SPEECHMATICS

Speechmatics API V2

General Availability (GA)

Table of Contents

Introduction

This document contains information about the new V2 API for the Speechmatics Speech as a Service (SaaS), including output schemas and usage examples.

> [info] For info
>
> Where the V2 Jobs API differs from the V1 API this is called out with a note like this.

Tools

Code samples in this guide use curl for making HTTP requests to the API, and the jq tool to parse and display JSON responses.

Windows

On a Windows PC you can use these download and installation links to get these tools:

```
https://curl.haxx.se/download.html
https://stedolan.github.io/jq/download/
```

Linux

Use the relevant package manager for your flavour of Linux, which will either be:

```
apt install curl jq
```

or

```
yum install curl jq
```

Mac OS X

On the Mac, the easiest way to install these utilities is using Homebrew:

```
brew install curl jq
```

API Specification

Because the new API is written using the OpenAPI Specification it is possible to auto-generate client libraries using Swagger Codegen. If you are interested in trying this out then please contact support@speechmatics.com.

The V2 API is an evolution of the V1 API, but it uses a more standardised authentication method, provides a richer set of output formats, and has a structured JSON configuration that is easier to extend in future.

Upcoming feature enhancements will only be added to the V2 API; the V1 API for transcription will be discontinued after November 2019.

This document refers to version 201906.0 of the service. The JSON output format is NaN.

For the full V2 API specification please refer to the Jobs API Reference section.

Overview

What's New

The following features are now fully supported using the V2 API

- Custom Dictionary
- Channel Diarization

What's Changed

New hostname

Rather than using the api.speechmatics.com hostname, you should use the hostname asr.api.speechmatics.com.

Authorization headers

Access to the API requires use of an authorization token ('auth token'). In the V1 API your auth token is passed as a query string parameter on the URI. In the V2 API it is done using an Authorization header, which is the recommended OAuth2 approach.

> [info] Auth tokens
>
> Currently there is no way to generate an auth token; if you require a new auth token please contact support@speechmatics.com. In the future we will provide the ability to generate new tokens for the V2 API.

User endpoint

There is no longer a `/user` endpoint, instead jobs are referenced using a simpler `/jobs` endpoint, with user access being controlled using the new authentication header. V2 API calls use a `/v2` path in the URL. Requests to submit or refer to jobs now look like this:

```
https://asr.api.speechmatics.com/v2/jobs/
```

A separate authentication service will be added in future that will provide the equivalent capabilities of `/user` .

Status endpoint

The `/status` endpoint has been removed.

Speaker diarization

Speaker diarization is now *off* by default.

Form fields for configuration

A JSON configuration object ('config JSON') replaces the form fields that were previously used for configuration of a job. The audio file name is still specified as a form field, but all other configuration is passed in the config JSON.

Job IDs are strings

We now use a random string job ID value to refer to jobs, rather than an incrementing integer. The Job IDs that you see will look like this: `yjbmf9kqub`.

Legacy JSON output format dropped

A richer JSON format (json-v2) is now used which provides support for new features. Plain text output (txt) is still available. In the JSON transcript output you will see the following:

```
"format": "2.4"
```

ISO 8601 timestamps

Timestamps are now represented in ISO 8601 format, for example: `2018-10-02T13:10:25Z`. Coordinated Universal Time (UTC) is used (indicated by the *Z* suffix).

Metadata

The `meta` form parameter is replaced with the `tracking` element in config JSON. This supports a title, list of tags and a customer-defined JSON object. You can use this information to track jobs through your workflow.

Egress IP adresses (for whitelisting)

You may want to apply a whitelist to the Speechmatics SaaS; for example for your callback service for notfications. To ensure that only POST request notifications from the Speechmatics service are allowed use the following two public IP addresses: 52.236.176.166, 40.85.99.235 for your whitelist.

Language Support

The Speechmatics V2 API now supports the following languages:

- English (en)
- German (de)
- Spanish (es)
- French (fr)
- Italian (it)
- Dutch (nl)
- Portuguese (pt)
- Japanese (ja)
- Korean (ko)
- Danish (da)
- Polish (pl)
- Catalan (ca)
- Hindi (hi)
- Russian (ru)
- Swedish (sv)
- Bulgarian (bg)
- Slovenian (sl)
- Czech (cs)
- Greek (el)
- Finnish (fi)

- Hungarian (hu)
- Croatian (hr)
- Lithuanian (lt)
- Latvian (lv)
- Romanian (ro)
- Slovakian (sk)
- Mandarin (cmn)
- Norwegian (no)
- Arabic (ar)

Note: it is not our intention to update the legacy V1 SaaS with new language models; in order to ensure you get access to the newest features and most accurate transcription results we recommend that you use the V2 API.

