

捷通华声

灵云公有云能力平台

HTTP 开发手册-ASR

灵云科技 源自清华 服务全球

北京捷通华声科技股份有限公司

Beijing SinoVoice Technology Co., Ltd.

2016 年 04 月

目 录

1. 概述.....	1
2. HTTP 头.....	1
3. ASR 访问接口	2
3.1. ASR 识别请求接口	2
3.2. 访问	2
3.3. 服务器应答	2
3.4. 示例	4
3.5. x-task-config 参数说明.....	9

1. 概述

灵云能力平台 HTTP ASR 接口，客户端通过该接口使用 HTTP POST 方式与灵云能力平台进行通讯，并获取返回数据。接口支持流式识别方式，即在录音的同时可以将数据分为多次上传，灵云能力平台可以实时地进行识别过程，以提高响应速度。

2. 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	abcd1234	应用标识	必选，由捷通公司提供
2	x-sdk-version	5.0	sdk 版本号	必选，使用 http 接口时，可以指定为 5.0
3	x-request-date	2015-6-18 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=asr.cloud.freetalk, audioformat=alaw	任务参数信息	必选，为 name=value 形式，多个参数以逗号隔开
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)
6	x-udid	101:1234567890	设备标识	必选，如使用设备取设备标识号，如不使用设备设置为例子中的默认值
7	x-result-format	xml	返回数据格式	可选，指定返回数据的格式，支持 xml 和 json 两种，默认为 xml
8	x-max-json-response	1024	限制返回的 json 形式结果时，指定返回结果中每段的最大长度	可选，仅当 x-result-format 指定为 json 时有效，最小为 1024

注：头的内容必须为英文。

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；

3. ASR 访问接口

3.1. ASR 识别请求接口

本接口在提交识别请求时，同步在 http 请求的回应消息中返回识别状态或识别结果。

3.2. 访问

(POST) [ServiceUrl]/asr/Recognise

说明：

包体中传输语音数据

需要以下 HTTP 头：x-app-key、x-sdk-version、x-request-date、x-task-config、x-session-key、x-udid

3.3. 服务器应答

应答 Header 及含义：

顺序	名称	值（例子）	含义	备注
1	time_used	100	服务端处理耗时，单位是毫秒	必选
2	result_size	300,2000	返回 json 格式时，各分段的长度，第一个值为 json 字符串的长度，单位是字节数；	可选，仅当 x-result-format 指定为 json 时返回

注：result_size 对结果的影响描述

当结果中 NLU 内容的 content, text, answer 等字段超过 x-max-json-response 时，该字段的值将依次被提出，放置到返回 json 的后面，字段名被修改为 content_ref, text_ref, answer_ref，值设为占位符 \$3, \$2 等，该值代表本字段在 result_size 中拥有的段数；提取结果时首先通过 result_size 获取各段的长度信息，获取 json 字符串和相应的字段真实值。

应答字段及含义(XML/JSON):

顺序	名称	值（例子）	含义	备注
----	----	-------	----	----

1	ResCode	Success (InProgress,Failed)	成功（处理中，失败）	必选
2	ResMessage	成功（处理中，失败）	处理结果的详细描述	必选
3	Result_Token	1_1_20_28868_20141218093132_3923	结果令牌	必选
4	ResultCount	2	结果个数	可选
5	ResultIndex	2	返回结果序号。结果为空是为参数 0，但最后一个结果为空时为有效序号。	可选
6	SegmentCount	1	有效结果段个数	可选
7	SegmentIndex	1	段序号，为整个返回结果中的段序号	可选
7	Result	XML 格式： <Result> < ResultText>你好</ResultText> <Score>90</Score> </Result> JSON 格式： { "Result": { "ResultText": "你好", "Score": "90" } }	识别结果，其中 ResultText 是结果文本；Score 为结果得分。	成功时必选
8	Context	XML 格式： <Context>{"city":null,		使用

		<pre>"date":"20140812","description":"天气查询","domain":"weather"}</Context></pre> <p>JSON 格式:</p> <pre>{ "Context": { "city": null, "date": "20140812", "description": "天气查询", "domain": "weather" } }</pre>		asr.cloud.dialog 作为 capkey, 成功时必选
9	ErrorNo	8	失败时错误码	失败时必选
10	StartTime	音频段的起始时间, 单位 ms		可选
11	EndTime	音频段的结束时间, 单位 ms		可选

