Package ‘mscstts’

October 28, 2020

Type Package
Version 0.6.2
Title R Client for the Microsoft Cognitive Services 'Text-to-Speech' REST API
Description R Client for the Microsoft Cognitive Services 'Text-to-Speech' REST API, including voice synthesis. A valid account must be registered at the Microsoft Cognitive Services website <https://azure.microsoft.com/services/cognitive-services/> in order to obtain a (free) API key. Without an API key, this package will not work properly.
License GPL-3
Encoding UTF-8
LazyData true
Suggests knitr, rmarkdown, covr, testthat (>= 2.1.0), rvest, xml2, zoo
URL https://github.com/muschellij2/mscstts
BugReports https://github.com/muschellij2/mscstts/issues
RoxygenNote 7.1.1
Imports httr, jsonlite, tuneR
Depends R (>= 2.10)
NeedsCompilation no
Author John Muschelli [aut, cre] (<https://orcid.org/0000-0001-6469-1750>)
Maintainer John Muschelli <muschellij2@gmail.com>
Repository CRAN
Date/Publication 2020-10-28 15:50:02 UTC

R topics documented:

ms_create_ssml .................................................. 2
ms_get_tts_key .................................................. 3
ms_create_ssml

Create SSML for Passing to Synthesize

Description
Create SSML for Passing to Synthesize

Usage
```r
ms_create_ssml(
  script,
  voice = NULL,
  gender = c("Female", "Male"),
  language = "en-US",
  escape = FALSE
)
```

Arguments
- **script**: A character vector of lines to be spoken
- **voice**: full voice name, usually from `ms_language_to_ms_name`. Will override language and gender.
- **gender**: Sex of the Speaker
- **language**: Language to be spoken, must be from `ms_language_codes`
- **escape**: Should non-standard characters be substituted? Should not be used if `script` has SSML tags
- **token**: An authentication token, base-64 encoded usually from `ms_get_tts_token`. If not provided, will be created from `ms_get_tts_token`
- **api_key**: Microsoft Cognitive Services API key, if token is not provided.
- **region**: Subscription region for your key. See [https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/overview](https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/overview)
### ms_get_tts_key

**Value**

A character string of the text and SSML markup

**Examples**

```r
code(ms_create_ssml("hey I really like things & dogs", escape = TRUE))
code(ms_create_ssml("hey I really like things"))
code(ms_create_ssml("hey I <emphasis level="strong">really like</emphasis> things'")
code(ms_create_ssml("hey I <emphasis level="strong">really like</emphasis> things'", escape = TRUE))
```

---

**ms_get_tts_key**  
Find API Key for Microsoft Text To Speech (TTS) or Cognitive Services

**Description**

Determines if `option(ms_tts_key)` or `option(ms_tts_key)` is set. If not, it stops and returns an error. If so, returns the value.

**Usage**

```r
code(ms_get_tts_key(api_key = NULL, error = TRUE))
code(ms_have_tts_key(api_key = NULL))
code(ms_set_tts_key(api_key))
code(ms_valid_tts_key(api_key = NULL, region = ms_regions()))
```

**Arguments**

- `api_key`  
  Microsoft Cognitive Services API key
- `error`  
  Should the function error if `api_key = NULL`?
- `region`  
  Subscription region for your key. See [https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/overview](https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/overview)

**Value**

API key

**Note**

You can either set the API key using `option(ms_tts_key)` or have it accessible by `api_key = Sys.getenv("MS_TTS_API_KEY")`, or `api_key = Sys.getenv("MS_TTS_API_KEY1")`, or `api_key = Sys.getenv("MS_TTS_API_KEY2")`
**ms_get_tts_token**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ms_get_tts_token</code></td>
<td>Get Microsoft Text To Speech (TTS) or Cognitive Services Token from API Key</td>
<td><code>ms_get_tts_token(api_key = NULL, region = ms_regions())</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>ms_auth_url(region = ms_regions())</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>ms_expired_token(token = NULL)</code></td>
</tr>
</tbody>
</table>