Current Limitations

Email Notifications

There is currently no support for email notification of job completion using the V2 API. However there is full support for notifications using webhooks.

Alignment

Alignment jobs are *not* supported by the V2 API. If you want to submit alignment jobs then you should continue to do so using the V1 API.

API How-To Guide

Basics

In this section we show how to submit an audio file for batch transcription. We show how to poll for the status of a job, and to return a transcript once the job is done. Later on we show an alternate method of retrieving the transcript using a webhook-based notification.

Submitting a Job

The simplest configuration for a transcription job is simply to specify the language that you wish to use. This is encoded in the `language` property of the transcription configuration like this:

```
curl -L -X POST 'https://asr.api.speechmatics.com/v2/jobs' \
   -H 'Authorization: Bearer NDFjOTE3NGEtOWVm' \
   -F data_file=@example.wav \
   -F config='{
     "type": "transcription",
     "transcription_config": { "language": "en" }
   }' \
   | jq
```

Here `NDFjOTE3NGEtOWVm` is a sample authorization token. You would replace this with the token that you have been provided with.

> [info] model parameter
>
> The `model` form parameter in V1 is replaced with the `language` element of `transcription_config`

The JSON response that you see will contain the unique Job ID. For full details see the Jobs API Reference. An example response will look like this:

```
HTTP/1.1 201 Created
Content-Length: 20
Content-Type: application/json
Request-Id: 4d46aa73e1a4c5a6d4ba6c31369e7b2e
Strict-Transport-Security: max-age=15724800; includeSubDomains
X-Azure-Ref: 0XdIUXQAAAADlflaR0qvRQpReZYf+q+FHTE9OMjFFREdFMDMwNwBhN2JjOWQ4MC02YjBiLTQ1NWEtYjE3MS01NGJkZmNiYWE0Y
Tk=
Date: Thu, 27 Jun 2019 14:27:45 GMT

{"id":"dlhsd8d69i"}
```

> [info] model parameter
>
> The `200 OK` status code in V1 is replaced with a `201 Created` status code in V2. This indicates that the job has been accepted for processing; it does not necessarily mean that the job will complete successfully.

The `Request-ID` unqiuely identifies your API request. If you ever need to raise a support ticket we recommend that you include the `Request-ID` if possible, as it will help to identify your job. The `Strict-Transport-Security` response header indicates that only HTTPS access is allowed. The `X-Azure-Ref` response header identifies the load-balancer request.

If you do not specify an auth token in the request, or if the token provided is invalid, then you'll see a 401 status code like this:

```
HTTP/1.1 401 Authorization Required
```

> [info] Maximum job size
>
> The size limit for jobs is 12 hours of audio or 4 GB file size. For example, if you submit a file that is 4.2 GB in size, or 13 hours in length then it may get rejected. You can use tools like `sox` or `ffmpeg` to break up large audio files. If you require additional help or have a need for submission of larger files, please contact support@speechmatics.com.

Checking Job Status

A `201 Created` status indicates that the job was accepted for processing. To check that the job is being processed you can check to see where it is in the processing pipeline as follows:

```
curl -L -X GET 'https://asr.api.speechmatics.com/v2/jobs/yjbmf9kqub' \
   -H 'Authorization: Bearer NDFjOTE3NGEtOWVm' \
   | jq
```

The response is a JSON object containing details of the job, with the `status` field showing whether the job is still processing or not. A value of *done* means that the transcript is now ready. Jobs will typically take less than half the audio length to process; so an audio file that is 40 minutes in length should be ready within 20 minutes of submission. See Jobs API Reference for details. An example response looks like this:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "job": {
    "config": {
      "notification_config": null,
      "transcription_config": {
        "additional_vocab": null,
        "channel_diarization_labels": null,
        "language": "en"
      },
      "type": "transcription"
    },
    "created_at": "2019-01-17T17:50:54.113Z",
    "data_name": "example.wav",
    "duration": 275,
    "id": "yjbmf9kqub",
    "status": "running"
  }
}
```

Possible job status types include:

- running
- done
- rejected
- deleted
- expired

> [info] Alignment jobs
>
> Alignment is not supported with the V2 API. Speechmatics supports alignment jobs only on the V1 API.

Retrieving a Transcript

Transcripts can be retrieved as follows:

```
curl -L -X GET 'https://asr.api.speechmatics.com/v2/jobs/yjbmf9kqub/transcript' \
   -H 'Authorization: Bearer NDFjOTE3NGEtOWVm' \
   | jq
```

The supported formats are `json-v2` (the default), and plain text `txt`. The format is set using the `format` query string parameter, for example:

