

Alibaba Cloud

智能语音交互
录音文件识别

文档版本：20220708

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.接口说明	05
2.Java SDK	21

1.接口说明

录音文件识别是针对已经录制完成的录音文件，进行离线识别的服务。录音文件识别是非实时的，识别的文件需要提交基于HTTP可访问的URL地址，不支持提交本地文件。

使用限制

- 支持单轨和双轨的.wav、.mp3、.m4a、.wma、.aac、.ogg、.amr、.flac格式录音文件识别。
- 文件大小不超过512 MB。
- 需要识别的录音文件必须存放在某服务上，可以通过URL访问。
 - 推荐使用阿里云OSS：如果OSS中文件访问权限为公开，可参见[公共读Object](#)，获取文件访问链接；如果OSS中文件访问权限为私有，可参见[私有Object](#)，通过SDK生成有效时间的访问链接。
 - 您也可以把录音文件存放在自行搭建的文件服务器，提供文件下载。请保证HTTP的响应头（Header）中Content-Length的长度值和Body中数据的真实长度一致，否则会导致下载失败。
- 上传的录音文件URL的访问权限需要设置为公开，URL中只能使用域名不能使用IP地址、不可包含空格，请尽量避免使用中文。

可用URL	不可用URL
<code>https://gw.alipayobjects.com/os/bmw-prod/0574ee2e-f494-45a5-820f-63aee583045a.wav</code>	<code>http://127.0.0.1/sample.wav</code> <code>D:\files\sample.wav</code>

- 录音文件识别属于离线识别服务，对于并发数没有限制，对于QPS（Queries Per Second）的限制如下：
 - POST方式的录音文件识别请求调用接口，用户级别QPS限制为200。
 - GET方式的录音文件识别结果查询接口，用户级别QPS限制为500。
- 免费用户每日可识别不超过2小时时长的录音文件。
- 提交录音文件识别请求后，免费用户的识别任务在24小时内完成并返回识别文本；付费用户的识别任务在3小时内完成并返回识别文本。识别结果在服务端可保存72小时。

说明

一次性上传大规模数据（半小时内上传超过500小时时长的录音）的除外。有大规模数据识别需求的用户，请联系售前专家。

- 智能分轨功能只支持8k单通道的语音。
- 支持调用方式：轮询方式和回调方式。
- 支持语言模型定制。更多信息请参见[语言模型定制](#)。
- 支持热词。更多信息请参见[热词](#)。
- 支持汉语普通话、方言、欧美英语等多种模型识别。查看支持的方言和语种，请参见[语音识别服务支持的方言模型和语种都有哪些？](#)

使用步骤

1. 了解您的录音文件格式和采样率，根据业务场景在管控台选择合适的场景模型。
2. 将录音文件存放至OSS。

如果OSS中文件访问权限为公开，可参见[公共读Object](#)，获取文件访问链接；如果OSS中文件访问权限为私有，可参见[私有Object](#)，通过SDK生成有有效时间的访问链接。

说明

您也可以把录音文件存放在自行搭建的文件服务器，提供文件下载。请保证HTTP的响应头（Header）中 `Content-Length` 的长度值和Body中数据的真实长度一致，否则会导致下载失败。

3. 客户端提交录音文件识别请求。

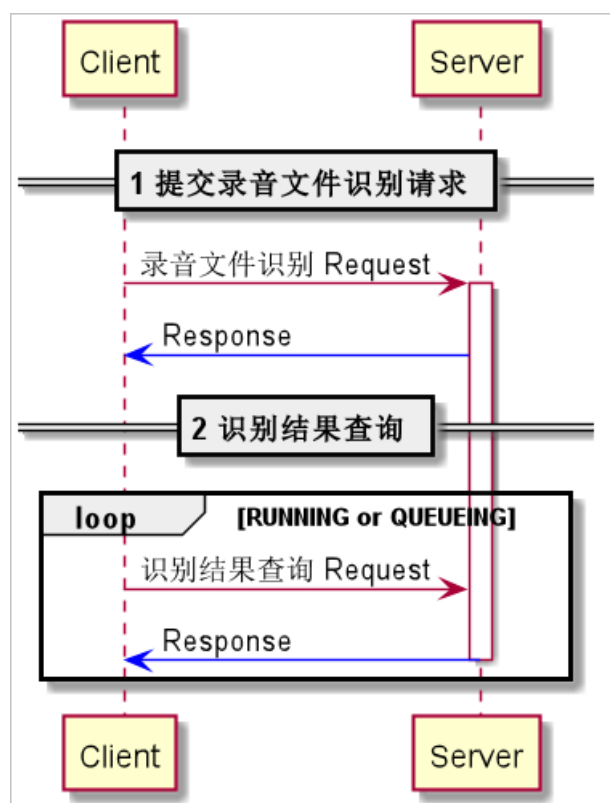
正常情况下，服务端返回该请求任务的ID，用以查询识别结果。

4. 客户端发送识别结果查询请求。

通过步骤3获取的请求任务ID查询录音文件识别的结果，目前识别的结果在服务端可保存72小时。

交互流程

客户端与服务端的交互流程如图所示。



说明

所有服务端的响应都会在返回信息的header包含表示本次识别任务的TaskId参数，请记录该值。如果发生错误，请将该值和错误信息提交到工单。

各地域POP调用参数

地域	调用参数
华东2（上海）	• regionId="cn-shanghai" • endpointName="cn-shanghai" • domain="filetrans.cn-shanghai.aliyuncs.com"
华北2（北京）	• regionId="cn-beijing" • endpointName="cn-beijing" • domain="filetrans.cn-beijing.aliyuncs.com"
华南1（深圳）	• regionId="cn-shenzhen" • endpointName="cn-shenzhen" • domain="filetrans.cn-shenzhen.aliyuncs.com"

