Yandex Speechkit SDK

Release 1.3.0

Tikhon Petrishchev

May 09, 2023

CONTENTS:

Speechkit is python SDK for Yandex SpeechKit API.

CHAPTER

ONE

INSTALLATION

Assuming that you have Python and virtualenv installed, set up your environment and install the required dependencies like this:

```
$ git clone https://github.com/TikhonP/yandex-speechkit-lib-python.git
$ cd yandex-speechkit-lib-python
$ virtualenv venv
...
$ . venv/bin/activate
$ python -m pip install -r requirements.txt
$ python -m pip install .
```

Or you can install the library using pip:

```
python -m pip install speechkit
```

CHAPTER

TWO

API REFERENCE

2.1 speechkit package

speechkit Python SDK for using Yandex Speech recognition and synthesis.

class speechkit.**DataStreamingRecognition**(*session*, *language_code=None*, *model=None*, *profanity_filter=None*, *partial_results=None*, *single_utterance=None*, *audio_encoding=None*, *sample_rate_hertz=None*, *raw_results=None*)

Bases: `object`

Data streaming mode allows you to simultaneously send audio for recognition and get recognition results over the same connection.

Unlike other recognition methods, you can get intermediate results while speech is in progress. After a pause, the service returns final results and starts recognizing the next utterance.

After receiving the message with the recognition settings, the service starts a recognition session. The following limitations apply to each session:

1. You can't send audio fragments too often or too rarely. The time between messages to the service should be approximately the same as the duration of the audio fragments you send, but no more than 5 seconds. For example, send 400 ms of audio for recognition every 400 ms.
2. Maximum duration of transmitted audio for the entire session: 5 minutes.
3. Maximum size of transmitted audio data: 10 MB.

To use this type of recognition, you need to create function that yields bytes data

Example

```
>>> chunk_size = 4000
>>> session = Session.from_jwt("jwt")
>>> data_streaming_recognition = DataStreamingRecognition(
...     session,
...     language_code='ru-RU',
...     audio_encoding='LINEAR16_PCM',
...     session=8000,
...     partial_results=False,
...     single_utterance=True,
... )
...
>>> def gen_audio_from_file_function():
...     with open('/path/to/pcm_data/speech.pcm', 'rb') as f:
```

(continues on next page)

(continued from previous page)

```
...         data = f.read(chunk_size)
...         while data != b'':
...             yield data
...             data = f.read(chunk_size)
...
>>> for i in data_streaming_recognition.recognize(gen_audio_capture_function):
...     print(i)  # (['text'], final_flag, end_of_utterance_flag)
...
```

Read more about streaming recognition in Yandex streaming recognition docs

Initialize *speechkit.DataStreamingRecognition*

Parameters

- **session** (speechkit._auth.Session) – Session instance for auth
- **language_code** (*string | None*) – The language to use for recognition. Acceptable values: *ru-ru* (case-insensitive, used by default): Russian, *en-us* (case-insensitive): English, *tr-tr* (case-insensitive): Turkish.
- **model** (*string | None*) – The language model to be used for recognition. The closer the model is matched, the better the recognition result. You can only specify one model per request. Default value: *general*.
- **profanity_filter** (*boolean | None*) – The profanity filter. Acceptable values: *true*: Exclude profanity from recognition results, *false* (default): Do not exclude profanity from recognition results.
- **partial_results** (*boolean | None*) – The intermediate results filter. Acceptable values: *true*: Return intermediate results (part of the recognized utterance). For intermediate results, final is set to false, *false* (default): Return only the final results (the entire recognized utterance).
- **single_utterance** (*boolean | None*) – Flag that disables recognition after the first utterance. Acceptable values: *true*: Recognize only the first utterance, stop recognition, and wait for the user to disconnect, *false* (default): Continue recognition until the end of the session.
- **audio_encoding** (*string | None*) – The format of the submitted audio. Acceptable values: *LINEAR16_PCM*: LPCM with no WAV header, *OGG_OPUS* (default): OggOpus format.
- **sample_rate_hertz** (*integer | None*) – (int64) The sampling frequency of the submitted audio. Required if format is set to LINEAR16_PCM. Acceptable values: *48000* (default): Sampling rate of 48 kHz, *16000*: Sampling rate of 16 kHz, *8000*: Sampling rate of 8 kHz.
- **raw_results** (*boolean | None*) – Flag that indicates how to write numbers. *true*: In words. *false* (default): In figures.