```
curl -L -X GET 'https://asr.api.speechmatics.com/v2/jobs/yjbmf9kqub/transcript?format=txt' \
   -H 'Authorization: Bearer NDFjOTE3NGEtOWVm'
```

See Jobs API Reference for details.

> [info] Legacy JSON output format
>
> The current `json` output format for the V1 API is considered legacy as it does not support the new features like *Channel Diarization* as well as allowing future feature expansion.

The `job` property contains information about the job such as the job ID and duration.

The `metadata` property contains whatever tracking information was included when the job was submitted, as well as the configuration that was used to submit the job.

The `results` section which is an array of words comprising the actual transcript. An example transcript using the JSON output format is shown:

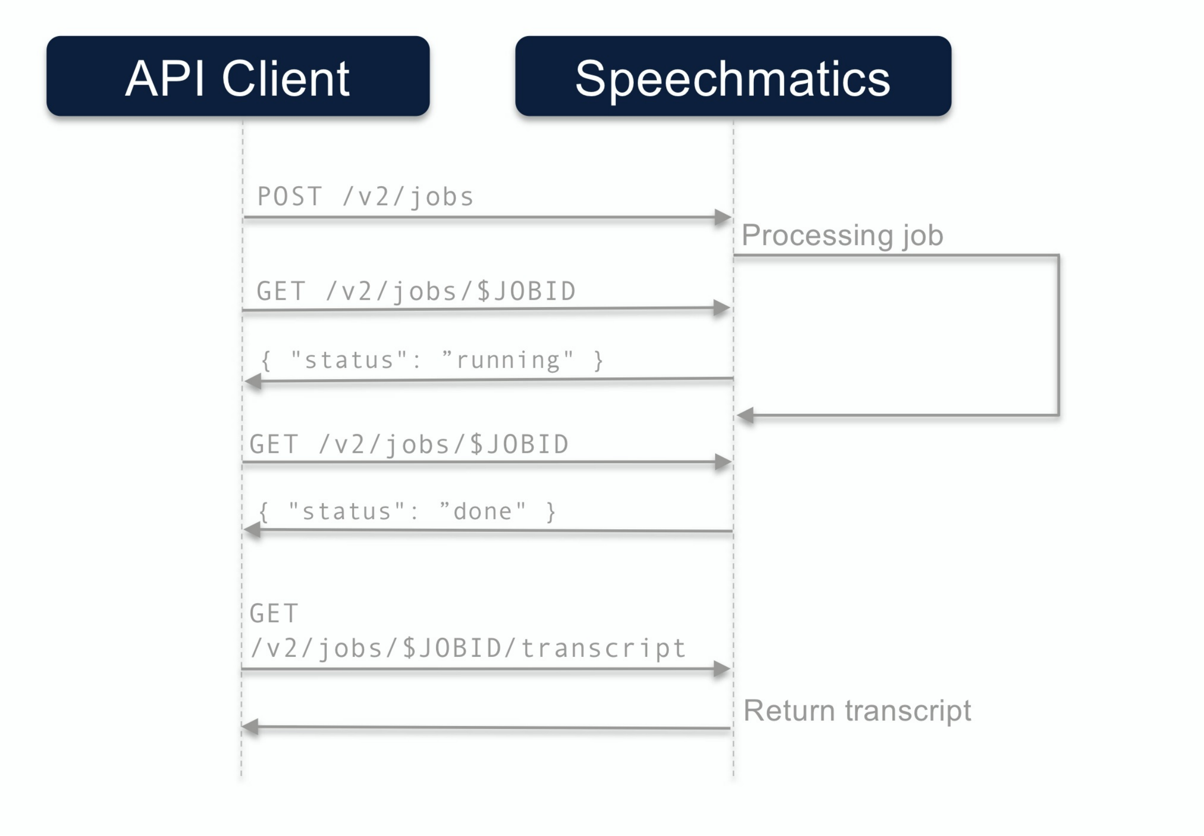
```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: application/vnd.speechmatics.v2+json
Request-Id: aad4eb68bca69f3f277d202456bb0b15
Strict-Transport-Security: max-age=15724800; includeSubDomains
X-Azure-Ref: 0ztQUXQAAAAA4ifGtw2COQKyb52QoNgX4TE9OMjFFREdFMDMyMABhN2JjOWQ4MC02YjBiLTQ1NWEtYjE3MS01NGJkZmNiYWE0Y
Tk=
Date: Thu, 27 Jun 2019 14:38:05 GMT

