

离线语音转写系统

java 接口层开发接入规范

北京中科信利技术有限公司
2021 年 07 月

目 录

1 规范简介.....	1
2 离线语音转写接入规范.....	2
2.1 识别开发接口.....	2
2.1.1 功能描述.....	2
2.1.2 离线语音识别接口--流式传输.....	2
2.1.3 离线语音识别接口--文件传输.....	10
2.1.4 查询/操作离线引擎信息.....	13
2.1.5 离线语音识别接口--TSocket.....	18
2.1.6 离线识别结果.....	20
2.2 json-rpc 协议错误码.....	20
2.3 错误码.....	21
3 参考文献.....	21

1 规范简介

本文档为开发者提供一个离线语音识别系统接口层开发规范，适用于民用市场通用版本。

流式传输接口：

数据发送方式：基于语音流方式，通过 **http** 协议将音频文件分包发送到离线语音识别系统进行识别；

结果获取方式：用户发送完音频数据文件后，采用轮询的方式向服务端发送请求，以请求获取识别结果。

文件式传输接口：

数据发送方式：发送语音文件的绝对路径或者 **http** 文件路径/

结果获取方式：**http** 文件路径发送方式可通过回调接口将识别结果返回，非 **http** 语音任务方式可主动将识别结果文件推送至指定服务器。

2 离线语音转写接入规范

2.1 识别开发接口

2.1.1 功能描述

➤ 离线语音识别

所有音频数据上传完成，转写成全部上传音频对应的文本

引擎含有并发校验，流式传输请在发送 Session end 接口后开启新的会话请求。

2.1.2 离线语音识别接口—流式传输

2.1.2.1 Session begin

接口地址

Nginx URL: <http://ip:port/offlineasr>

Tomcat URL: <http://ip:port/ability/offlineasr>

请求方法

POST

Content-Type application/json-rpc

Content-Length HTTP 消息实体的传输长度

Accept application/json-rpc

请求参数列表

节点深度	参数	值类型	参数含义	备注
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串“2.0”	必选
1	method	字符串	命令字，流式传输为:offline_stream	必选
1	id	整数	JsonRpc 规定参数，暂未使用	必选
1	params	JSON 对象	方法参数	必选
2	type	字符串	识别语音类型(当前引擎只支持中文普通话): 离线语音识别 xx 代表使用引擎，例如: 01:使用离线语音识别第一个离线集群 02:使用离线语音识别第二个离线集群 ...	必选
2	cmd	字符串	值为 ssb，代表会话开始	必选
2	postfix	字符串	返回结果格式(默认为.xml)	可选

			Q1.xml Q1 格式 xml 文件 sent.txt 识别结果文件 itn.txt 转阿拉伯之后识别结果文件 punc.txt 加标点之后识别结果文件 sentItnPunc_result.txt 转阿拉伯加 标点之后识别结果文件 emotionItnPunc_result.txt 转阿拉伯加 标点之后情绪文件	
2	user	字符串	鉴权用户名	可选
2	token	字符串	鉴权密钥	可选

请求参数

```

{
    "jsonrpc": "2.0",
    "method": "offline_stream",
    "id": 1,
    "params": {
        "type": "01",
        "cmd": "ssb",
        "postfix": "sent.txt",
        "user": "xxx",
        "token": "xxx"
    }
}

```

正确返回参数列表

节点深度	参数	值类型	参数含义
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"
1	id	整数	JsonRpc 规定参数，暂未使用
1	result	JSON 对象	正确返回参数
2	ret	整数	接口返回值，0 表示正常返回
2	sid	字符串	请求的唯一标识号

正确示例

```

{
    "jsonrpc": "2.0",
    "id": 1,
    "result": {
        "ret": 0,
        "sid": "4d5660000000"
    }
}