接口调用方式

录音文件识别服务是以RPC风格的POP API方式提供录音文件识别接口，将参数封装到每一个请求中，每个请求即对应一个方法，执行的结果放在response中。需要识别的录音文件必须存放在某服务上（推荐[阿里云OSS](#)），可以通过URL访问。使用阿里云OSS，同一地域可以通过内网访问，不计外网流量费用，具体方法请参见[使用录音文件识别时如何设置OSS内网地址](#)。

录音文件识别POP API包括两部分：POST方式的“录音文件识别请求调用接口”（用户级别QPS（queries per second）限制为200）、GET方式的“录音文件识别结果查询接口”（用户级别QPS限制为500）。

- 识别请求调用接口：
 - 当采用轮询方式时，提交录音文件识别任务，获取任务ID，供后续轮询使用。
 - 当采用回调方式时，提交录音文件识别任务和回调URL，任务完成后会把识别结果POST到回调地址，要求回调地址可接收POST请求。

说明

由于历史原因，早期发布的录音文件识别服务（默认为2.0版本）的回调方式和轮询方式的识别结果在JSON字符串的风格和字段上均有不同，下文将作说明。录音文件识别服务在4.0版本对回调方式做了优化，使得回调方式的识别结果与轮询方式的识别结果保持一致，均为驼峰风格的JSON格式字符串。

如果您已接入录音文件识别服务，即没有设置录音文件识别服务的版本，默认为2.0版，可以继续使用；如果您新接入录音文件识别服务，请设置服务版本为4.0。

输入参数及说明：

提交录音文件识别请求时，需要设置输入参数，以JSON格式的字符串传入请求对象的Body，JSON格式如下：

```
{
  "appkey": "your-appkey",
  "file_link": "https://gw.alipayobjects.com/os/bmw-prod/0574ee2e-f494-45a5-820f-63ae583045a.wav",
  "auto_split": false,
  "version": "4.0",
  "enable_words": false,
  "enable_sample_rate_adaptive": true,
  // valid_times: 获取语音指定时间段的识别内容，若不需要，则无需填写。
  "valid_times": [
    {
      "begin_time": 200,
      "end_time": 2000,
      "channel_id": 0
    }
  ]
}
```

参数	值类型	是否必选	说明
appkey	String	是	管控台的项目Appkey。
file_link	String	是	存放录音文件的地址，需要在管控台中将对应项目的模型设置为支持该音频场景的模型。
version	String	是	设置录音文件识别服务的版本，请设置为“4.0”，默认为“2.0”。
enable_words	Boolean	否	是否开启返回词信息，默认为false，开启时需要设置version为“4.0”。
enable_sample_rate_adaptive	Boolean	否	是否将大于16 kHz采样率的音频进行自动降采样（降为16 kHz），默认为false，开启时需要设置version为“4.0”。
enable_callback	Boolean	否	是否启用回调功能，默认值为false。

参数	值类型	是否必选	说明
callback_url	String	否	回调服务的地址，enable_callback取值为true时，本字段必选。URL支持HTTP和HTTPS协议，host不可使用IP地址。
auto_split	Boolean	否	是否开启智能分轨（开启智能分轨，即可在双方对话的语音情景下，依据每句话识别结果中的ChannelId，判断该句话的发言人为哪一方。通常先发言一方ChannelId为0。只支持8000 Hz采样率单通道语音）。
enable_inverse_text_normalization	Boolean	否	是否打开ITN，中文数字将转为阿拉伯数字输出，默认值为false，开启时需要设置version为“4.0”。
enable_disfluency	Boolean	否	是否打开声音顺滑，默认值false，开启时需要设置version为“4.0”。
enable_punctuation_prediction	Boolean	否	是否给句子加标点。默认值true（加标点）。
valid_times	List< ValidTime >	否	有效时间段信息，用来排除一些不需要的时间段。
max_end_silence	Integer	否	允许的最大结束静音，取值范围：200~6000，默认值450，单位为毫秒。
max_single_segment_time	Integer	否	允许单句话最大结束时间，最小值5000，默认值14000。单位为毫秒。

参数	值类型	是否必选	说明
customization_id	String	否	通过POP API创建的定制模型ID，默认不添加。
class_vocabulary_id	String	否	创建的类热词表ID，默认不添加。
vocabulary_id	String	否	创建的泛热词表ID，默认不添加。
enable_semantic_sentence_detection	Boolean	否	是否启用语义断句，取值：true/false，默认值false。
enable_timestamp_alignment	Boolean	否	是否启用时间戳校准功能，取值：true/false，默认值false。
first_channel_only	Boolean	否	是否只识别首个声道，取值：true/false。 ■ 默认为空：8k处理双声道，16k处理单声道。 ■ false：8k处理双声道，16k处理双声道。 ■ true：8k处理单声道，16k处理单声道。  说明 计费模式： ■ 8k处理双声道，按单声道计费，即音频时长进行计费。 ■ 16k处理双声道，按双声道计费，即声道数×音频时长进行计费。