**Arguments**

- **api_key**: Microsoft Cognitive Services API key
- **region**: Subscription region for your key. See [https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/overview#reference-docs](https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/overview#reference-docs)
- **token**: An authentication of class `token`, likely from `ms_get_tts_token`

**Value**

A list of the request, and token

**Examples**

```r
if (ms_valid_tts_key()) {
  token = ms_get_tts_token()
}
```
**ms_language_to_ms_name**

*Language and Gender to Microsoft Voice Name*

**Description**

Language and Gender to Microsoft Voice Name

**Usage**

```r
ms_language_to_ms_name(language = "en-US", gender = c("Female", "Male"))
```

```r
ms_validate_language_gender(language = "en-US", gender = c("Female", "Male"))
```

**Arguments**

- `language`: A language code, see `ms_language_codes`
- `gender`: Either Male or Female, not all languages support both genders

**Value**

A character string of the name of the voice

**Examples**

```r
ms_language_to_ms_name()
ms_validate_language_gender()
```

---

**ms_list_voices**

*List Voices*

**Description**

List Voices

**Usage**

```r
ms_list_voices(token = NULL, api_key = NULL, region = NULL, ...)
```

**Arguments**

- `token`: An authentication token, base-64 encoded usually from `ms_get_tts_token`. If not provided, will be created from `ms_get_tts_token`
- `api_key`: Microsoft Cognitive Services API key, if token is not provided.
- `region`: Subscription region for your key. See [https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/overview](https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/overview)
- `...`: Additional arguments to send to GET
ms_locales

Value
A data.frame of the names and their long names.

Examples
```r
## Not run:
if (ms_have_tts_key()) {
  res = ms_list_voices()
}

## End(Not run)

## Not run:
res = ms_list_voices(region = "eastus")
## End(Not run)
```

---

**ms_locales**  
*Locales for Microsoft Translate*

Description
Locales for Microsoft Translate

Usage
- `ms_locales()`
- `ms_locale_df()`
- `ms_languages()`
- `ms_language_codes()`

Value
A list of Locales and the Speech Voices

Note
This is generated from the JSON output of `ms_list_voices`

Examples
- `ms_locales()`
- `ms_locale_df()`
**ms_locales_df**

*Detailed Names of Microsoft Locales and Voices*

**Description**

Detailed Names of Microsoft Locales and Voices

**Usage**

`ms_locales_df`

**Format**

A `data.frame` with 2 columns:

- **locale** full name with Microsoft Server Speech Text to Speech Voice in there
- **display_name** short display name
- **local_name** local name
- **short_name** shorter than locale name, but has language code as well
- **gender** gender of the voice
- **code** the 5 character language code, separated by a hyphen, also referred to as a locale
- **sample_rate** sample rate (in Hz) of the voice
- **voice_type** Standard or Neural

---

**ms_locale_names**

*Names of Microsoft Locales*

**Description**

Names of Microsoft Locales

**Usage**

`ms_locale_names`

**Format**

A `data.frame` with 2 columns:

- **code** the 5 character language code, separated by a hyphen, also referred to as a locale
- **language** the name of the language
### ms_read_synthesis

**Read Synthesized output**

**Description**

Read Synthesized output

**Usage**

```r
ms_read_synthesis(output)
```

**Arguments**

- `output`: List from `ms_synthesize` with elements `output_format` and `content`

**Value**

A Wave Object

**Note**

The `tuneR` package cannot read all different types of the output here.

### ms_synthesize

**Get Microsoft Text To Speech (TTS) or Cognitive Services Token from API Key**

**Description**

Get Microsoft Text To Speech (TTS) or Cognitive Services Token from API Key

**Usage**

```r
ms_synthesize(
  script,
  token = NULL,
  api_key = NULL,
  gender = c("Female", "Male"),
  language = "en-US",
  voice = NULL,
  output_format = c("raw-16khz-16bit-mono-pcm", "raw-8khz-8bit-mono-mulaw",
                   "riff-8khz-8bit-mono-alaw", "riff-8khz-8bit-mono-mulaw",
                   "riff-16khz-16bit-mono-pcm", "audio-16khz-128kbitrate-mono-mp3",
                   "audio-16khz-64kbitrate-mono-mp3", "audio-16khz-32kbitrate-mono-mp3",
                   "raw-24khz-16bit-mono-pcm", "riff-24khz-16bit-mono-pcm",
                   "audio-24khz-160kbitrate-mono-mp3", "audio-24khz-96kbitrate-mono-mp3",
                   "audio-24khz-48kbitrate-mono-mp3"),
  ...)
**ms_synthesize**