```

错误返回参数列表

节点深度	参数	值类型	参数含义
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"
1	id	整数	JsonRpc 规定参数，暂未使用
1	error	JSON 对象	错误返回参数
2	code	整数	jsonRpc 错误码
2	message	字符串	jsonRpc 错误消息
2	data	JSON 对象	错误返回参数
3	errno	整数	错误码，见 2.3
3	errmsg	字符串	错误说明

错误示例

```

{
  "jsonrpc": "2.0",
  "id": 1,
  "error": {
    "code": -34000,
    "message": "server error",
    "data": {
      "errno": -32010,
      "errmsg": "params cmd must be required!"
    }
  }
}

```

2.1.2.2 Audio write

接口地址

Nginx URL: <http://ip:port/offlineasr>

Tomcat URL: <http://ip:port/ability/offlineasr>

请求方法

POST

Content-Type application/json-rpc

Content-Length HTTP 消息实体的传输长度

Accept application/json-rpc

请求参数列表

节点深度	参数	值类型	参数含义	备注
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"	必选
1	method	字符串	命令字，流式传输为:offline_stream	必选
1	id	整数	与请求的唯一标示号保持一致	必选
1	params	JSON 对象	方法参数	必选
2	cmd	字符串	值为 auw，代表音频写入	必选
2	sid	字符串	请求的唯一标识号:session_begin 接口返回的 sid	必选
2	data	字符串	音频数据: base64 编码的音频数据流	必选

			建议每包音频数据编码前大小低于 1M	
2	audioStatus	整数	音频状态: 1: 表示第一块音频数据 2: 表示中间音频数据 4: 最后一块音频数据。	必选
2	postfix	字符串	音频后缀名(默认为 pcm)	必选
2	channels	字符串	声道数:(channels/sterro_on 不传默认为单声道) 0: 单声道 1: 双声道	可选
2	stereo_on	字符串	双声道类型:(左右声道内容是否一致,单声道默认为 4): 3: 不一致 4: 一致	可选
2	urgent	整数	任务优先级:默认为 1, 可通过配置增加优先级(1 为最低)	可选

请求参数

```
{
  "jsonrpc": "2.0",
  "method": "offline_stream",
  "id": 1,
  "params": {
    "cmd": "auw",
    "sid": "4d5660000000",
    "type": "1",
    "data": "base64 编码的音频数据",
    "audioStatus": 1,
    "postfix": "mp3",
    "channels": "0",
    "stereo_on": "4",
    "urgent": 1
  }
}
```

正确返回参数列表

节点深度	参数	值类型	参数含义
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"
1	id	整数	JsonRpc 规定参数, 暂未使用
1	result	JSON 对象	正确返回参数
2	msg	字符串	返回消息内容

正确示例

```
{
  "jsonrpc": "2.0",
```

```

    "id": 1,
    "result": {
        "msg": "data is accepted!"
    }
}

```

错误返回参数列表

节点深度	参数	值类型	参数含义
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"
1	id	整数	JsonRpc 规定参数，暂未使用
1	error	JSON 对象	错误返回参数
2	code	整数	jsonRpc 错误码
2	message	字符串	jsonRpc 错误消息
2	data	JSON 对象	错误返回参数
3	errno	整数	错误码，见 2.3
3	errmsg	字符串	错误说明

错误示例

```

{
    "jsonrpc": "2.0",
    "id": 1,
    "error": {
        "code": -34000,
        "message": "server error",
        "data": {
            "errno": -32010,
            "errmsg": "params cmd must be required!"
        }
    }
}

```

2.1.2.3 get result

接口地址

Nginx URL: <http://ip:port/offlineasr>

Tomcat URL: <http://ip:port/ability/offlineasr>

请求方法

POST

Content-Type application/json-rpc

Content-Length HTTP 消息实体的传输长度

Accept application/json-rpc

请求参数列表

节点深度	参数	值类型	参数含义	备注
------	----	-----	------	----

1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"	必选
1	method	字符串	命令字，流式传输为:offline_stream	必选
1	id	整数	与请求的唯一标示号保持一致	必选
1	params	JSON 对象	方法参数	必选
2	cmd	字符串	值为 grs，代表获取结果	必选
2	sid	字符串	请求的唯一标识号:session_begin 接口返回的 sid。	必选