其中，ValidTime对象参数说明如下表所示。

参数	值类型	是否必选	说明
begin_time	Int	是	有效时间段的起始点时间偏移，单位为毫秒。
end_time	Int	是	有效时间段的结束点时间偏移，单位为毫秒。
channel_id	Int	是	有效时间段的作用音轨序号（从0开始）。

输出参数及说明：

服务端返回录音文件识别请求的响应，响应的输出参数为JSON格式的字符串：

```
{
  "TaskId": "4b56f0c4b7e611e88f34c33c2a60****",
  "RequestId": "E4B183CC-6CFE-411E-A547-D877F7BD****",
  "StatusText": "SUCCESS",
  "StatusCode": 21050000
}
```

返回HTTP状态：200表示成功，更多状态码请查阅HTTP状态码。

属性	值类型	是否必选	说明
TaskId	String	是	识别任务ID。
RequestId	String	是	请求ID，仅用于联调。
StatusCode	Int	是	状态码。
StatusText	String	是	状态说明。

● 识别结果查询接口：

提交完录音文件识别请求后，按照如下参数设置轮询识别结果。

输入参数：

通过提交录音文件识别请求获得的任务ID作为识别结果查询接口参数，获取识别结果。在接口调用过程中，需要设置一定的查询时间间隔。



注意

查询接口有QPS限制（100QPS），超过QPS之后则可能报错：

`Throttling.User : Request was denied due to user flow control.`。建议查询接口的轮询间隔时长适当延长，不宜过短。

属性	值类型	是否必选	说明
TaskId	String	是	识别任务ID。

输出参数及说明：

服务端返回识别结果查询请求的响应，响应的输出参数为JSON格式的字符串。

○ 正常返回：以录音文件nls-sample-16k.wav（文件为单轨）识别结果为例。

```
{
  "TaskId": "d429dd7dd75711e89305ab6170fe****",
  "RequestId": "9240D669-6485-4DCC-896A-F8B31F94****",
  "StatusText": "SUCCESS",
  "BizDuration": 2956,
  "SolveTime": 1540363288472,
  "StatusCode": 21050000,
  "Result": {
    "Sentences": [{
      "EndTime": 2365,
      "SilenceDuration": 0,
      "BeginTime": 340,
      "Text": "北京的天气。",
      "ChannelId": 0,
      "SpeechRate": 177,
      "EmotionValue": 5.0
    }]
  }
}
```

如果开启enable_callback/callback_url，且设置服务版本为4.0，回调识别结果为：

```
{
  "Result": {
    "Sentences": [{
      "EndTime": 2365,
      "SilenceDuration": 0,
      "BeginTime": 340,
      "Text": "北京的天气。",
      "ChannelId": 0,
      "SpeechRate": 177,
      "EmotionValue": 5.0
    }]
  },
  "TaskId": "36d01b244ad811e9952db7bb7ed2****",
  "StatusCode": 21050000,
  "StatusText": "SUCCESS",
  "RequestTime": 1553062810452,
  "SolveTime": 1553062810831,
  "BizDuration": 2956
}
```

? 说明

- RequestTime为时间戳（单位为毫秒，例如1553062810452转换为北京时间为2019/3/20 14:20:10），表示录音文件识别提交请求的时间。
- SolveTime为时间戳（单位为毫秒），表示录音文件识别完成的时间。

◦ 排队中：

```
{
  "TaskId": "c7274235b7e611e88f34c33c2a60****",
  "RequestId": "981AD922-0655-46B0-8C6A-5C836822****",
  "StatusText": "QUEUEING",
  "StatusCode": 21050002
}
```

◦ 识别中：

```
{
  "TaskId": "c7274235b7e611e88f34c33c2a60****",
  "RequestId": "8E908ED2-867F-457E-82BF-4756194A****",
  "StatusText": "RUNNING",
  "BizDuration": 0,
  "StatusCode": 21050001
}
```

◦ 异常返回：以文件下载失败为例。

```
{
  "TaskId": "4cf25b7eb7e711e88f34c33c2a60****",
  "RequestId": "098BF27C-4CBA-45FF-BD11-3F532F26****",
  "StatusText": "FILE_DOWNLOAD_FAILED",
  "BizDuration": 0,
  "SolveTime": 1536906469146,
  "StatusCode": 41050002
}
```

 说明

更多异常情况请查看下面的服务状态码的错误状态码及解决方案。

返回HTTP状态：200表示成功，更多状态码请查阅HTTP状态码。

属性	值类型	是否必选	说明
TaskId	String	是	识别任务ID。
StatusCode	Int	是	状态码。
StatusText	String	是	状态说明。
RequestId	String	是	请求ID，用于调试。
Result	Object	是	识别结果对象。
Sentences	List< SentenceResult >	是	识别的结果数据。当 StatusText为SUCCEED时存在。
Words	List< WordResult >	否	词信息，获取时需设置 enable_words为true，且设置服务version为" 4.0" 。
BizDuration	Long	是	识别的音频文件总时长，单位为毫秒。

