[128];
    char tmp[1024];

    if((use_asr&16)==0) return -1;
```

```
if (boot==0)
{
    boot=1;
    GetPrivateProfileString("SET", "URL", "http://127.0.0.1:3998/doasr", ycasr_url, 28, config);

    GetPrivateProfileString("SET", "USERID", "13606060253", userid, 32, config);
    GetPrivateProfileString("SET", "TOKEN", "xxxx13606060253", token, 64, config);
}

time_t now=time(NULL);
char host[MAX_BUFFER_SIZE];
memset(host, 0, sizeof(host));
_sprintf(host, sizeof(host), "%s", ycasr_url);
printf("host:%s", host);
CURL *curl;
CURLcode res;
struct curl_httppost *post=NULL;
struct curl_httppost *last=NULL;
curl_formadd(&post, &last,
             CURLFORM_COPYNAME, "userid",
             CURLFORM_COPYCONTENTS, userid, CURLFORM_END);

curl_formadd(&post, &last,
             CURLFORM_COPYNAME, "token",
             CURLFORM_COPYCONTENTS, token, CURLFORM_END);

    curl_formadd(&post, &last,
                CURLFORM_COPYNAME, "file",
                CURLFORM_BUFFER, "upload.wav",
                CURLFORM_BUFFERPTR, audiodata,
                CURLFORM_BUFFERLENGTH, content_len,
                CURLFORM_END);

curl = curl_easy_init();
curl_easy_setopt(curl, CURLOPT_URL, host);
curl_easy_setopt(curl, CURLOPT_TIMEOUT, 30);
curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, asrwritefunc);
curl_easy_setopt(curl, CURLOPT_WRITEDATA, &index);
res = curl_easy_perform(curl);
if (res != CURLE_OK)
{
```

```
        printf("perform curl error:%d.\n", res);
        //return -1;
    }
    curl_easy_cleanup(curl);
    return 0;
}

int main()
{
    char buffer[256000];
    int c=0;
    int res=-1, size=0, index=1;

    FILE* fp1=fopen("test.wav", "r+b");
    if (fp1==NULL)
    {
        printf("open test.wav fail!");
        return -1;
    }
    else
    {
        fseek(fp1, 100, SEEK_SET); //跳过 wav 头
        while (!feof(fp1))
        {
            size=fread(buffer+c, 1, 8000, fp1);
            if (size<0) break;
            c=c+size;
            if (c>=240000) break; //最大 240K
        }
        fclose(fp1);
    }
    if (c>8000) res=yc_asr(index, buffer, c);
}
```

六.技术支持

联系电话：0592-5998812

邮箱：info@talentedsoft.com