请求参数

```
{
  "jsonrpc": "2.0",
  "method": "offline_stream",
  "id": 1,
  "params": {
    "cmd": "grs",
    "sid": "4d5660000000",
    "type": "1"
  }
}
```

正确返回参数列表

节点深度	参数	值类型	参数含义
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"
1	id	整数	JsonRpc 规定参数，暂未使用
1	result	JSON 对象	正确返回参数
2	ret	整数	接口返回值，0 表示正常返回
2	sid	字符串	请求的唯一标识号
2	recStatus	整数	离线语音识别结果状态: 2: 正在获取识别结果 5: 整个会话识别结果获取完成
2	result	字符串	识别结果

正确示例:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "ret": 0,
    "sid": "4d5660000000",
    "recStatus": 5,
    "result": "xml 文件内容"
  }
}
```

错误返回参数列表

节点深度	参数	值类型	参数含义
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"
1	id	整数	JsonRpc 规定参数，暂未使用
1	error	JSON 对象	错误返回参数
2	code	整数	jsonRpc 错误码
2	message	字符串	jsonRpc 错误消息
2	data	JSON 对象	错误返回参数
3	errno	整数	错误码，见 2.3
3	errmsg	字符串	错误说明

错误示例

```

{
  "jsonrpc": "2.0",
  "id": 1,
  "error": {
    "code": -34000,
    "message": "server error",
    "data": {
      "errno": -32010,
      "errmsg": "params cmd must be required!"
    }
  }
}

```

2.1.2.4 session end

接口地址

Nginx URL: <http://ip:port/offlineasr>

Tomcat URL: <http://ip:port/ability/offlineasr>

请求方法

POST

Content-Type application/json-rpc

Content-Length HTTP 消息实体的传输长度

Accept application/json-rpc

请求参数列表

节点深度	参数	值类型	参数含义	备注
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"	必选
1	method	字符串	命令字，流式传输为:offline_stream	必选
1	id	整数	与请求的唯一标示号保持一致	必选
1	params	JSON 对象	方法参数	必选
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"	必选
2	cmd	命令字	值为 sse，代表会话结束	必选
2	sid	会话 id	字符串	必选

请求参数

```
{
    "jsonrpc": "2.0",
    "method": "offline_stream",
    "id": 1,
    "params": {
        "cmd": "sse",
        "sid": "4d5660000000"
    }
}
```

正确返回参数列表

节点深度	参数	值类型	参数含义
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"
1	id	整数	JsonRpc 规定参数，暂未使用
1	result	JSON 对象	正确返回参数
2	ret	整数	接口返回值，0 表示正常返回
2	sid	字符串	请求的唯一标识号

正确示例

```
{
    "jsonrpc": "2.0",
    "id": 1,
    "result": {
        "ret": 0,
        "sid": "4d5660000000"
    }
}
```

错误返回参数列表

节点深度	参数	值类型	参数含义
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"
1	id	整数	JsonRpc 规定参数，暂未使用
1	error	JSON 对象	错误返回参数
2	code	整数	jsonRpc 错误码
2	message	字符串	jsonRpc 错误消息
2	data	JSON 对象	错误返回参数
3	errno	整数	错误码，见 2.3
3	errmsg	字符串	错误说明

错误示例

```
{
    "jsonrpc": "2.0",
    "id": 1,
    "error": {
        "code": -34000,

```

