

Package ‘text2speech’

October 14, 2022

Type Package

Title Text to Speech

Version 0.2.13

Description Unifies different text to speech engines, such as Google, Microsoft, and Amazon. Text synthesis can be done in any engine with a simple switch of an argument denoting the service requested. The 'aws.polly' package has been orphaned and can be found from the CRAN archives.

License GPL-3

Suggests aws.polly, covr, rmarkdown, testthat, stringi

Encoding UTF-8

LazyData true

VignetteBuilder knitr

RoxygenNote 7.1.0

URL <https://github.com/muschellij2/text2speech>

BugReports <https://github.com/muschellij2/text2speech/issues>

Imports aws.signature, dplyr, googleAuthR, googleLanguageR, httr, mscstts (>= 0.5.1), tuneR, magrittr, knitr, utils

NeedsCompilation no

Author John Muschelli [aut, cre] (<<https://orcid.org/0000-0001-6469-1750>>)

Maintainer John Muschelli <muschellij2@gmail.com>

Repository CRAN

Date/Publication 2020-06-30 15:40:02 UTC

R topics documented:

pcm_to_wav	2
tts_amazon_auth	3
tts_bind_wav	4

tts_google	4
tts_speak_engine	6
tts_voices	7

Index	9
--------------	----------

pcm_to_wav	<i>Convert PCM to Wav</i>
------------	---------------------------

Description

Convert PCM to Wav

Usage

```
pcm_to_wav(
  input,
  output = tempfile(fileext = ".wav"),
  sample_rate = 16000,
  extensible = FALSE
)
```

Arguments

input	output from 'get_synthesis' from aws.polly or PCM filename
output	output file for Wav file
sample_rate	Sampling rate for tuneR::Wave
extensible	passed to tuneR::writeWave

Value

A filename of the output

Examples

```
fname = system.file("extdata", "pcm_file.wav", package = "text2speech")
res = pcm_to_wav(fname)
testthat::expect_error(tuneR::readWave(fname))
testthat::expect_is(tuneR::readWave(res), "Wave")

## Not run:
if (requireNamespace("aws.polly", quietly = TRUE)) {
  text = "hey, ho, let's go!"
  if (tts_amazon_auth()) {
    res = tts_amazon(text, output_format = "wav")
  }
}

## End(Not run)
```

tts_amazon_auth	<i>Authorize Text-to-Speech Engine</i>
-----------------	--

Description

Authorize Text-to-Speech Engine

Usage

```
tts_amazon_auth(key_or_json_file = NULL, ...)  
  
tts_auth(  
  service = c("amazon", "google", "microsoft"),  
  key_or_json_file = NULL,  
  ...  
)  
  
tts_google_authenticated()  
  
tts_amazon_authenticated()  
  
tts_microsoft_authenticated(...)  
  
tts_google_auth(key_or_json_file = NULL, ...)  
  
tts_microsoft_auth(key_or_json_file = NULL, ...)
```

Arguments

key_or_json_file	Either an API key (for Microsoft) or JSON file (for Google)
...	Additional arguments to pass to use_credentials or ms_get_tts_token
service	type of synthesis engine

Value

A logical indicator of authorization

Examples

```
tts_auth("google")  
tts_auth("microsoft")  
  
tts_google_authenticated()  
tts_microsoft_authenticated()  
  
tts_google_auth()
```

```
tts_microsoft_auth()
if (requireNamespace("aws.polly", quietly = TRUE)) {
  tts_auth("amazon")
  tts_amazon_authenticated()
  tts_amazon_auth()
}
```

tts_bind_wav

Bind Wavs together

Description

Bind Wavs together

Usage

```
tts_bind_wav(result)
```

Arguments

result A data.frame from `text2speech::tts()`.