```r
escape = FALSE,
region = NULL,
api = c("tts", "bing"),
...
)

ms_region(region = ms_regions())

ms_regions()

ms_set_region(region = ms_regions())

ms_synthesize_api_url(api = c("tts", "bing"), region = NULL)
```

### Arguments

- **script**
  - A character vector of lines to be spoken

- **token**
  - An authentication token, base-64 encoded usually from `ms_get_tts_token`. If not provided, will be created from `ms_get_tts_token`

- **api_key**
  - Microsoft Cognitive Services API key, if token is not provided.

- **gender**
  - Sex of the Speaker

- **language**
  - Language to be spoken, must be from `ms_language_codes`

- **voice**
  - full voice name, usually from `ms_language_to_ms_name`. Will override language and gender.

- **output_format**
  - Format of the output, see https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/how-to-migrate-from-bing-speech for more information

- **escape**
  - Should non-standard characters be substituted? Should not be used if script has SSML tags. See `ms_create_ssml`

- **region**
  - Subscription region for your key. See https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/overview#reference-docs

- **api**
  - which API to authorize on, either tts for text to speech or bing for Bing text to speech API

- **...**
  - Additional arguments to send to POST

### Value

A list of the request, content, token, and ‘SSML’.

### Note

The content is likely in a binary format and the output depends on the ‘output_format’ chosen. For example, if the ‘output_format’ is an ‘MP3’, then see below example
Examples

```r
## Not run:
if (ms_have_tts_key()) {
  res = ms_synthesize(
    script = "hey, how are you doing? I'm doing pretty good",
    output_format = "audio-16khz-128kbitrate-mono-mp3")
  tmp <- tempfile(fileext = ".mp3")
  writeBin(res$content, con = tmp)
  mp3 = tuneR::readMP3(tmp)
}

## End(Not run)
ms_regions()
```

---

**print.token**

**Print method for token**

### Description

Print method for token

### Usage

```r
## S3 method for class 'token'
print(x, reveal = FALSE, ...)
reveal(x, ...)
```

### Arguments

- `x` an object used to select a method.
- `reveal` Should the token be revealed
- `...` further arguments passed to or from other methods

### Examples

```r
x = "asdf"
class(x) = "token"
print(x)
attr(x, "timestamp") = Sys.time()
print(x)
print(x, reveal = TRUE)
```
Index

* datasets
  ms_locale_names, 7
  ms_locales_df, 7

GET, 5
ms_auth_url (ms_get_tts_token), 4
ms_create_ssml, 2, 9
ms_expired_token (ms_get_tts_token), 4
ms_get_tts_key, 3
ms_get_tts_token, 2, 4, 4, 5, 9
ms_have_tts_key (ms_get_tts_key), 3
ms_language_codes, 2, 5, 9
ms_language_codes (ms_locales), 6
ms_language_to_ms_name, 2, 5, 9
ms_languages (ms_locales), 6
ms_list_voices, 5, 6
ms_locale_df (ms_locales), 6
ms_locale_names, 7
ms_locales, 6
ms_locales_df, 7
ms_read_synthesis, 8
ms_region (ms_synthesize), 8
ms_regions (ms_synthesize), 8
ms_set_region (ms_synthesize), 8
ms_set_tts_key (ms_get_tts_key), 3
ms_synthesize, 8, 8
ms_synthesize_api_url (ms_synthesize), 8
ms_valid_tts_key (ms_get_tts_key), 3
ms_validate_language_gender (ms_language_to_ms_name), 5
ms_voice_info (ms_create_ssml), 2

POST, 9
print.token, 10
reveal (print.token), 10