```

    "message": "server error",
    "data": {
        "errno": -32010,
        "errmsg": "params cmd must be required!"
    }
}

```

2.1.3 离线语音识别接口--文件传输

接口地址

Nginx URL: <http://ip:port/offlineasr>

Tomcat URL: <http://ip:port/ability/offlineasr>

请求方法

POST

Content-Type application/json-rpc

Content-Length HTTP 消息实体的传输长度

Accept application/json-rpc

2.1.3.1 send url

识别任务请求参数列表

节点深度	参数	值类型	参数含义	备注
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串“2.0”	必选
1	method	字符串	命令字，流式传输为:offline_file	必选
1	id	整数	JsonRpc 规定参数，暂未使用	必选
1	params	JSON 对象	方法参数	必选
2	type	字符串	识别语音类型(当前引擎只支持中文普通话): 离线语音识别 xx 代表使用引擎，例如: 01:使用离线语音识别第一个离线集群 02:使用离线语音识别第二个离线集群 ...	必选
2	sid	字符串	请求的唯一标识号,若为空则由引擎生成	可选
2	postfix	字符串	返回结果格式(默认为.xml) Q1.xml Q1 格式 xml 文件 sent.txt 识别结果文件 itn.txt 转阿拉伯之后识别结果文件 punc.txt 加标点之后识别结果文件 sentItnPunc_result.txt 转阿拉伯加 标点之后识别结果文件	可选

			emotionItnPunc_result.txt 转阿拉伯加标点之后情绪文件	
2	user	字符串	鉴权用户名	可选
2	token	字符串	鉴权密钥	可选
2	voiceFilePath	字符串	语音地址,例如: /home/user/audio/123.wav http://127.0.0.1/audio/123.wav	必选
2	sttFilePath	字符串	下发非 HTTP 语音任务时必选: 语音处理结果文件地址(绝对路径)	可选
2	callBackPath	字符串	结果回调地址(Post 请求)	必选
2	serverIp	字符串	下发非 HTTP 语音任务时必选: 文件服务器地址	可选
2	channels	字符串	声道数:(channels/sterro_on 不传默认为单声道) 0: 单声道 1: 双声道	可选
2	stereo_on	字符串	双声道类型:(左右声道内容是否一致,单声道默认为 4): 3: 不一致 4: 一致	可选
2	urgent	整数	任务优先级:默认为 1,可通过配置增加优先级(1 为最低)	整数

http 语音任务请求参数

```
{
  "jsonrpc": "2.0",
  "method": "offline_file",
  "params": {
    "type": 1,
    "sid": "xxx",
    "postfix": "sent.txt",
    "user": "xxx",
    "postfix": "xxx",
    "voiceFilePath": "http://127.0.0.1/audio/123.wav",
    "callBackPath": "http://127.0.01:8080/ability/callback",
    "channels": "0",
    "stereo_on": "4",
    "urgent": 1
  },
  "id": "1"
}
```

非 http 语音任务请求参数

```
{
  "jsonrpc": "2.0",
  "method": "offline_file",

```

```

"params": {
    "type": 1,
    "sid": "xxx",
    "postfix": "sent.txt",
    "user": "xxx",
    "postfix": "xxx",
    "voiceFilePath": " /home/user/audio/123.wav ",
    "sttFilePath ": " /home/user/audioXml/123.xml ",
    "serverIp ": "127.0.0.1 ",
    "callBackPath": "http://127.0.0.1:8080/ability/ callback",
    "channels": "0",
    "stereo_on": "4",
    "urgent": 1
},
"id": "1"}

```

2.1.3.2callback result

回调任务正确返回参数列表

节点深度	参数	值类型	参数含义
1	rtnCode	整数	返回码，正确返回码为 200
1	rtnMsg	字符串	返回信息： HTTP 语音任务返回信息为识别结果文件内容，例如 xml 文件内容 非 HTTP 语音任务返回信息为 SUCCESS
1	sid	字符串	请求的唯一标识号
1	voiceFilePath	字符串	识别任务语音地址

HTTP 语音任务正确示例