属性	值类型	是否必选	说明
SolveTime	Long	是	时间戳，单位为毫秒，录音文件识别完成的时间。

其中，单句结果SentenceResult参数如下。

属性	值类型	是否必选	说明
ChannelId	Int	是	该句所属音轨ID。
BeginTime	Int	是	该句的起始时间偏移，单位为毫秒。
EndTime	Int	是	该句的结束时间偏移，单位为毫秒。
Text	String	是	该句的识别文本结果。
EmotionValue	Int	是	情绪能量值，取值为音量分贝值/10。取值范围：[1,10]。值越高情绪越强烈。
SilenceDuration	Int	是	本句与上一句之间的静音时长，单位为秒。
SpeechRate	Int	是	本句的平均语速，单位：字数/分钟。

○ 开启返回词信息：

如果enable_words设置为true，且设置服务version为"4.0"，服务端的识别结果将包含词信息。使用轮询方式和回调方式获得的词信息相同，以轮询方式的识别结果为例：

```
{
  "StatusCode": 21050000,
  "Result": {
    "Sentences": [{
      "SilenceDuration": 0,
      "EmotionValue": 5.0,
      "ChannelId": 0,
      "Text": "北京的天气。",
      "BeginTime": 340,
      "EndTime": 2365,
      "SpeechRate": 177
    }],
    "Words": [{
      "ChannelId": 0,
      "Word": "北京",
      "BeginTime": 640,
      "EndTime": 940
    }, {
      "ChannelId": 0,
      "Word": "的",
      "BeginTime": 940,
      "EndTime": 1120
    }, {
      "ChannelId": 0,
      "Word": "天气",
      "BeginTime": 1120,
      "EndTime": 2020
    }
  ],
  "SolveTime": 1553236968873,
  "StatusText": "SUCCESS",
  "RequestId": "027B126B-4AC8-4C98-9FEC-A031158F****",
  "TaskId": "b505e78c4c6d11e9a213e11db149****",
  "BizDuration": 2956
}
```

Word对象说明：

属性	值类型	是否必选	说明
BeginTime	Int	是	词开始时间，单位为毫秒。
EndTime	Int	是	词结束时间，单位为毫秒。
ChannelId	Int	是	该词所属音轨ID。

属性	值类型	是否必选	说明
Word	String	是	词信息。

服务状态码

正常状态码

状态码	状态描述	状态含义	解决方案
21050000	SUCCESS	成功。	POST方式的识别请求接口调用成功，或者GET方式的识别结果查询接口调用成功。
21050001	RUNNING	录音文件识别任务运行中。	请稍后再发送GET方式的识别结果查询请求。
21050002	QUEUEING	录音文件识别任务排队中。	请稍后再发送GET方式的识别结果查询请求。
21050003	SUCCESS_WITH_NO_VALID_FRAGMENT	识别结果查询接口调用成功，但是没有识别到语音。	检查录音文件是否有语音，或者语音时长太短。

错误状态码

 说明

状态码以“4”开头表示客户端错误，以“5”开头表示服务端错误。

状态码	描述	含义	解决方案
41050001	USER_BIZDURATION_QUOTA_EXCEED	单日时间超限（免费用户每日可识别不超过2小时时长的录音文件）。	建议从免费版升级到商用版。 如业务量较大，请联系商务洽谈，邮件地址： nls_support@service.aliyun.com。

状态码	描述	含义	解决方案
41050002	FILE_DOWNLOAD_FAILED	文件下载失败。	检查录音文件路径是否正确，是否可以外网访问和下载。
41050003	FILE_CHECK_FAILED	文件格式错误。	检查录音文件是否是单轨/双轨的WAV格式、MP3格式。
41050004	FILE_TOO_LARGE	文件过大。	检查录音文件大小是否超过512 MB。
41050005	FILE_NORMALIZE_FAILED	文件归一化失败。	检查录音文件是否有损坏，是否可以正常播放。
41050006	FILE_PARSE_FAILED	文件解析失败。	检查录音文件是否有损坏，是否可以正常播放。
41050007	MKV_PARSE_FAILED	MKV解析失败。	检查录音文件是否有损坏，是否可以正常播放。
41050008	UNSUPPORTED_SAMPLE_RATE	采样率不匹配。	检查实际语音的采样率和控制台上Appkey绑定的ASR模型采样率是否一致，或者将本篇文档中自动降采样的参数enable_sample_rate_adaptive设置为true。
41050009	UNSUPPORTED_ASR_GROUP	ASR分组不支持。	确认请求参数appkey值是否设置正确，或者是否与阿里云账号的AccessKey ID同一个账号。
41050010	FILE_TRANS_TASK_EXPIRED	录音文件识别任务过期。	TaskId不存在，或者已过期。
41050011	REQUEST_INVALID_FILE_URL_VALUE	请求file_link参数非法。	确认file_link参数格式是否正确。
41050012	REQUEST_INVALID_CALLBACK_URL_VALUE	请求callback_url参数非法。	确认callback_url参数格式是否正确，是否为空。