{
   "format":"2.4",
   "job":{
      "created_at":"2019-01-17T17:50:54.113Z",
      "data_name":"example.wav",
      "duration":275,
      "id":"yjbmf9kqub"
   },
   "metadata":{
      "created_at":"2019-01-17T17:52:26.222Z",
      "transcription_config":{
         "diarization":"none",
         "language":"en"
      },
      "type":"transcription"
   },
   "results":[
      {
         "alternatives":[
            {
               "confidence":0.9,
               "content":"Just",
               "language":"en",
               "speaker":"UU"
            }
         ],
         "end_time":1.07,
         "start_time":0.9,
         "type":"word"
      },
      {
         "alternatives":[
            {
               "confidence":1,
               "content":"this",
               "language":"en",
               "speaker":"UU"
            }
         ],
         "end_time":1.44,
         "start_time":1.11,
         "type":"word"
      },
      {
         "alternatives":[
            {
               "confidence":1,
               "content":".",
               "language":"en",
               "speaker":"UU"
            }
         ],
         "end_time":273.64,
         "start_time":273.64,
         "type":"punctuation"
      }
   ]
}
```

> [info] created_at timestamps
>
> There are two `created_at` timestamps, one in the job section which indicates when the job was submitted and another in the metadata section which indicates when the output was produced. The time difference provides the transcription time, including any system delays as well as the actual time taken to process the job.

Polling Call Flow

The call flow for the polling method looks like this:



Fetch URL

The previous example showed how to create a job from a locally uploaded audio file. If you store your digital media in cloud storage (for example AWS S3 or Azure Blob Storage) you can also submit a job by providing the URL of the audio file. The configuration uses a `fetch_data` section, which looks like this:

```
curl -L -X POST 'https://asr.api.speechmatics.com/v2/jobs' \
   -H 'Authorization: Bearer NDFjOTE3NGEtOWVm' \
   -F config='{
      "type": "transcription",
      "transcription_config": { "language": "en" },
      "fetch_data": { "url": "https://s3.us-east-2.amazonaws.com/bucketname/jqld_/20180804102000/profile.m4v" }
    }' \
   | jq
```

Some cloud storage solutions may require authentication. You can use the `auth_headers` property in the `fetch_data` section to provide the headers necessary to access the resource.

Notifications

The method described in the previous section assumes that polling is used to check on the status of the job, before making the call to retrieve the transcript. A more convenient method is to use notifications. This involves a callback to a web service that you control. An HTTP POST is then made from the Speechmatics service to your web server once the transcript is available.

The notification support offered in V1 has been extended and generalized in V2 to support a wider range of customer integration scenarios:

- A callback can be configured to *not* include any data attachments, so it can be used as a lightweight signal that results are ready to be fetched through the normal API without needing to poll.
- Multiple pieces of content can be sent as multiple attachments in one request, allowing any combination of the input(s) and output(s) of the job to be forwarded to another processing stage. Formatting options for outputs can be specified per attachment. The full details of the job are available as a content item, as well as being included in the json-v2 transcript format.
- You can setup multiple notifications to different endpoints: for instance you can send a jobinfo notificaiton to one service, and the transcript notification to another.
- Callbacks with a single attachment will send the content item as the HTTP request body, rather than using multipart mode. This allows writing an individual item to an object store like Amazon S3.
- HTTP PUT methods are now supported to allow uploading of content directly to an object store such as S3.
- A set of additional HTTP request headers can be specified:
    - To satisfy authentication / authorization requirements for systems that do not support auth tokens in query parameters.
    - To control behaviour of an object store or another existing service endpoint.
- Multiple callbacks can be specified per job.
    - This allows sending individual pieces of content to different URLs, eg. to allow uploading the audio and transcript to an object store as distinct objects for a downstream workflow.
    - It allows sending arbitrary combinations of the inputs/outputs to multiple destinations, to support a fanout workflow.

Callbacks will be invoked in parallel and so may complete in any order. If a downstream workflow depends on getting several items of content delivered as separate callbacks (eg. uploaded as separate items to S3), then the downstream processing logic will need to be robust to the ordering of uploads and the possibility that only some might succeed.

To ensure that the callbacks you receive come from Speechmatics you can apply a whitelist using the IP addresses 52.236.176.166 and 40.85.99.235.

> [info] Email notification
>
> Email notification is not currently supported.

Configuring the Callback

The callback is specified at job submission time using the `notification_config` config object. For example:

```
curl -L -X POST 'https://asr.api.speechmatics.comv2/jobs' \
   -H 'Authorization: Bearer NDFjOTE3NGEtOWVm' \
   --form data_file=@example.wav \
   --form config='{
     "type": "transcription",
     "transcription_config": { "language": "en" },
     "notification_config": [
       {
         "url": "https://collector.example.org/callback",
         "contents": [ "transcript", "data" ],
         "auth_headers": [
           "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhb"
         ]
       }
     ]
   }' \
   | jq
```

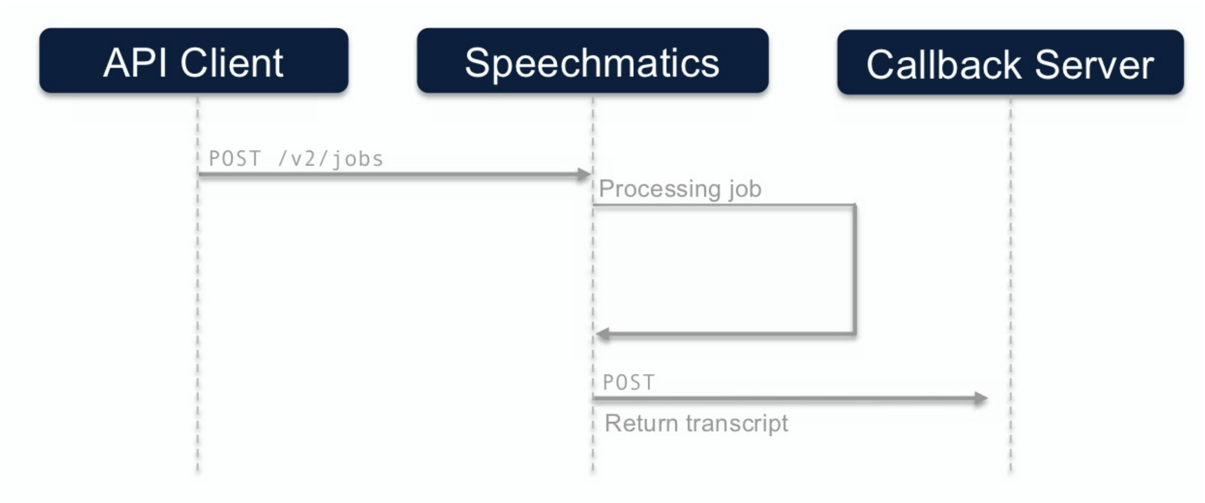
This example assumes you have implemented a `/callback` endpoint on host `collector.example.org` that listens for POST requests containing Speechmatics transcripts. These requests are only accepted if the auth token eyJ0eXAiOiJKV1QiLCJhb is used (note this is the auth token that *your* service accepts, *not* the Speechmatics auth token).

Accepting the Callback

You need to ensure that the service that you implement to receive the callback notification is capable of processing the Speechmatics transcript using the format that has been specified in the config JSON. When testing your integration you should check the error logs on your web service to ensure that notifications are being accepted and processed correclty.

Notification Call Flow

The call flow for the notification method looks like this:



Channel Diarization

Diarization is a way of dividing up an input stream into marked segments according to speaker identity. The V1 diarization feature (aka 'Speaker Diarization') where all input channels are mixed into a single stream for processing, is still available through the `transcription_config` object:

```
"transcription_config": {
    "language": "en",
    "diarization": "speaker"
}
```

The V2 API also supports Channel Diarization which can be used to add your own speaker or channel labels to the transcript. With Channel Diarization, multiple input channels are processed individually and collated into a single transcript. In order to use this method of diarization your input audio must have been transcoded into multiple channels or streams.

In order to use this feature you set the `diarization` property to `channel` and (optionally) specify the `channel_diarization_labels`:

```
"transcription_config": {
    "language": "en",
    "diarization": "channel",
    "channel_diarization_labels": [
      "Presenter",
      "Questions"
    ]
}
```

If you do not specify any labels then defaults will be used. The number of labels you use should be the same as the number of channels in your audio. Additional labels are ignored. When the transcript is returned a `channel` property for each word will indicate the speaker, for example:

```
"results": [
    {
    "type": "word",
    "end_time": 1.8,
    "start_time": 1.45,
    "channel": "Presenter"
    "alternatives": [
        {
        "display": { "direction": "ltr" },
        "language": "en",
        "content": "world",
        "confidence": 0.76
        }
    ]
    }
]
```

Custom Dictionary

The Custom Dictionary feature can be accessed through the `additional_vocab` property. This is a list of custom words or phrases that should be recognized. Custom Dictionary Sounds is an extension to this to allow alternative pronunciations to be specified in order to aid recognition, or provide for alternative transcriptions:

```
"transcription_config": {
    "language": "en",
    "additional_vocab": [
      {
        "content": "gnocchi",
        "sounds_like": [
          "nyohki",
          "nokey",
          "nochi"
        ]
      },
      {
        "content": "CEO",
        "sounds_like": [
          "C.E.O."
        ]
      },
      { "content": "financial crisis" }
    ]
}
```

You can specify up to 1000 words or phrases (per job) in your custom dictionary.

In the above example, the words *gnocchi* and *CEO* have pronunciations applied to them; the phrase *financial crisis* does not require a pronunciation. The `content` property represents how you want the word to be output in the transcript. This allows you to implement a simple profanity filter for example by using the `content` property to replace redacted words with "*" for instance.

Metadata and Job Tracking

It is now possible to attach richer metadata to a job using the `tracking` configuration. The tracking object contains the following properties:

Name	Type	Description	Notes
title	str	The title of the job.	[optional]
reference	str	External system reference.	[optional]
tags	list[str]		[optional]
details	object	Customer-defined JSON structure.	[optional]

This allows you to track the job through your own management workflow using whatever information is relevant to you.

```
curl -L -X POST 'https://asr.api.speechmatics.com/v2/jobs' \
   -H 'Authorization: Bearer NDFjOTE3NGEtOWVm' \
   --form data_file=@example.wav \
   --form config='{
     "type": "transcription",
     "transcription_config": { "language": "en" },
     "tracking": {
        "title": "ACME Q12018 Statement",
        "reference": "/data/clients/ACME/statements/segs/2018Q1-seg8",
        "tags": [ "quick-review", "segment" ],
        "details": {
           "client": "ACME Corp",
           "segment": 8,
           "seg_start": 963.201,
           "seg_end": 1091.481
         }
      }
   }' \
   | jq
```

Deleting a Job

The current retention period for jobs is 7 days; after this time jobs are removed. If you wish to save jobs you should import them into your own data store before they are removed.

You can remove jobs yourself once you no longer require them by explicitly deleting a job as follows:

```
curl -L -X DELETE 'https://asr.api.speechmatics.com/v2/jobs/yjbmf9kqub' \
   -H 'Authorization: Bearer NDFjOTE3NGEtOWVm' \
```

Troubleshooting

Basics

If you see errors when you submit a job and need to raise a support ticket, then please open a ticket by emailing support@speechmatics.com.

Please provide the Request-Id and X-Azure-Ref headers, and (if you have it) the Job ID. A response for a successful job should look like this:

```
HTTP/1.1 201 Created
Content-Length: 20
Content-Type: application/json
Request-Id: df6ec500191bf2c02ee42d519fbea34f
Strict-Transport-Security: max-age=15724800; includeSubDomains
X-Azure-Ref: 01mDmXAAAAABUtz6ZgS7QRI2F3f5B+19wTE9OMjFFREdFMDMxMgBhN2JjOWQ4MC02YjBiLTQ1NWEtYjE3MS01NGJkZmNiYWE0Y
Tk=
Date: Thu, 23 May 2019 08:59:08 GMT

{"id":"z2jfp1jwu1"}
```

In this example, the Request-Id is `df6ec500191bf2c02ee42d519fbea34f` and the Job ID is `z2jfp1jwu1`.

An error will be reported as a 4XX or 5XX status code; instead of the job ID an error message will be returned. The Request-Id can be included in the ticket to support to help support engineers troubleshoot the problem. Here's an example:

```
HTTP/1.1 401 Unauthorized
Content-Length: 179
Content-Type: text/html
Request-Id: 00adae505fa8cb7c7f24c6bf52c60048
Strict-Transport-Security: max-age=15724800; includeSubDomains
X-Azure-Ref: 0LWLmXAAAAAB2mDIdAu7kSZ/kKszi7eH4TE9OMjFFREdFMDMwOQBhN2JjOWQ4MC02YjBiLTQ1NWEtYjE3MS01NGJkZmNiYWE0Y
Tk=
Date: Thu, 23 May 2019 09:04:49 GMT

