

捷通华声
灵云公有云能力平台
4.0.0
HTTP 开发手册-NLU

灵云科技 源自清华 服务全球

北京捷通华声语音技术有限公司
Beijing InfoQuick SinoVoice Speech Technology Corp.

2015 年 01 月

目 录

- 1. 概述..... 1
- 2. NLU 识别访问接口..... 1
 - 2.1. 接口描述..... 1
 - 2.2. 访问..... 1
 - 2.3. HTTP 头..... 1
 - 2.4. intention 参数说明..... 2
 - 2.5. Needcontent 说明..... 3
 - 2.6. 请求数据说明..... 3
 - 2.7. 服务器应答..... 4
 - 2.8. 错误码描述..... 5

1. 概述

灵云公有云能力平台 NLU 服务器对外提供 http 识别访问接口。

NLU 识别访问接口完全由灵云体系控制，包括集群环境配置、能力分派、授权机制、计费机制等，其部署也是和灵云其它能力的对外接口部署在一起的。

灵云公有云能力平台对该协议的支持情况如下表所示：

支持该协议的版本号	备注
3.8.0	从 3.8.0 版本开始支持该协议

2. NLU 识别访问接口

2.1. 接口描述

外部系统（如灵云官网演示系统）调用本接口进行问题识别，并同步返回识别结果。

外部系统直接利用预先定义好的ServiceUrl，也即服务接口，不用象灵云SDK一样从cloudUrl中通过applyServiceUrl接口获得ServiceUrl。

为和灵云能力接口规范一致，目前采用HTTP头传递必要配置信息，包体为二进制数据，响应为xml格式。

2.2. 访问

（POST）[ServiceUrl]/nlu/Recognize

说明：

需要以下http头：x-app-key、x-sdk-version、x-tid、x-eid、x-request-date、x-session-key、x-task-config。

包体：包含问题内容及上下文的json字符串，utf8编码。

2.3. HTTP 头

客户端POST请求时HTTP头信息，必须带上如下：

顺序	名称	值（例子）	含义	备注
1	x-app-key	ac5d5431	应用标识	必选，由捷通公司提供
2	x-sdk-version	3.8	sdk 版本号	必选，使用 http 接口时，可以指定为 3.8
3	x-request-date	2014-6-18 10:10:11	请求时间	必选，每个任务的请求时间不能重复
4	x-task-config	capkey=nlu.cloud, intention=weather;call; map,lon=12332,lat=322 32	任务参数信息	为 name=value 形式，多个参数以逗号 隔开 capkey 为必选 Intention 必选,标识了请求的领域列表 Lon,lat 可选，为选择了 poi 时的辅助参 数
5	x-session-key	18EA9384AA34	请求数据签名	必选 x-session-key 生成算法说明： x-session-key =md5 (x-request-date + devkey) devkey 是应用开发者密钥，在注册开发 者用户时由开发者网站统一分配，用作 请求数据签名；
6	x-tid	49652372	天行 ID	可选，通过注册接口获取
7	x-eid	42072569	设备 ID	可选，通过注册接口获取

注：头的内容必须为英文，仅支持utf8字符编码。

2.4. intention 参数说明

intention的参数代表了用于进行识别的领域：

<1>:poi

可以随带参数

Lon	经度
Lat	纬度

<2>:weather

<3>:call

<4>:map

2.5. Needcontent 说明

Taskconfig 中,根据是否需要内容,needcontent=yes/or。

服务端可以根据客户意图获取相应的答案,如果客户端不需要此答案,则可将 needcontent 置为 no,即:needcontent=no,则服务端不会去获取,直接将答案内容置为 null。

即:

```
<?xml version="1.0"
encoding="UTF-8"?><ResponseInfo><ResCode>Success</ResCode><ResMessage>Success</ResMessage><ErrorNo>0</ErrorNo><Result_Token>1_8_60_6816_20140901134703_2818</Result_Token><ResultText>{"answer":{"content":"","intention":{"action":"query","date":"20140901","domain":"weather","entry_score":502100,"place":null}}}
</ResultText><Context>{"action":"query","date":"20140901","domain":"weather","entry_score":502100}
</Context></ResponseInfo>
```

如:将 needcontent 设置为 yes,即:needcontent=yes 结果中会包含从内容平台获取的答案:

```
<?xml version="1.0"
encoding="UTF-8"?><ResponseInfo><ResCode>Success</ResCode><ResMessage>Success</ResMessage><ErrorNo>0</ErrorNo><Result_Token>1_8_60_6816_20140901134703_2818</Result_Token><ResultText>{"answer":{"content":"[{\"date\":\"2014-08-29\", \"description\": \"中雨\", \"temperature\": {\"high\": \"高温 26℃\", \"low\": \"低温 19℃\"}, \"wind\": {\"direction\": \"微风\", \"power\": \"小于 3 级\"}}]\", \"intention\": {\"action\": \"query\", \"date\": \"20140901\", \"domain\": \"weather\", \"entry_score\": 502100, \"place\": null}}}</ResultText><Context>{\"action\": \"query\", \"date\": \"20140901\", \"domain\": \"weather\", \"entry_score\": 502100}
</Context></ResponseInfo>
```

2.6. 请求数据说明

在传递请求时, 请求内容格式设计为:

```
{
  "question":
  {
    "query": "北京今天的天气怎么样啊"
  },
  "context": {}
}
```

```
}

```

其中,context为上次识别中的结果。

2.7. 服务器应答

应答格式: XML 文件

XML 的元素 (Element) 及含义

顺序	名称	值 (例子)	含义	备注
1	ResCode	Success (FAILED)	成功 (失败)	必选
2	ResMessage	处理成功 (引擎尚未初始化。)	具体错误描述信息	必选
3	ErrorNo	0 (1)	实际处理的错误码	必选
3	Result_Token	1_8_20_21608_20131118192712_0	结果令牌	成功时必选
4	ResultText	<pre>< ResultText> { "answer": { "content": "", "intention": { "action": "query", "date": "20140901", "domain": "weather", "entry_score": 502100, "place": "北京" } } } </ResultText> <Context> { "action": "query", "date": "20140901", "domain": "weather", "entry_score": 502100 } </pre>	识别结果 json 字符串	成功时必选

		} </Context>		
--	--	-----------------	--	--

2.8. 错误码描述

如果正确,则resCode为Success,且errono为0.

如果出现错误, resCode为FAILED, ErrorNo相应的错误码,并且会在resMessage域中返回错误说明。ErrorNo具体定义如下:

0:处理成功。

60001: engine has not be initied(引擎尚未初始化)。

60002: start to stop engine, do not accept new recognize(服务开始关闭,不再接受请求)。

60003: request token is empty(请求token为空)。

60004: request config error(请求的配置信息错误)。

60005: there is wrong intention privilege request in task config(请求的配置信息中,包含了错误的领域名称(或者无权限访问))。

60006: session find or found failed,maybe license error(引擎建立session失败)。

60007: Load domains failed(引擎加载领域数据失败)。

60008: session is not init(识别时,引擎session句柄尚未初始化)。

60009: question json invalid(解析请求json数据失败)。

60010: think engine process failed(引擎识别失败)。

60011: engine process 0 intention(没有识别出意图)。

60012: there is no itention in app domains(发送请求的app支持的领域列表为空)。