状态码	描述	含义	解决方案
41050013	REQUEST_PARAMETER_INVALID	请求参数无效。	确认请求task值为有效JSON格式字符串。
41050014	REQUEST_EMPTY_APPKEY_VALUE	请求参数appkey值为空。	确认是否设置了appkey参数值。
41050015	REQUEST_APPKEY_UNREGISTERED	请求参数appkey未注册。	确认请求参数appkey值是否正确，或者是否与阿里云账号的AccessKey ID同一个账号。
41050021	RAM_CHECK_FAILED	RAM检查失败。	检查您的RAM用户是否已经授权调用语音服务的API，请参见 RAM用户权限配置 。
41050023	CONTENT_LENGTH_CHECK_FAILED	content-length 检查失败。	检查下载文件时，HTTP response中的content-length与文件实际大小是否一致。
41050024	FILE_404_NOT_FOUND	需要下载的文件不存在。	检查需要下载的文件是否存在。
41050025	FILE_403_FORBIDDEN	没有权限下载需要的文件。	检查是否有权限下载录音文件。
41050026	FILE_SERVER_ERROR	请求的文件所在的服务不可用。	检查请求的文件所在的服务是否可用。
41050029	ASR_RESPONSE_HAVE_HO_WORDS	未识别出有效文字。	检查音频内容。
50000000	INTERNAL_ERROR	默认的服务端错误。	如果偶现可以忽略，重复出现请提交工单。
51050000	INTERNAL_ERROR	内部通用错误。	如果偶现可以忽略，重复出现请提交工单。

状态码	描述	含义	解决方案
51050001	VAD_FAILED	VAD失败。	如果偶现可以忽略，重复出现请提交工单。
51050002	RECOGNIZE_FAILED	内部alistr识别失败。	如果偶现可以忽略，重复出现请提交工单。
51050003	RECOGNIZE_INTERRUPT	内部alistr识别中断。	如果偶现可以忽略，重复出现请提交工单。
51050004	OFFER_INTERRUPT	内部写入队列中断。	如果偶现可以忽略，重复出现请提交工单。
51050005	FILE_TRANS_TIMEOUT	内部整体超时失败。	如果偶现可以忽略，重复出现请提交工单。
51050006	FRAGMENT_FAILED	内部分断失败。	如果偶现可以忽略，重复出现请提交工单。

历史版本说明

如果您已接入录音文件识别，即没有设置服务版本为4.0，将默认使用录音文件识别2.0版本。回调方式的识别结果与轮询方式的识别结果在JSON字符串的风格和字段上有所不同，如果开启enable_callback/callback_url，回调识别结果为：

```
{
  "result": [{
    "begin_time": 340,
    "channel_id": 0,
    "emotion_value": 5.0,
    "end_time": 2365,
    "silence_duration": 0,
    "speech_rate": 177,
    "text": "北京的天气。"
  }],
  "task_id": "3f5d4c0c399511e98dc025f34473****",
  "status_code": 21050000,
  "status_text": "SUCCESS",
  "request_time": 1551164878830,
  "solve_time": 1551164879230,
  "biz_duration": 2956
}
```

2.Java SDK

本文介绍如何使用阿里云智能语音服务提供的Java SDK，包括SDK的安装方法及SDK代码示例。

前提条件

- 使用SDK前，请先阅读接口说明，详情请参见[接口说明](#)。
- 本文中的SDK只适用于2.0版语音服务，如果您当前为1.0版需要首先开通2.0版服务，详情请参见[开通服务](#)。

SDK说明

录音文件识别的Java示例使用了阿里云Java SDK的CommonRequest提交录音文件识别请求和识别结果查询，采用的是RPC风格的POP API调用。阿里云Java SDK CommonRequest的使用方法请参见[使用CommonRequest进行调用](#)。

注意

阿里云Java SDK不支持Android开发。

Java依赖

您只需依赖阿里云Java SDK的核心库与阿里开源库fastjson。阿里云Java SDK的核心库版本支持3.5.0及以上（如果版本在4.0.0及以上，需要根据错误提示补充对应的第三方依赖）。

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>3.7.1</version>
</dependency>
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>fastjson</artifactId>
  <version>1.2.49</version>
</dependency>
```

示例说明

说明

下载nls-sample-16k.wav。示例中使用的WAV录音文件为PCM编码格式16000Hz采样率，模型设置为通用模型。

鉴权

通过传入阿里云账号的AccessKey ID和AccessKey Secret（获取方式请参见[开通服务](#)），调用阿里云Java SDK得到client，示例如下：

```
final String accessKeyId = "您的AccessKey Id";
final String accessKeySecret = "您的AccessKey Secret";
/**
 * 地域ID
 */
final String regionId = "cn-shanghai";
final String endpointName = "cn-shanghai";
final String product = "nls-filetrans";
final String domain = "filetrans.cn-shanghai.aliyuncs.com";