```

{
    "rtnCode": 200,
    "rtnMsg": "识别结果文件内容",
    "sid": "872a8fb2ecee47bfbc617fa8d23d93c0",
    "voiceFilePath": "http://127.0.0.1/audio/123.wav"
}

```

非 HTTP 语音任务正确示例

```

{
    "rtnCode": 200,
    "rtnMsg": "SUCCESS",
    "sid": "872a8fb2ecee47bfbc617fa8d23d93c0",
    "voiceFilePath": "/home/user /audio/123.wav"
}

```

回调任务错误返回参数列表

节点深度	参数	值类型	参数含义
1	rtnCode	整数	返回码，错误返回码见 2.3
1	rtnMsg	字符串	返回信息
1	sid	字符串	请求的唯一标识号
1	voiceFilePath	字符串	识别语音地址

HTTP 语音任务错误示例

```
{
  "rtnCode": -32036,
  "rtnMsg": "The voice file does not exist! ",
  "sid": "872a8fb2ecee47bfb617fa8d23d93c0",
  "voiceFilePath ": " http://127.0.0.1/audio/123.wav "
}
```

非 HTTP 语音任务错误示例

```
{
  "rtnCode": -32036,
  "rtnMsg": "The voice file does not exist! ",
  "sid": "872a8fb2ecee47bfb617fa8d23d93c0",
  "voiceFilePath ": "/home/user /audio/123.wav"
}
```

2.1.4 查询/操作离线引擎信息

2.1.4.1 setup_thread

接口地址

Nginx URL: <http://ip:port/offlineasr>

Tomcat URL: <http://ip:port/ability/offlineasr>

请求方法

POST

Content-Type application/json-rpc

Content-Length HTTP 消息实体的传输长度

Accept application/json-rpc

请求参数列表

节点深度	参数	值类型	参数含义	备注
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"	必选
1	method	字符串	命令字，流式传输为:setup_thread	必选
1	id	整数	与请求的唯一标示号保持一致	必选
1	params	JSON 对象	方法参数	必选
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"	必选

2	type	字符串	线程操作类型 1: 查询线程数(当 machinfo 未空时, 返回所有引擎线程数, 不为空返回指定引擎线程数) 2: 设置线程数(machinfo 与 threadnum 不可为空)	必选
2	machinfo	字符串	引擎地址	必选
2	threadNum	字符串	要设置的线程数	必选

请求参数

查询线程数:

```
{
  "jsonrpc": "2.0",
  "method": "setup_thread",
  "id": 1,
  "params": {
    "type": "1",
    "machinfo": "127.0.0.1"//可为空
  }
}
```

设置线程数:

```
{
  "jsonrpc": "2.0",
  "method": "setup_thread",
  "id": 1,
  "params": {
    "type": "2",
    "machinfo": "127.0.0.1",
    "threadNum": "10"
  }
}
```

正确返回参数列表

节点深度	参数	值类型	参数含义
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"
1	id	整数	JsonRpc 规定参数, 暂未使用
1	result	JSON 对象	正确返回参数
2	ret	整数	接口返回值, 0 表示正常返回
2	msg	Object	返回值 当为查询线程数返回值时, 含义为:

			当前线程数:最大线程数 Ps: 当前线程数初始值与最大线程数相同
<p>正确示例</p> <p>查询线程数(machinfo 不为空):</p> <pre>{ "jsonrpc": "2.0", "id": "1", "result": { "ret": 0, "msg": "10:10" } }</pre> <p>查询线程数(machinfo 为空):</p> <pre>{ "jsonrpc": "2.0", "id": "1", "result": { "ret": 0, "msg": [{"127.0.0.1": "10:10"}, {"127.0.0.2": "5:10"}] } }</pre> <p>设置线程数:</p> <pre>{ "jsonrpc": "2.0", "id": "1", "result": { "ret": 0, "msg": "Set decoder thread count successfully" } }</pre>			

错误返回参数列表

节点深度	参数	值类型	参数含义
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"
1	id	整数	JsonRpc 规定参数, 暂未使用
1	error	JSON 对象	错误返回参数
2	code	整数	jsonRpc 错误码
2	message	字符串	jsonRpc 错误消息
2	data	JSON 对象	错误返回参数
3	errno	整数	错误码, 见 2.3
3	errmsg	字符串	错误说明

错误示例

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "error": {
```

```

    "code": -34000,
    "message": "server error",
    "data": {
        "errno": -32010,
        "errmsg": "params cmd must be required!"
    }
}