<html>
<head><title>401 Authorization Required</title></head>
<body>
<center><h1>401 Authorization Required</h1></center>
<hr><center>nginx/1.15.8</center>
</body>
</html>
```

Speechmatics ASR REST API

The Speechmatics Automatic Speech Recognition REST API is used to submit ASR jobs and receive the results. The only supported job type currently is transcription of audio files.

Version: 2.0.0

Contact information:
support@speechmatics.com

/jobs

POST

Summary: Create a new job.

Parameters

Name	Located in	Description	Required	Schema
Authorization	header	Customer API token	Yes	string
config	formData	JSON containing a `JobConfig` model indicating the type and parameters for the recognition job.	Yes	string
data_file	formData	The data file to be processed. Alternatively the data file can be fetched from a url specified in `JobConfig` .	No	file

Responses

Code	Description	Schema
201	OK	CreateJobResponse
400	Bad request	ErrorResponse
401	Unauthorized	ErrorResponse
403	Forbidden	ErrorResponse
500	Internal Server Error	ErrorResponse

GET

Summary: List all jobs.

Parameters

Name	Located in	Description	Required	Schema
Authorization	header	Customer API token	Yes	string

Responses

Code	Description	Schema
200	OK	RetrieveJobsResponse

401	Unauthorized	ErrorResponse
500	Internal Server Error	ErrorResponse

/jobs/{jobid}

GET

Summary: Get job details, including progress and any error reports.

Parameters

Name	Located in	Description	Required	Schema
Authorization	header	Customer API token	Yes	string
jobid	path	ID of the job.	Yes	string

Responses

Code	Description	Schema
200	OK	RetrieveJobResponse
401	Unauthorized	ErrorResponse
404	Not found	ErrorResponse
500	Internal Server Error	ErrorResponse

DELETE

Summary: Delete a job and remove all associated resources.

Parameters

Name	Located in	Description	Required	Schema
Authorization	header	Customer API token	Yes	string
jobid	path	ID of the job to delete.	Yes	string

Responses

Code	Description	Schema
200	The job that was deleted.	DeleteJobResponse
401	Unauthorized	ErrorResponse
404	Not found	ErrorResponse
500	Internal Server Error	ErrorResponse

/jobs/{jobid}/data

GET

Summary: Get the data file used as input to a job.

Parameters

Name	Located in	Description	Required	Schema
Authorization	header	Customer API token	Yes	string
jobid	path	ID of the job.	Yes	string

Responses

Code	Description	Schema
200	OK	file
401	Unauthorized	ErrorResponse
404	Not found	ErrorResponse
500	Internal Server Error	ErrorResponse

/jobs/{jobid}/transcript

GET

Summary: Get the transcript for a transcription job.

Parameters

Name	Located in	Description	Required	Schema
Authorization	header	Customer API token	Yes	string
jobid	path	ID of the job.	Yes	string
format	query	The transcripton format, either `txt` or `json-v2` (the default).	No	string

Responses

Code	Description	Schema
200	OK	RetrieveTranscriptResponse
401	Unauthorized	ErrorResponse
404	Not found	ErrorResponse
500	Internal Server Error	ErrorResponse

ErrorResponse

Name	Type	Description	Required
code	integer	The HTTP status code.	Yes
error	string	The error message.	Yes
detail	string	The details of the error.	No

TrackingData

Name	Type	Description	Required
title	string	The title of the job.	No
reference	string	External system reference.	No
tags	[string]	A set of keywords	No
details	object	Customer-defined JSON structure.	No

DataFetchConfig

Name	Type	Description	Required
url	string		Yes
auth_headers	[string]	A list of additional headers to be added to the input fetch request when using http or https. This is intended to support authentication or authorization, for example by supplying an OAuth2 bearer token.	No

TranscriptionConfig

Name	Type	Description	Required
language	string	Language model to process the audio input, normally specified as an ISO language code	Yes
additional_vocab	[object]	List of custom words or phrases that should be recognized. Alternative pronunciations can be specified to aid recognition.	No
diarization	string	Specify whether speaker or channel labels are added to the transcript. The default is `none`. - none: no speaker or channel labels are added. - speaker: speaker attribution is performed based on acoustic matching;all input channels are mixed into a single stream for processing. - channel: multiple input channels are processed individually and collated into a single transcript.	No
channel_diarization_labels	[string]	Transcript labels to use when using collating separate input channels.	No

NotificationConfig

Name	Type	Description	Required
url	string	The url to which a notification message will be sent upon completion of the job. The job `id` and `status` are added as query parameters, and any combination of the job inputs and outputs can be included by listing them in `contents`. If `contents` is empty, the body of the request will be empty. If only one item is listed, it will be sent as the body of the request with `Content-Type` set to an appropriate value such as `application/octet-stream` or `application/json`. If multiple items are listed they will be sent as named file attachments using the multipart content type. If `contents` is not specified, the `transcript` item will be sent as a file attachment named `data_file`, for backwards compatibility. If the job was rejected or failed during processing, that will be indicated by the status, and any output items that are not available as a result will be omitted. The body formatting rules will still be followed as if all items were available.	Yes