3.4. 示例

3.4.1 术语解释

非流式识别: 将录音数据以一个数据段方式通过接口传送, 识别引擎处理完成后直接返回结果。

流式识别: 将录音数据以多个数据段通过接口传送, 识别引擎在收到有序的录音数据后开始进行处理的识别方式, 在接收完所有数据后返回识别结果。该方式的优点在于, 引擎不用等待接收完所有的录音数据后再处理, 缩短获得识别结果的等待时间。该方式通过 HTTP 头里 **x-task-config** 参数中的 **identify** 字段和 HTTP 头里 **x-udid** 组成的字符串来标识一次会话, 通过 **x-task-config** 参数中的 **index** 字段来标识一次识别会话中的录音数据分段序号。关于 **index** 的详情请参考后面的说明。

实时流反馈: 该方式通过 HTTP 头里 **x-task-config** 参数中的 **realtime** 字段进行控制, 当设置 **realtime=rt** 时, 识别结果可能会在每次分段请求时返回; 当设置 **realtime=no** 时, 识别结果会在引擎处理完所有的分段录音数据后返回。

3.4.2 实时流反馈 (realtime=rt) 的结果示例:

其中 SegmentIndex 的序号是本段结果在本次识别会话中的序列号。

识别完成结果示例:

XML 格式:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResponseInfo>
  <ResCode>Success</ResCode>
  <ResMessage>Success</ResMessage>
  <ErrorNo>0</ErrorNo>
```

```
<Result_Token>1_8_20_24956_20141111191307_2722</Result_Token>
<ResultCount>1</ResultCount>
<ResultIndex>2</ResultIndex>
<Result>
  <SegmentCount>2</SegmentCount>
  <Segment>
    <SegmentIndex>4</SegmentIndex>
    <Text>abcdefg</Text>
    <Score>106</Score>
    <StartTime>100</StartTime>
    <EndTime>3000</EndTime>
  </Segment>
  <Segment>
    <SegmentIndex>5</SegmentIndex>
    <Text>hijklmn</Text>
    <Score>102</Score>
    <StartTime>3500</StartTime>
    <EndTime>5000</EndTime>
  </Segment>
</Result>
</ResponseInfo>
```

JSON 格式:

```
{
  "ResponseInfo": {
    "ResCode": "Success",
    "ResMessage": "Success",
    "ErrorNo": "0",
    "Result_Token": "1_8_20_24956_20141111191307_2722",
    "ResultCount": "1",
    "ResultIndex": "2",
    "Result": {
      "SegmentCount": "2",
      "Segment": [
        {
          "SegmentIndex": "4",
          "Text": "abcdefg",
          "Score": "106",
          "StartTime": "100",
          "EndTime": "3000"
        },
        {
          "SegmentIndex": "5",
          "Text": "hijklmn",
```

```
        "Score": "102",
        "StartTime": "3500",
        "EndTime": "5000"
    }
]
}
}
```

实时流反馈中间结果示例:

XML 格式:

```
<?xmlversion="1.0" encoding="UTF-8"?>
<ResponseInfo>
  <ResCode>InProgress</ResCode>
  <ResMessage>expect more chunk</ResMessage>
  <ErrorNo>2007</ErrorNo>
  <Result_Token>1_8_20_24956_20141111191307_2722</Result_Token>
  <ResultCount>1</ResultCount>
  <ResultIndex>2</ResultIndex>
  <Result>
    <SegmentCount>2</SegmentCount>
    <Segment>
      <SegmentIndex>1</SegmentIndex>
      <Text>abcdefg</Text>
      <Score>106</Score>
      <StartTime>100</StartTime>
      <EndTime>3000</EndTime>
    </Segment>
    <Segment>
      <SegmentIndex>2</SegmentIndex>
      <Text>hijklmn</Text>
      <Score>102</Score>
      <StartTime>3500</StartTime>
      <EndTime>5000</EndTime>
    </Segment>
  </Result>
</ResponseInfo>
```

JSON 格式:

```
{
  "ResponseInfo": {
    "ResCode": "InProgress",
    "ResMessage": "expect more chunk",
    "ErrorNo": "2007",
```



```
"Result_Token": "1_8_20_24956_20141111191307_2722",
"ResultCount": "1",
"ResultIndex": "2",
"Result": {
  "SegmentCount": "2",
  "Segment": [
    {
      "SegmentIndex": "1",
      "Text": "abcdefg",
      "Score": "106",
      "StartTime": "100",
      "EndTime": "3000"
    },
    {
      "SegmentIndex": "2",
      "Text": "hijklmn",
      "Score": "102",
      "StartTime": "3500",
      "EndTime": "5000"
    }
  ]
}
}
```

识别失败结果示例:

XML 格式:

```
<?xml version="1.0"?>
<ResponseInfo>
  <ResCode>Failed</ResCode>
  <ResMessage>CheckSign failed</ResMessage>
  <ErrorNo>-8</ErrorNo>
  <Result_Token>1_8_20_21608_20131118192712_0</Result_Token>
</ResponseInfo>
```

JSON 格式:

```
{
  "ResponseInfo": {
    "ResCode": "Failed",
    "ResMessage": "CheckSign failed",
    "ErrorNo": "-8",
    "Result_Token": "1_8_20_21608_20131118192712_0"
  }
}
```

3.4.3 非流式识别/流式识别(realtime=no)的结果实例:

识别完成结果示例:

XML 格式:

```
<?xml version="1.0"?>
<ResponseInfo>
  <ResCode>Success</ResCode>
  <ResMessage>Success</ResMessage>
  <ErrorNo>0</ErrorNo>
  <Result_Token>1_8_20_24956_20141111191307_2722</Result_Token>
  <ResultCount>2</ResultCount>
  <Result>
    <Text>abcd</Text>
    <Score>90</Score>
  </Result>
  <Result>
    <Text>abce</Text>
    <Score>80</Score>
  </Result>
</ResponseInfo>
```

JSON 格式:

```
{
  "ResponseInfo": {
    "ResCode": "Success",
    "ResMessage": "Success",
    "ErrorNo": "0",
    "Result_Token": "1_8_20_24956_20141111191307_2722",
    "ResultCount": 2,
    "Result": [
      {
        "Text": "abcd",
        "Score": "90"
      },
      {
        "Text": "abce",
        "Score": "80"
      }
    ]
  }
}
```

中间结果示例:

XML 格式:

```
<?xml version="1.0"?>
<ResponseInfo>
  <ResCode>InProgress</ResCode>
  <ResMessage>expect more chunk</ResMessage>
  <ErrorNo>2007</ErrorNo>
  <Result_Token>1_8_20_24956_20141111191307_2722</Result_Token>
</ResponseInfo>
```

JSON 格式:

```
{
  "ResponseInfo": {
    "ResCode": "InProgress",
    "ResMessage": "expect more chunk",
    "ErrorNo": "2007",
    "Result_Token": "1_8_20_24956_20141111191307_2722"
  }
}
```

识别失败结果示例:

XML 格式:

```
<?xml version="1.0"?>
<ResponseInfo>
  <ResCode>Failed</ResCode>
  <ResMessage>CheckSign failed</ResMessage>
  <ErrorNo>-8</ErrorNo>
  <Result_Token>1_8_20_21608_20131118192712_0</Result_Token>
</ResponseInfo>
```

JSON 格式:

```
{
  "ResponseInfo": {
    "ResCode": "Failed",
    "ResMessage": "CheckSign failed",
    "ErrorNo": "-8",
    "Result_Token": "1_8_20_21608_20131118192712_0"
  }
}
```

3.5. x-task-config 参数说明

使用 ASR 进行语音数据识别时需要指定使用的引擎，目前支持引擎有：

✧ **capkey**（语音合成能力名称，必选）

capkey	备注
asr.cloud.freetalk	基于自由说引擎的语法识别，支持通用(common)、导航(poi)、音乐(music)、购物(shopping)、电信(telecom)等领域。
asr.cloud.freetalk.poi	基于自由说引擎的语法识别，仅支持导航(poi)领域。
asr.cloud.freetalk.music	基于自由说引擎的语法识别，仅支持音乐(music)领域。
asr.cloud.dialog	基于自由说引擎的意图识别，支持通用(common)领域、购物(shopping)， 不支持实时反馈结果。
asr.cloud.freetalk.uyghur	基于自由说引擎的维语识别，仅支持通用(common)领域
asr.cloud.freetalk.english	基于自由说引擎的语法识别，仅支持通用(common)领域。
asr.cloud.freetalk.taiwan	基于自由说引擎的语法识别，支持通用(common)、导航(poi)领域。
asr.cloud.freetalk.cantonese	基于自由说引擎的语法识别，仅支持通用(common)领域。

✧ **audioformat**（语音格式，必选），目前支持以下格式

audioformat	备注
pcm16k16bit	16k16bit 录音数据，支持带 wav 封装头
pcm8k16bit	8k16bit 录音数据，支持带 wav 封装头
ulaw16k8bit	16k8bit u-law 录音数据，支持带 wav 封装头
ulaw8k8bit	8k8bit u-law 录音数据，支持带 wav 封装头
alaw16k8bit	16k8bit a-law 录音数据，支持带 wav 封装头
alaw8k8bit	8k8bit a-law 录音数据，支持带 wav 封装头
amr	AMR 窄带 (8k16bit) 压缩录音数据

speex	speex 压缩格式，需使用捷通公司自定义的封装格式
opus	opus 压缩格式，需使用捷通公司自定义的封装格式

注意：实时流反馈时不支持音频数据带 wav 封装头。

✧ **addpunc**（是否需要为识别结果添加标点，可选）：

yes	需要
no	不需要（缺省）

✧ **domain**（指定ASR识别领域，可选）：

common	通用识别（缺省）（仅支持 16k 录音数据）
poi	导航（仅支持 16k 录音数据）
music	音乐（仅支持 16k 录音数据）
shopping	购物（仅支持 16k 录音数据）
telecom	电信（仅支持 8k 录音数据）
carnav_common	车载通用（仅支持 16k 录音数据）
carnav_poi	车载导航（仅支持 16k 录音数据）

✧ **intention**（指定意图搜索领域，可选）：

在使用 asr.cloud.dialog 进行意图识别时，指定 NLU 意图搜索的领域，多个领域间用“;”

分割，例如：intention=weather;map。

只有当capkey为asr.cloud.dialog时，此参数才有效；其他capkey时将忽略此参数。

asr.cloud.dialog支持的intention:

weather	天气
map	地图
contacts	打电话
music	音乐
train	列车
flight	航班
news	新闻
message	发短信
schedule	提醒

stock	股票
app	应用控制
musicControl	音乐控制
videoControl	视频控制
video	视频点播
deviceControl	终端设备控制
carControl	车载控制
story	故事
joke	笑话
radio	收音机
calendar	日历
tvChannels	频道切换
cookbook	菜谱
website	打开网站
exchangeRate	汇率
websearch	搜索网页
translation	翻译
weibo	发微博
shopping	购物
hotel	酒店查询
restaurant	餐馆查询
yellowpage	黄页
vehicle	智能车载
smartHome	智能家居
himalayan	喜马拉雅点播
poetry	古诗词
arithmetic	简单四则运算
playmusic	音乐领域精确匹配

✧ **qa**（指定问答对话的领域，可选）

在使用 `asr.cloud.dialog` 进行意图识别时，指定 NLU 问答对话的领域，多个领域间用";"分割，例如：`qa=123;abc`

只有当 `capkey` 为 `asr.cloud.dialog` 时，此参数才有效；其他 `capkey` 时将忽略此参数。

✧ **orderby** 意图结果排序规则，可选

指定返回意图结果的排序规则，`orderby=score/intention`，其他参数非法；

`score` 表示按照意图结果分数进行排序，`intention` 表示按照传入时的 `intention` 参数顺序进行排序。默认为 `score`。

✧ **context_len**（指定 `context` 数据长度，可选）

在使用 `asr.cloud.dialog` 进行意图识别时，将上次返回结果中的 `Context` 放在数据后面作为会话的上下文。

只有当 `capkey` 为 `asr.cloud.dialog` 时，此参数才有效；其他 `capkey` 时将忽略此参数。

✧ **resver**（指定意图识别返回结果版本，可选）

在使用 `asr.cloud.dialog` 进行意图识别时，指定返回结果的版本，目前有效取值为 1、2，缺省为 1；

值为 1 时，只返回一个结果，且 NLU 返回的 `json` 中结果为字符串；例如：