```

2.1.4.2 query state

接口地址

Nginx URL: <http://ip:port/offlineasr>

Tomcat URL: <http://ip:port/ability/offlineasr>

请求方法

POST

Content-Type application/json-rpc

Content-Length HTTP 消息实体的传输长度

Accept application/json-rpc

请求参数列表

节点深度	参数	值类型	参数含义	备注
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"	必选
1	method	字符串	命令字, 流式传输为:query_state	必选
1	id	整数	与请求的唯一标示号保持一致	必选
1	params	JSON 对象	方法参数	必选
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"	必选
2	branch	整数	调用类型: 1:查询待转写任务数量 2:查询正在转写任务数量 3:查询单条任务的处理状态	必选
2	type	字符串	离线语音识别 xx 代表使用引擎集群, 例如: 01:使用离线语音识别第一个离线集群 02:使用离线语音识别第二个离线集群 ... ps:当 branch 为 1, 默认为查询 01 集群	可选
2	machinfo	字符串	引擎地址(查询具体引擎并发)	可选
2	sid	字符串	任务请求标识(查询任务处理状态时使用)	可选

请求参数

查询待转写任务数量:

```

{
    "jsonrpc": "2.0",
    "method": "query_state",

```

```

    "id": 1,
    "params": {
        "branch": "1",
        "type": "1" //为空默认查询 01 集群
    }
}

```

查询正在转写任务数量:

```

{
    "jsonrpc": "2.0",
    "method": "query_state",
    "id": 1,
    "params": {
        "branch": "2",
        "machinfo": "127.0.0.1" //为空默认查询全部引擎
    }
}

```

查询单条任务处理状态:

```

{
    "jsonrpc": "2.0",
    "method": "query_state",
    "id": 1,
    "params": {
        "branch": "3",
        "sid": "xxx"
    }
}

```

正确返回参数列表

节点深度	参数	值类型	参数含义
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"
1	id	整数	JsonRpc 规定参数，暂未使用
1	result	JSON 对象	正确返回参数
2	ret	整数	接口返回值，0 表示正常返回
2	msg	整数	返回值

			当为查询任务数量时，返回值为对应任务数量 当为查询处理状态时，返回值释义如下： 0:正在排队中 1:正在处理中 2:处理完成 3:处理失败
正确示例 <pre>{ "jsonrpc": "2.0", "id": "1", "result": { "ret": 0, "msg": 2 } }</pre>			

错误返回参数列表

节点深度	参数	值类型	参数含义
1	jsonrpc	字符串	JsonRpc 版本号 固定值字符串"2.0"
1	id	整数	JsonRpc 规定参数，暂未使用
1	error	JSON 对象	错误返回参数
2	code	整数	jsonRpc 错误码
2	message	字符串	jsonRpc 错误消息
2	data	JSON 对象	错误返回参数
3	errno	整数	错误码，见 2.3
3	errmsg	字符串	错误说明

错误示例

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "error": {
    "code": -34000,
    "message": "server error",
    "data": {
      "errno": -32010,
      "errmsg": "params cmd must be required!"
    }
  }
}
```

2.1.5 离线语音识别接口--TSocket

接口地址

IP:PORT

thrift 版本: 0.11.0

thrift 接口描述文件:

```
struct Data
```

```
{  
    1:required string    action,           // action  
    2:required binary    buff,           // data  
}
```