recognize(*gen_audio_function*, **args*, ***kwargs*)

Recognize streaming data, gen_audio_function must yield audio data with parameters given in init. Pass args and kwargs to pass it into `gen_audio_function()`.

Parameters

gen_audio_function (*function*) – Function generates audio data

Returns

yields tuple, where first element is list of alternatives text, second final (boolean) flag, third

endOfUtterance (boolean) flag, ex. (['text'], False, False)

Return type
tuple

recognize_raw(*gen_audio_function*, **args*, ***kwargs*)

Recognize streaming data, gen_audio_function must yield audio data with parameters given in init. Answer type read in Yandex Docs. Pass args and kwargs to pass it into `gen_audio_function()`.

Parameters
gen_audio_function (*function*) – Function generates audio data

Returns
Yields recognized data in raw format

Return type
speechkit._recognition.yandex.cloud.ai.stt.v2.stt_service_pb2.StreamingRecognitionResponse

class speechkit.**RecognitionLongAudio**(*session*, *service_account_id*, *aws_bucket_name=None*, *aws_credentials_description='Default AWS credentials created by `speechkit` python SDK'*, *aws_region_name='ru-central1'*, *aws_access_key_id=None*, *aws_secret=None*)

Bases: `object`

Long audio fragment recognition can be used for multi-channel audio files up to 1 GB. To recognize long audio fragments, you need to execute 2 requests:

- Send a file for recognition.
- Get recognition results.

Example

```
>>> recognizeLongAudio = RecognitionLongAudio(session, '<service_account_id>')
>>> recognizeLongAudio.send_for_recognition('file/path')
>>> if recognizeLongAudio.get_recognition_results():
...     data = recognizeLongAudio.get_data()
...
>>> recognizeLongAudio.get_raw_text()
...'raw recognized text'
```

Initialize *speechkit.RecognitionLongAudio*

Parameters

- **session** (speechkit.Session) – Session instance for auth
- **service_account_id** (*string*) – Yandex Cloud Service account ID
- **aws_bucket_name** (*string*) – Optional AWS bucket name
- **aws_credentials_description** (*string*) – AWS credentials description
- **aws_region_name** (*string*) – AWS region name
- **aws_access_key_id** (*string*) – Optional AWS access key. Can be got by *.get_aws_credentials*. If None will be generated automatically
- **aws_secret** (*string*) – Optional AWS secret. Can be got by *.get_aws_credentials*. If None will be generated automatically

static get_aws_credentials(*session*, *service_account_id*, *aws_credentials_description='Default AWS credentials created by `speechkit` python SDK'*)

Get AWS credentials from yandex cloud

Parameters

- **session** (speechkit.Session) – Session instance for auth
- **service_account_id** (*string*) – Yandex Cloud Service account ID
- **aws_credentials_description** (*string*) – AWS credentials description

Returns
tuple with strings (access_key_id, secret)

get_data()

Get the response. Use *speechkit.RecognitionLongAudio.get_recognition_results()* first to store _answer_data

Contain a list of recognition results (*chunks[]*).