```
{
  "result": "福华饭店|富华饭店|福华大饭店",
  "answer": {
    "content": null,
    "intention": {
      "action": "poi list",
      "domain": "map",
      "entry_score": "501500",
      "mode": "search poi",
      "place": "福华饭店"
    }
  }
}
```

值为 2 时，可返回多个结果，且 NLU 返回的 `json` 中结果为字符串数组；例如：

```
{
  "result": "福华饭店|富华饭店|福华大饭店",
  "content": [],
  "intention": [
    {
```

```
        "action": "poi list",
        "domain": "map",
        "entry_score": "501500",
        "mode": "search poi",
        "place": "福华饭店"
    },
    {
        "action": "poi list",
        "domain": "map",
        "entry_score": "501200",
        "mode": "search poi",
        "place": "富华饭店"
    },
    {
        "action": "poi list",
        "domain": "map",
        "entry_score": "501200",
        "mode": "search poi",
        "place": "福华大饭店"
    }
]
```

只有当 capkey 为 asr.cloud.dialog 时，此参数才有效；其他 capkey 时将忽略此参数。

✧ **candnum**（指定意图识别返回结果最大个数，可选）

在使用 asr.cloud.dialog 进行意图识别，且 resver 值为 2 时，指定返回结果最大个数。

只有当 capkey 为 asr.cloud.dialog，且 resver 值为 2 时，此参数才有效；其他情况将忽略此参数。

✧ **needcontent**（指定是否需要获取意图识别内容，可选）

在使用 asr.cloud.dialog 进行意图识别时，服务端可以根据意图结果获取相应内容，如果客户端不需要此项，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。needcontent=yes/no，其他参数非法；缺省为 no。

只有当 capkey 为 asr.cloud.dialog，且 intention 取值正常时，此参数才有效；其他情况将忽略此参数。

✧ **needanswer**（指定是否需要获取 QA 的答案，可选）

在使用 asr.cloud.dialog 进行意图识别时，服务端可以根据客户问题获取相应的答案，如果客户端不需要此答案，则可将 needanswer 置为 no，即：needanswer=no，则服务端不

会去获取，直接将答案内容置为 null。needanswer=yes/no，其他参数非法；缺省为 no。

只有当 capkey 为 asr.cloud.dialog，且 qa 取值正常时，此参数才有效；其他情况将忽略此参数。

✧ **identify(会话标识，必选)：**

该字段为会话的标识，和 x-udid 字段一起作为一次会话的唯一标识。在不同会话之间此唯一标识不可相同（即不同识别任务的[x-udid+indentify]不同）。若采用非流式识别方式，则此参数可不设置。

✧ **index(识别序号，流式识别/实时流反馈时必选)：**

某次会话的请求序号，从 1 开始，最后一次的序号以负数作为终结。可能的序列号举例：

- -1 此时请求数据只有一块
- 1, 2, -3 此时请求数据有三块，此时只有最后一块会返回识别结果，其余均返回 ResCode 为 InProgress（见实时流反馈中间块返回结果示例）

若采用非流式识别，此参数为可选；但如果设置此值，则必须是-1。

✧ **realtime(结果反馈方式，可选)：**

当realtime为rt时，实时流反馈；当realtime为no时，流式识别。

其他可选字段：

- ✧ **lon** 经度，目前保留，将来为导航识别时使用。
- ✧ **lat** 纬度，目前保留，将来为导航识别时使用。
- ✧ **areacode** 城市编码，目前保留，将来为导航识别时使用。
- ✧ **vadhead** 静音检测头部参数，以ms为单位，缺省值10000ms，范围[0-30000]，当超出范围时取缺省值，不返回错误。
- ✧ **vadseg** 静音检测尾部参数，以ms为单位，缺省值500ms，范围[0-30000]，当超出范围时则取缺省值，不返回错误。