```
service MsgPlatform
```

```
{  
    Data Call(1:Data data),           // C/S Interface  
}
```

请求参数列表 (Data. buff)

节点深度	参数	值类型	参数含义	备注
1	sid	字符串	请求的唯一标识号	必选
1	audioName	字符串	音频名	必选
1	audioUrl	字符串	音频路径:(data 为空时, audioUrl 目录下音频引擎能访问到)	必选
1	data	字符串	音频数据: base64 编码的音频数据流	必选
1	ResultType	字符串	返回结果格式(默认为.xml) Q1.xml Q1 格式 xml 文件 sent.txt 识别结果文件 itn.txt 转阿拉伯之后识别结果文件 punc.txt 加标点之后识别结果文件 sentItnPunc_result.txt 转阿拉伯加标点之后识别结果文件 emotionItnPunc_result.txt 转阿拉伯加标点之后情绪文	可选
1	channels	字符串	声道数:(channels/stereo_on 不传默认为单声道) 0: 单声道 1: 双声道	可选
1	stereo_on	字符串	双声道类型:(左右声道内容是否一致,单声道默认为 4): 3: 不一致 4: 一致	可选

请求参数:

```
{  
    "sid": "4d5660000000",  
    "audioName": "4d5660000000.wav",  
    "audioUrl": "/home/thinkit/tempvoice/" ,  
    "data": "base64 编码的音频数据",  
    "ResultType": ".xml",  
    "channels": "0",  
}
```

```
"stereo_on": "4"

}
```

返回参数列表

节点深度	参数	值类型	参数含义
1	voiceStatus	字符串	返回处理状态：2-成功；3-失败；
1	result	字符串	返回结果

正确示例：

```
{
  "voiceStatus": "2",
  "result": "xml 文件内容"
}
```

错误示例：

```
{
  "voiceStatus": "3",
  "result": "voice transcode failed"
}
```

2.1.6 离线识别结果



转写xml格式说明.docx

附件：

2.2 json-rpc 协议错误码

Jsonrpc 错误码	说明
-32700	服务端接收到无效的 json。该错误发生于服务器尝试解析 json 文本
-32600	发送的 json 不是一个有效的请求对象
-32601	该方法不存在或无效
-32602	无效的方法参数。
-32603	JSON-RPC 内部错误
-34000	服务端错误

2.3 错误码

错误编码	说明
-32010	缺少 CMD 参数或 CMD 参数为空
-32011	缺少 CHANNELS 参数或 CHANNELS 参数为空
-32012	缺少 STEREO_ON 参数或 STEREO_ON 参数为空
-32013	连接 REDIS 失败
-32014	DATA 数据为空
-32015	缺少 SID 参数或 SID 参数为空
-32016	Redis 队列内缓存数量超限
-32017	SID 不存在, 请发送 SSB 获取 SID
-32018	AUW 已发送结束状态标识, 请勿继续发送语音
-32019	Type 参数值异常
-32020	获取返回文件类型失败
-32024	操作线程数参数异常
-32025	获取引擎列表失败
-32026	获取引擎信息失败
-32027	设置引擎线程数超过最大值
-32028	设置引擎线程数失败
-32029	AUW 已经结束
-32030	USER 或者 TOKEN 参数不存在或者为空
-32031	USER 与 TOKEN 不匹配
-32032	USER 超过并发限制
-32033	USER 超过通讯量限制
-32034	USER 获取并发量失败
-32035	USER 获取通讯量失败
-32036	找不到语音文件
-32037	找不到 XML 文件
-32038	语音文件处理失败
-32039	无法识别语音处理状态
-32040	语音文件推送失败
-32041	branch 或者 urgent 参数值异常
-32042	语音太短
-32043	异常语音不能处理

3 参考文献

1. 《JSON-RPC 2.0 规范(中文版)》
2. <http://www.jsonrpc.org/historical/json-rpc-2-0.html>