Returns

None if text not found ot Each result in the chunks[] list contains the following fields:

- *alternatives[]*: List of recognized text alternatives. Each alternative contains the following fields:
  - *words[]*: List of recognized words:
    - ***startTime*: Time stamp of the beginning of the word in the recording. An error of 1-2 seconds** is possible.
    - *endTime*: Time stamp of the end of the word. An error of 1-2 seconds is possible.
    - ***word*: Recognized word. Recognized numbers are written in words (for example, twelve rather** than 12).
    - *confidence*: This field currently isn't supported. Don't use it.
    - ***text*: Full recognized text. By default, numbers are written in figures. To recognition** the entire text in words, specify true in the raw_results field.
    - *confidence*: This field currently isn't supported. Don't use it.
- *channelTag*: Audio channel that recognition was performed for.

Return type
list | None

get_raw_text()

Get raw text from _answer_data data

Returns
Text

Return type
string

get_recognition_results()

Monitor the recognition results using the received ID. The number of result monitoring requests is limited, so consider the recognition speed: it takes about 10 seconds to recognize 1 minute of single-channel audio.

Returns
State of recognition is done or not

Return type
boolean

send_for_recognition(*file_path*, ***kwargs*)

Send a file for recognition

Parameters

- **file_path** (*string*) – Path to input file
- **folder_id** (*string*) – ID of the folder that you have access to. Don't specify this field if you make a request on behalf of a service account.
- **languageCode** (*string*) – The language that recognition will be performed for. Only Russian is currently supported (*ru-RU*).
- **model** (*string*) – The language model to be used for recognition. Default value: *general*.
- **profanityFilter** (*boolean*) – The profanity filters.
- **audioEncoding** (*string*) – The format of the submitted audio. Acceptable values:
  - *LINEAR16_PCM*: LPCM with no WAV _header.
  - *OGG_OPUS* (default): OggOpus format.
- **sampleRateHertz** (*integer*) – The sampling frequency of the submitted audio. Required if format is set to LINEAR16_PCM. Acceptable values:
  - *48000* (default): Sampling rate of 48 kHz.
  - *16000*: Sampling rate of 16 kHz.
  - *8000*: Sampling rate of 8 kHz.
- **audioChannelCount** (*integer*) – The number of channels in LPCM files. By default, *1*. Don't use this field for OggOpus files.
- **rawResults** (*boolean*) – Flag that indicates how to write numbers. *true*: In words. *false* (default): In figures.

Return type
None

class speechkit.**Session**(*auth_type*, *credential*, *folder_id*, *x_client_request_id_header=False*, *x_data_logging_enabled=False*)

Bases: `object`

Class provides yandex API authentication.

Stores credentials for given auth method

Parameters

- **auth_type** (*string*) – Type of auth may be *Session.IAM_TOKEN()* or *Session.API_KEY()*
- **folder_id** (*string | None*) – Id of the folder that you have access to. Don't specify this field if you make a request on behalf of a service account.
- **credential** (*string*) – Auth key iam or api key

- **x_client_request_id_header** (*boolean*) – include x-client-request-id. *x-client-request-id* is a unique request ID. It is generated using uuid. Send this ID to the technical support team to help us find a specific request in the system and assist you. To get x_client_request_id_header use *Session.get_x_client_request_id()* method.
- **x_data_logging_enabled** (*boolean*) – A flag that allows data passed by the user in the request to be saved. By default, we do not save any audio or text that you send. If you pass the true value in this header, your data is saved. This data, along with the request ID, will help the Yandex technical support team solve your problem.

API_KEY = 'api_key'

Api key if api-key auth, value: 'api_key'

IAM_TOKEN = 'iam_token'

Iam_token if iam auth, value: 'iam_token'

property auth_method

Get auth method it may be *Session.IAM_TOKEN* or *Session.API_KEY*

classmethod from_api_key(*api_key*, *folder_id=None*, *x_client_request_id_header=False*, *x_data_logging_enabled=False*)

Creates session from api key

Parameters

- **api_key** (*string*) – Yandex Cloud Api-Key
- **folder_id** (*string | None*) – Id of the folder that you have access to. Don't specify this field if you make a request on behalf of a service account.
- **x_client_request_id_header** (*boolean*) – include x-client-request-id. *x-client-request-id* is a unique request ID. It is generated using uuid. Send this ID to the technical support team to help us find a specific request in the system and assist you. To get x_client_request_id_header use *Session.get_x_client_request_id()* method.
- **x_data_logging_enabled** (*boolean*) – A flag that allows data passed by the user in the request to be saved. By default, we do not save any audio or text that you send. If you pass the true value in this header, your data is saved. This data, along with the request ID, will help the Yandex technical support team solve your problem.