IAcsClient client;
// 设置endpoint
DefaultProfile.addEndpoint(endpointName, regionId, product, domain);
// 创建DefaultAcsClient实例并初始化
DefaultProfile profile = DefaultProfile.getProfile(regionId, accessKeyId, accessKeySecret);
client = new DefaultAcsClient(profile);
```

录音文件识别请求调用接口

Java 示例采用轮询的方式，提交录音文件识别请求，获取任务ID，供后续轮询使用。

② 说明

参数设置请参见[接口说明](#)，只需设置JSON字符串中的参数，其他方法的参数值保持不变。

```
/**
 * 创建CommonRequest 设置请求参数
 */
CommonRequest postRequest = new CommonRequest();
postRequest.setDomain("filetrans.cn-shanghai.aliyuncs.com"); // 设置域名, 固定值。
postRequest.setVersion("2018-08-17"); // 设置API的版本号, 固定值。
postRequest.setAction("SubmitTask"); // 设置action, 固定值。
postRequest.setProduct("nls-filetrans"); // 设置产品名称, 固定值。
// 设置录音文件识别请求参数, 以JSON字符串的格式设置到请求Body中。
JSONObject taskObject = new JSONObject();
taskObject.put("appkey", "您的appkey"); // 项目的Appkey
taskObject.put("file_link", "您的录音文件访问链接"); // 设置录音文件的链接
taskObject.put(KEY_VERSION, "4.0"); // 新接入请使用4.0版本, 已接入(默认2.0)如需维持现状, 请注释掉该参数设置。
String task = taskObject.toJSONString();
postRequest.putBodyParameter("Task", task); // 设置以上JSON字符串为Body参数。
postRequest.setMethod(MethodType.POST); // 设置为POST方式请求。
/**
 * 提交录音文件识别请求
 */
String taskId = ""; // 获取录音文件识别请求任务的ID, 以供识别结果查询使用。
CommonResponse postResponse = client.getCommonResponse(postRequest);
if (postResponse.getHttpStatus() == 200) {
    JSONObject result = JSONObject.parseObject(postResponse.getData());
    String statusText = result.getString("StatusText");
    if ("SUCCESS".equals(statusText)) {
        System.out.println("录音文件识别请求成功响应: " + result.toJSONString());
        taskId = result.getString("TaskId");
    }
    else {
        System.out.println("录音文件识别请求失败: " + result.toJSONString());
        return;
    }
}
else {
    System.err.println("录音文件识别请求失败, Http错误码: " + postResponse.getHttpStatus());
    System.err.println("录音文件识别请求失败响应: " + JSONObject.toJSONString(postResponse));
    return;
}
```

录音文件识别结果查询

使用获取的任务ID, 查询录音文件识别的结果。

```

/**
 * 创建CommonRequest，设置任务ID。
 */
CommonRequest getRequest = new CommonRequest();
getRequest.setDomain("filetrans.cn-shanghai.aliyuncs.com"); // 设置域名，固定值。
getRequest.setVersion("2018-08-17"); // 设置API版本，固定值。
getRequest.setAction("GetTaskResult"); // 设置action，固定值。
getRequest.setProduct("nls-filetrans"); // 设置产品名称，固定值。
getRequest.putQueryParameter("TaskId", taskId); // 设置任务ID为查询参数。
getRequest.setMethod(MethodType.GET); // 设置为GET方式的请求。
/**
 * 提交录音文件识别结果查询请求
 * 以轮询的方式进行识别结果的查询，直到服务端返回的状态描述为“SUCCESS”、“SUCCESS_WITH_NO_VALID_FRAGMENT”，或者为错误描述，则结束轮询。
 */
String statusText = "";
while (true) {
    CommonResponse getResponse = client.getCommonResponse(getRequest);
    if (getResponse.getHttpStatus() != 200) {
        System.err.println("识别结果查询请求失败，Http错误码： " + getResponse.getHttpStatus());
        ;

        System.err.println("识别结果查询请求失败： " + getResponse.getData());
        break;
    }
    JSONObject result = JSONObject.parseObject(getResponse.getData());
    System.out.println("识别查询结果： " + result.toJSONString());
    statusText = result.getString("StatusText");
    if ("RUNNING".equals(statusText) || "QUEUEING".equals(statusText)) {
        // 继续轮询
        Thread.sleep(3000);
    }
    else {
        break;
    }
}
if ("SUCCESS".equals(statusText) || "SUCCESS_WITH_NO_VALID_FRAGMENT".equals(statusText)) {
    System.out.println("录音文件识别成功！");
}
else {
    System.err.println("录音文件识别失败！");
}
}

```

示例代码

```

import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.profile.DefaultProfile;
public class FileTransJavaDemo {

```