		The user-agent header is set to `Speechmatics API V2` in all cases.	
contents	[string]	Specifies a list of items to be attached to the notification message. When multiple items are requested, they are included as named file attachments. Possible options: ● \`jobinfo\`: A summary of the job. See JobInfo. ● \`transcript\`: The transcript in the default format. ● \`transcript.json-v2\`: The transcript in \`json-v2\` format. ● \`transcript.txt\`: The transcript in \`txt\` format. ● \`data\`: The audio file submitted for the job.	No
method	string	The method to be used with http and https urls. The default is POST.	No
auth_headers	[string]	A list of additional headers to be added to the notification request when using http or https. This is intended to support authentication or authorization, for example by supplying an OAuth2 bearer token.	No

JobConfig

JSON object that contains various groups of job configuration parameters. Based on the value of `type`, a type-specific object such as `transcription_config` is required to be present to specify all configuration settings or parameters needed to process the job inputs as expected.

If the results of the job are to be forwarded on completion, `notification_config` can be provided with a list of callbacks to be made; no assumptions should be made about the order in which they will occur.

Customer specific job details or metadata can be supplied in `tracking`, and this information will be available where possible in the job results and in callbacks.

Name	Type	Description	Required
type	string		Yes
fetch_data	DataFetchConfig		No
transcription_config	TranscriptionConfig		No
notification_config	[NotificationConfig]		No
tracking	TrackingData		No

CreateJobResponse

Name	Type	Description	Required
id	string	The unique ID assigned to the job. Keep a record of this for later retrieval of your completed job.	Yes

JobDetails

Name	Type	Description	Required
created_at	dateTime	The UTC date time the job was created.	Yes
data_name	string	Name of the data file submitted for job.	Yes
duration	integer	The file duration (in seconds).	Yes
id	string	The unique id assigned to the job.	Yes
		The status of the job. ***queued*** *- The job is waiting to run.*	

status	string	`running` - The job is actively running. ***`done`*** *- The job completed successfully.* `rejected` - The job was accepted at first, but later could not be processed by the transcriber. ***`deleted`*** *- The user deleted the job.* `expired` - The system deleted the job. Usually because the job was in the `done` state for a very long time.	Yes
config	JobConfig		Yes

RetrieveJobsResponse

Name	Type	Description	Required
jobs	[JobDetails]		Yes

RetrieveJobResponse

Name	Type	Description	Required
job	JobDetails		Yes

DeleteJobResponse

Name	Type	Description	Required
job	JobDetails		Yes

JobInfo

Summary information about an ASR job, to support identification and tracking.

Name	Type	Description	Required
created_at	dateTime	The UTC date time the job was created.	Yes
data_name	string	Name of data file submitted for job.	Yes
duration	integer	The data file audio duration (in seconds).	Yes
id	string	The unique id assigned to the job.	Yes
tracking	TrackingData		No

RecognitionMetadata

Summary information about the output from an ASR job, comprising the job type and configuration parameters used when generating the output.

Name	Type	Description	Required
created_at	dateTime	The UTC date time the transcription output was created.	Yes
type	string		Yes
transcription_config	TranscriptionConfig		No

RecognitionDisplay

Name	Type	Description	Required
direction	string		Yes

RecognitionAlternative

List of possible job output item values, ordered by likelihood.

Name	Type	Description	Required
content	string		Yes
confidence	float		Yes
language	string		Yes
display	RecognitionDisplay		No
speaker	string		No

RecognitionResult

An ASR job output item. The primary item types are `word` and `punctuation` . Other item types may be present, for example to provide semantic information of different forms.

Name	Type	Description	Required
channel	string		No
start_time	float		Yes
end_time	float		Yes
type	string	New types of items may appear without being requested; unrecognised item types can be ignored.	Yes
alternatives	[RecognitionAlternative]		Yes

RecognitionOutput

The `results` element contains a list of ASR job output items, ordered by start time and sub-ordered by end time and type. The primary item types are `word` and `punctuation` ; the most likely value for each of these items can be combined in the order of occurence to form a linear transcript. Items may overlap in time if there are multiple channels. Other item types may be present which overlap with the primary items and each other, for example to provide different forms of semantic information as time-based annotations.

Name	Type	Description	Required
metadata	RecognitionMetadata		Yes
results	[RecognitionResult]		Yes

RetrieveTranscriptResponse

Name	Type	Description	Required
format	string	Speechmatics JSON transcript format version number.	Yes

job	JobInfo		Yes
output	RecognitionOutput		Yes