Returns

Session instance

Return type

Session

classmethod from_jwt(*jwt_token*, *folder_id=None*, *x_client_request_id_header=False*, *x_data_logging_enabled=False*)

Creates Session from JWT token

Parameters

- **jwt_token** (*string*) – JWT
- **folder_id** (*string | None*) – Id of the folder that you have access to. Don't specify this field if you make a request on behalf of a service account.
- **x_client_request_id_header** (*boolean*) – include x-client-request-id. *x-client-request-id* is a unique request ID. It is generated using uuid. Send this ID

to the technical support team to help us find a specific request in the system and assist you. To get x_client_request_id_header use *Session.get_x_client_request_id()* method.

- **x_data_logging_enabled** (*boolean*) – A flag that allows data passed by the user in the request to be saved. By default, we do not save any audio or text that you send. If you pass the true value in this header, your data is saved. This data, along with the request ID, will help the Yandex technical support team solve your problem.

Returns
Session instance

Return type
Session

classmethod from_yandex_passport_oauth_token(*yandex_passport_oauth_token*, *folder_id*, *x_client_request_id_header=False*, *x_data_logging_enabled=False*)

Creates Session from oauth token Yandex account

Parameters

- **yandex_passport_oauth_token** (*string*) – OAuth token from Yandex.OAuth
- **folder_id** (*string*) – Id of the folder that you have access to. Don't specify this field if you make a request on behalf of a service account.
- **x_client_request_id_header** (*boolean*) – include x-client-request-id. *x-client-request-id* is a unique request ID. It is generated using uuid. Send this ID to the technical support team to help us find a specific request in the system and assist you. To get x_client_request_id_header use *Session.get_x_client_request_id()* method.
- **x_data_logging_enabled** (*boolean*) – A flag that allows data passed by the user in the request to be saved. By default, we do not save any audio or text that you send. If you pass the true value in this header, your data is saved. This data, along with the request ID, will help the Yandex technical support team solve your problem.

Returns
Session instance

Return type
Session

get_x_client_request_id()

Get generated x_client_request_id value, if enabled on init, else *None*

property header

Authentication header.

Returns
Dict in format *{'Authorization': 'Bearer or Api-Key {iam or api_key}'}*

Return type
dict

property streaming_recognition_header

Authentication header for streaming recognition

Returns
Tuple in format *('authorization', 'Bearer or Api-Key {iam or api_key}')*

Return type
tuple

class speechkit.**ShortAudioRecognition**(*session*)

Bases: object

Short audio recognition ensures fast response time and is suitable for single-channel audio of small length.
Audio requirements:

1. Maximum file size: 1 MB.
2. Maximum length: 30 seconds.
3. Maximum number of audio channels: 1.

If your file is larger, longer, or has more audio channels, use *speechkit.RecognitionLongAudio*.

Initialization *speechkit.ShortAudioRecognition*

Parameters
session (speechkit.Session) – Session instance for auth

recognize(*data*, ***kwargs*)