```
// 地域ID, 常量, 固定值。
public static final String REGIONID = "cn-shanghai";
public static final String ENDPOINTNAME = "cn-shanghai";
public static final String PRODUCT = "nls-filetrans";
public static final String DOMAIN = "filetrans.cn-shanghai.aliyuncs.com";
public static final String API_VERSION = "2018-08-17";
public static final String POST_REQUEST_ACTION = "SubmitTask";
public static final String GET_REQUEST_ACTION = "GetTaskResult";
// 请求参数
public static final String KEY_APP_KEY = "appkey";
public static final String KEY_FILE_LINK = "file_link";
public static final String KEY_VERSION = "version";
public static final String KEY_ENABLE_WORDS = "enable_words";
// 响应参数
public static final String KEY_TASK = "Task";
public static final String KEY_TASK_ID = "TaskId";
public static final String KEY_STATUS_TEXT = "StatusText";
public static final String KEY_RESULT = "Result";
// 状态值
public static final String STATUS_SUCCESS = "SUCCESS";
private static final String STATUS_RUNNING = "RUNNING";
private static final String STATUS_QUEUEING = "QUEUEING";
// 阿里云鉴权client
IAcsClient client;
public FileTransJavaDemo(String accessKeyId, String accessKeySecret) {
    // 设置endpoint
    try {
        DefaultProfile.addEndpoint(ENDPOINTNAME, REGIONID, PRODUCT, DOMAIN);
    } catch (ClientException e) {
        e.printStackTrace();
    }
    // 创建DefaultAcsClient实例并初始化
    DefaultProfile profile = DefaultProfile.getProfile(REGIONID, accessKeyId, accessKeySecret);
    this.client = new DefaultAcsClient(profile);
}
public String submitFileTransRequest(String appKey, String fileLink) {
    /**
     * 1. 创建CommonRequest, 设置请求参数。
     */
    CommonRequest postRequest = new CommonRequest();
    // 设置域名
    postRequest.setDomain(DOMAIN);
    // 设置API的版本号, 格式为YYYY-MM-DD。
    postRequest.setVersion(API_VERSION);
    // 设置action
    postRequest.setAction(POST_REQUEST_ACTION);
    // 设置产品名称
    postRequest.setProduct(PRODUCT);
    /**
     * 2. 设置录音文件识别请求参数, 以JSON字符串的格式设置到请求Body中。
     */
    JSONObject taskObject = new JSONObject();
    // 设置appkey
```

```

taskObject.put(KEY_APP_KEY, appKey);
// 设置音频文件访问链接
taskObject.put(KEY_FILE_LINK, fileLink);
// 新接入请使用4.0版本，已接入（默认2.0）如需维持现状，请注释掉该参数设置。
taskObject.put(KEY_VERSION, "4.0");
// 设置是否输出词信息，默认为false，开启时需要设置version为4.0及以上。
taskObject.put(KEY_ENABLE_WORDS, true);
String task = taskObject.toJSONString();
System.out.println(task);
// 设置以上JSON字符串为Body参数。
postRequest.putBodyParameter(KEY_TASK, task);
// 设置为POST方式的请求。
postRequest.setMethod(MethodType.POST);
/**
 * 3. 提交录音文件识别请求，获取录音文件识别请求任务的ID，以供识别结果查询使用。
 */
String taskId = null;
try {
    CommonResponse postResponse = client.getCommonResponse(postRequest);
    System.err.println("提交录音文件识别请求的响应: " + postResponse.getData());
    if (postResponse.getHttpStatus() == 200) {
        JSONObject result = JSONObject.parseObject(postResponse.getData());
        String statusText = result.getString(KEY_STATUS_TEXT);
        if (STATUS_SUCCESS.equals(statusText)) {
            taskId = result.getString(KEY_TASK_ID);
        }
    }
} catch (ClientException e) {
    e.printStackTrace();
}
return taskId;
}

public String getFileTransResult(String taskId) {
    /**
     * 1. 创建CommonRequest，设置任务ID。
     */
    CommonRequest getRequest = new CommonRequest();
    // 设置域名
    getRequest.setDomain(DOMAIN);
    // 设置API版本
    getRequest.setVersion(API_VERSION);
    // 设置action
    getRequest.setAction(GET_REQUEST_ACTION);
    // 设置产品名称
    getRequest.setProduct(PRODUCT);
    // 设置任务ID为查询参数
    getRequest.putQueryParameter(KEY_TASK_ID, taskId);
    // 设置为GET方式的请求
    getRequest.setMethod(MethodType.GET);
    /**
     * 2. 提交录音文件识别结果查询请求
     * 以轮询的方式进行识别结果的查询，直到服务端返回的状态描述为“SUCCESS”或错误描述，则结束轮询。
     */
}

```

```

String result = null;
while (true) {
    try {
        CommonResponse getResponse = client.getCommonResponse(getRequest);
        System.err.println("识别查询结果: " + getResponse.getData());
        if (getResponse.getHttpStatus() != 200) {
            break;
        }
        JSONObject rootObj = JSONObject.parseObject(getResponse.getData());
        String statusText = rootObj.getString(KEY_STATUS_TEXT);
        if (STATUS_RUNNING.equals(statusText) || STATUS_QUEUEING.equals(statusText)
    ) {
        // 继续轮询，注意设置轮询时间间隔。
        Thread.sleep(10000);
    }
    else {
        // 状态信息为成功，返回识别结果；状态信息为异常，返回空。
        if (STATUS_SUCCESS.equals(statusText)) {
            result = rootObj.getString(KEY_RESULT);
            // 状态信息为成功，但没有识别结果，则可能是由于文件里全是静音、噪音等导致识别
            // 为空。
            if (result == null) {
                result = "";
            }
        }
        break;
    }
} catch (Exception e) {
    e.printStackTrace();
}
}
return result;
}