Recognize text from BytesIO data given, which is audio

Parameters

- **data** (*io.BytesIO, bytes*) – Data with audio samples to recognize
- **lang** (*string*) – The language to use for recognition. Acceptable values:
  - *ru-RU* (by default) — Russian.
  - *en-US* — English.
  - *tr-TR* — Turkish.
- **topic** (*string*) – The language model to be used for recognition. Default value: *general*.
- **profanityFilter** (*boolean*) – This parameter controls the profanity filter in recognized speech.
- **format** (*string*) – The format of the submitted audio. Acceptable values:
  - *lpcm* — LPCM with no WAV _header.
  - *oggopus* (default) — OggOpus.
- **sampleRateHertz** (*string*) – The sampling frequency of the submitted audio. Used if format is set to lpcm. Acceptable values:
  - *48000* (default) — Sampling rate of 48 kHz.
  - *16000* — Sampling rate of 16 kHz.
  - *8000* — Sampling rate of 8 kHz.
- **folderId** (*string*) – ID of the folder that you have access to. Don't specify this field if you make a request on behalf of a service account.

Returns
The recognized text

Return type
string

class speechkit.**SpeechSynthesis**(*session*)

Bases: object

Generates speech from received text.

Initialize *speechkit.SpeechSynthesis*

Parameters

session (speechkit.Session) – Session instance for auth

synthesize(*file_path*, ***kwargs*)

Generates speech from received text and saves it to file

Parameters

- **file_path** (*string*) – The path to file where store data
- **text** (*string*) – UTF-8 encoded text to be converted to speech. You can only use one *text* and *ssml* field. For homographs, place a + before the stressed vowel. For example, *contr+ol* or *def+ect*. To indicate a pause between words, use -. Maximum string length: 5000 characters.
- **ssml** (*string*) – Text in SSML format to be converted into speech. You can only use one text and ssml fields.
- **lang** (*string*) – Language. Acceptable values:
  - *ru-RU* (default) — Russian.
  - *en-US* — English.
  - *tr-TR* — Turkish.
- **voice** (*string*) – Preferred speech synthesis voice from the list. Default value: *oksana*.
- **speed** (*string*) – Rate (speed) of synthesized speech. The rate of speech is set as a decimal number in the range from 0.1 to 3.0. Where:
  - *3.0* — Fastest rate.
  - *1.0* (default) — Average human speech rate.
  - *0.1* — Slowest speech rate.
- **format** (*string*) – The format of the synthesized audio. Acceptable values:
  - *lpcm* — Audio file is synthesized in LPCM format with no WAV _header. Audio properties:
    - Sampling — 8, 16, or 48 kHz, depending on the value of the *sample_rate_hertz* parameter.
    - Bit depth — 16-bit.
    - Byte order — Reversed (little-endian).
    - Audio data is stored as signed integers.
  - ***oggopus* (default) — Data in the audio file is encoded using the OPUS audio codec and compressed using**
    the OGG container format (OggOpus).
- **sample_rate_hertz** – The sampling frequency of the synthesized audio. Used if format is set to lpcm. Acceptable values: * *48000* (default): Sampling rate of 48 kHz. * *16000*: Sampling rate of 16 kHz. * *8000*: Sampling rate of 8 kHz.

- **folderId** (*string*) – ID of the folder that you have access to. Required for authorization with a user account (see the UserAccount resource). Don't specify this field if you make a request on behalf of a service account.

synthesize_stream(***kwargs*)

Generates speech from received text and return `io.BytesIO()` object with data.

Parameters

- **text** (*string*) – UTF-8 encoded text to be converted to speech. You can only use one *text* and *ssml* field. For homographs, place a + before the stressed vowel. For example, *contr+ol* or *def+ect*. To indicate a pause between words, use -. Maximum string length: 5000 characters.
- **ssml** (*string*) – Text in SSML format to be converted into speech. You can only use one text and ssml fields.
- **lang** (*string*) – Language. Acceptable values:
  - *ru-RU* (default) — Russian.
  - *en-US* — English.
  - *tr-TR* — Turkish.
- **voice** (*string*) – Preferred speech synthesis voice from the list. Default value: *oksana*.
- **speed** (*string*) – Rate (speed) of synthesized speech. The rate of speech is set as a decimal number in the range from 0.1 to 3.0. Where:
  - *3.0* — Fastest rate.
  - *1.0* (default) — Average human speech rate.
  - *0.1* — Slowest speech rate.
- **format** (*string*) – The format of the synthesized audio. Acceptable values:
  - *lpcm* — Audio file is synthesized in LPCM format with no WAV _header. Audio properties:
    - Sampling — 8, 16, or 48 kHz, depending on the value of the *sample_rate_hertz* parameter.
    - Bit depth — 16-bit.
    - Byte order — Reversed (little-endian).
    - Audio data is stored as signed integers.
  - ***oggopus* (default) — Data in the audio file is encoded using the OPUS audio codec and compressed using** the OGG container format (OggOpus).
- **sampleRateHertz** (*string*) – The sampling frequency of the synthesized audio. Used if format is set to lpcm. Acceptable values:
  - *48000* (default): Sampling rate of 48 kHz.
  - *16000*: Sampling rate of 16 kHz.
  - *8000*: Sampling rate of 8 kHz.

- **folderId** (*string*) – ID of the folder that you have access to. Required for authorization with a user account (see the UserAccount resource). Don't specify this field if you make a request on behalf of a service account.

2.2 speechkit.auth module

Utilities for Yandex Cloud authorisation IAM token Api-Key

speechkit.auth.**generate_jwt**(*service_account_id*, *key_id*, *private_key*, *exp_time=360*)

Generating JWT token for authorisation

Parameters

- **service_account_id** (*string*) – The ID of the service account whose key the JWT is signed with.
- **key_id** (*string*) – The ID of the Key resource belonging to the service account.
- **private_key** (*bytes*) – Private key given from Yandex Cloud console in bytes
- **exp_time** (*integer*) – Optional. The token expiration time delta in seconds. The expiration time must not exceed the issue time by more than one hour, meaning exp_time 3600. Default 360

Returns

JWT token

Return type

string

speechkit.auth.**get_api_key**(*yandex_passport_oauth_token=None*, *service_account_id=None*, *description='Default Api-Key created by `speechkit` python SDK'*)

Creates an API key for the specified service account.

Parameters

- **yandex_passport_oauth_token** (*string*) – OAuth token from Yandex OAuth
- **service_account_id** (*string*) – The ID of the service account whose key the Api-Key is signed with.
- **description** (*string*) – Description for api-key. Optional.

Returns

Api-Key

Return type

string

speechkit.auth.**get_iam_token**(*yandex_passport_oauth_token=None*, *jwt_token=None*)

Creates an IAM token for the specified identity. Getting IAM for Yandex account

Parameters

- **yandex_passport_oauth_token** (*string*) – OAuth token from Yandex OAuth
- **jwt_token** (*string*) – Json Web Token, can be generated by `speechkit.generate_jwt()`

Returns

IAM token

Return type

string

2.3 speechkit.utils module

Utilities functions, tht allow to use different api methods.

speechkit.utils.**list_of_service_accounts**(*session*, ***kwargs*)

Retrieves the list of ServiceAccount resources in the specified folder.

Parameters

- **session** (speechkit._auth.Session) – Session instance for auth
- **kwargs** (*dict*) – Additional parameters

Returns

List of dict with data of services accounts

Return type

list[dict]

2.4 speechkit.exceptions module

Basic speechkit exceptions.

exception speechkit.exceptions.**RequestError**(*answer: dict*, **args*)

Bases: Exception

Exception raised for errors while yandex api request

CHAPTER

THREE

SOME SHORTCUTS DOR YC CONSOLE TOOL

Get FOLDER_ID: FOLDER_ID=$(yc config get folder-id)

Create service-account: yc iam service-account create –name admin

Get id of service-account: SA_ID=$(yc iam service-account get –name admin –format json | jq .id -r)

Assign a role to the admin service account using its ID: yc resource-manager folder add-access-binding –id $FOLDER_ID –role admin –subject serviceAccount:$SA_ID

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

S

INDEX