public static void main(String args[]) throws Exception {
    if (args.length < 3) {
        System.err.println("FileTransJavaDemo need params: <AccessKey Id> <AccessKey Secret> <app-key>");
    }
    final String accessKeyId = args[0];
    final String accessKeySecret = args[1];
    final String appKey = args[2];
    String fileLink = "https://gw.alipayobjects.com/os/bmw-prod/0574ee2e-f494-45a5-820f-63aee583045a.wav";
    FileTransJavaDemo demo = new FileTransJavaDemo(accessKeyId, accessKeySecret);
    // 第一步：提交录音文件识别请求，获取任务ID用于后续的识别结果轮询。
    String taskId = demo.submitFileTransRequest(appKey, fileLink);
    if (taskId != null) {
        System.out.println("录音文件识别请求成功，task_id: " + taskId);
    }
    else {
        System.out.println("录音文件识别请求失败！");
        return;
    }
    // 第二步：根据任务ID轮询识别结果。
    String result = demo.getFileTransResult(taskId);

```

```

        String result = demo.getFileTransResult(taskId);
        if (result != null) {
            System.out.println("录音文件识别结果查询成功：" + result);
        }
        else {
            System.out.println("录音文件识别结果查询失败！");
        }
    }
}

```

② 说明

如果使用回调方式，请设置如下 `enable_callback`、`callback_url` 参数：

```

taskObject.put("enable_callback", true);
taskObject.put("callback_url", "回调地址");

```

回调服务示例：该服务用于回调方式获取识别结果，假设设置回调地址为 `http://ip:port/filetrans/callback/result`。

```

package com.example.filetrans;

import com.alibaba.fastjson.JSONObject;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

import java.util.regex.Matcher;
import java.util.regex.Pattern;

@RequestMapping("/filetrans/callback")
@RestController
public class FiletransCallBack {
    // 以4开头的状态码是客户端错误。
    private static final Pattern PATTERN_CLIENT_ERR = Pattern.compile("4105[0-9]*");
    // 以5开头的状态码是服务端错误。
    private static final Pattern PATTERN_SERVER_ERR = Pattern.compile("5105[0-9]*");
    // 必须是post方式
    @RequestMapping(value = "result", method = RequestMethod.POST)
    public void GetResult(@RequestBody String body) {
        System.out.println("body: " + body);
        try {
            // 获取JSON格式的文件识别结果。
            String result = body;
            JSONObject jsonResult = JSONObject.parseObject(result);
            // 解析并输出相关结果内容。
            System.out.println("获取文件中转写回调结果：" + result);
            System.out.println("TaskId: " + jsonResult.getString("TaskId"));
            System.out.println("StatusCode: " + jsonResult.getString("StatusCode"));
            System.out.println("StatusText: " + jsonResult.getString("StatusText"));
            Matcher matcherClient = PATTERN_CLIENT_ERR.matcher(jsonResult.getString("Status
Code"));

```

```

Matcher matcherServer = PATTERN_SERVER_ERR.matcher(jsonResult.getString("Status
Code"));
// 以2开头状态码为正常状态码, 回调方式正常状态只返回"21050000"。
if("21050000".equals(jsonResult.getString("StatusCode"))) {
    System.out.println("RequestTime: " + jsonResult.getString("RequestTime"));
    System.out.println("SolveTime: " + jsonResult.getString("SolveTime"));
    System.out.println("BizDuration: " + jsonResult.getString("BizDuration"));
    System.out.println("Result.Sentences.size: " +
        jsonResult.getJSONObject("Result").getJSONArray("Sentences").size());
    for (int i = 0; i < jsonResult.getJSONObject("Result").getJSONArray("Senten
ces").size(); i++) {
        System.out.println("Result.Sentences[" + i + "].BeginTime: " +
            jsonResult.getJSONObject("Result").getJSONArray("Sentences").getJSO
NObject(i).getString("BeginTime"));
        System.out.println("Result.Sentences[" + i + "].EndTime: " +
            jsonResult.getJSONObject("Result").getJSONArray("Sentences").getJSO
NObject(i).getString("EndTime"));
        System.out.println("Result.Sentences[" + i + "].SilenceDuration: " +
            jsonResult.getJSONObject("Result").getJSONArray("Sentences").getJSO
NObject(i).getString("SilenceDuration"));
        System.out.println("Result.Sentences[" + i + "].Text: " +
            jsonResult.getJSONObject("Result").getJSONArray("Sentences").getJSO
NObject(i).getString("Text"));
        System.out.println("Result.Sentences[" + i + "].ChannelId: " +
            jsonResult.getJSONObject("Result").getJSONArray("Sentences").getJSO
NObject(i).getString("ChannelId"));
        System.out.println("Result.Sentences[" + i + "].SpeechRate: " +
            jsonResult.getJSONObject("Result").getJSONArray("Sentences").getJSO
NObject(i).getString("SpeechRate"));
        System.out.println("Result.Sentences[" + i + "].EmotionValue: " +
            jsonResult.getJSONObject("Result").getJSONArray("Sentences").getJSO
NObject(i).getString("EmotionValue"));
    }
}
else if(matcherClient.matches()) {
    System.out.println("状态码以4开头表示客户端错误.....");
}
else if(matcherServer.matches()) {
    System.out.println("状态码以5开头表示服务端错误.....");
}
else {
}
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```