Value

A data.frame with the same structure as that of `tts`

Note

As the data are split due to limits of the API, then this allows the text and the results to be harmonized

tts_google

Text to Speech

Description

Text to Speech

Usage

```

tts_google(
  text,
  output_format = c("mp3", "wav"),
  voice = "en-US-Standard-C",
  bind_audio = TRUE,
  ...
)

tts_amazon(
  text,
  output_format = c("mp3", "wav"),
  voice = "Joanna",
  bind_audio = TRUE,
  ...
)

tts_microsoft(
  text,
  output_format = c("mp3", "wav"),
  voice = "Microsoft Server Speech Text to Speech Voice (en-US, ZiraRUS)",
  bind_audio = TRUE,
  ...
)

tts_default_voice(service = c("amazon", "google", "microsoft"))

tts(
  text,
  output_format = c("mp3", "wav"),
  ...,
  service = c("amazon", "google", "microsoft"),
  bind_audio = TRUE
)

```

Arguments

text	A character vector of text to speak
output_format	Format of output files
voice	A full voice name that can be passed to the service, such as the argument voice for <code>get_synthesis` from \code{aws.polly}</code> , or or <code>[mscsstts::ms_synthesize()]</code> or the name argument for <code>googleLanguageR::gl_talk()</code>
bind_audio	Should the <code>text2speech::tts_bind_wav()</code> be run on after the audio has been created, to ensure that the length of text and the number of rows is consistent? This affects the output format of some audio.
...	Additional arguments to <code>text2speech::tts_google()</code> , <code>text2speech::tts_amazon()</code> , or <code>text2speech::tts_microsoft()</code>

service service to use

Value

A data.frame of text and wav files

Note

All functions have a voice`` argument fro a full voice name that can be passed to the service, such as voicefo from aws.polly

Examples

```
tts_default_voice("amazon")
tts_default_voice("google")
tts_default_voice("microsoft")
## Not run:
text='<speack>
  He was caught up in the game.<break time="1s"/> In the middle of the
  10/3/2014 <sub alias="World Wide Web Consortium">W3C</sub> meeting,
  he shouted, "Nice job!" quite loudly. When his boss stared at him, he repeated
  <amazon:effect name="whispered">"Nice job,"</amazon:effect> in a
  whisper.
</speack>'

## End(Not run)

if (requireNamespace("stringi", quietly = TRUE)) {
  set.seed(1)
  text = stringi::stri_rand_lipsum(10)
  text[3] = paste0(text[3:length(text)], collapse = " " )
  text = text[c(1,3)]
  nchar(text)
  if (tts_auth("google")) {
    res = tts(text, service = "google", bind_audio = FALSE)
    testthat::expect_equal(nrow(res), length(text) + 1)
    bound = tts_bind_wav(res)
    testthat::expect_equal(nrow(bound), length(text))
  }
}
```

tts_speak_engine *Speak Enginge for knitr*

Description

Speak Enginge for knitr

Usage

```
tts_speak_engine(options)
```

Arguments

options A list of chunk options. Usually this is just the object options passed to the engine function; see [knit_engines](#)

Value

A character string generated from the source code and output using the appropriate output hooks.

Examples

```
knitr::knit_engines$set(speak = tts_speak_engine)
options = list(
  code = "hey let's go to the park",
  eval = FALSE,
  label = "random",
  fig.path = tempdir(),
  echo = TRUE, results = "asis",
  engine = "speak")
tts_speak_engine(options)
  if (tts_auth("google")) {
    options$eval = TRUE
    tts_speak_engine(options)
  }
```

tts_voices

Text to Speech Voices

Description

Text to Speech Voices

Usage

```
tts_voices(service = c("amazon", "google", "microsoft"), ...)
```

```
tts_amazon_voices(...)
```

```
tts_microsoft_voices(...)
```

```
tts_google_voices(...)
```

Arguments

service service to use
... Additional arguments to service voice listings.

Value

A data.frame of language codes, voices, genders, and language names

Examples

```
if (tts_microsoft_auth()) {
  tts_voices(service = "microsoft")
}
if (tts_google_auth()) {
  tts_voices(service = "google")
}
if (requireNamespace("aws.polly", quietly = TRUE)) {
  if (tts_amazon_auth()) {
    tts_voices(service = "amazon")
  }
}
```


Index

googleLanguageR::gl_talk(), 5

knit_engines, 7

ms_get_tts_token, 3

pcm_to_wav, 2

text2speech::tts(), 4

text2speech::tts_bind_wav(), 5

tts (tts_google), 4

tts_amazon (tts_google), 4

tts_amazon_auth, 3

tts_amazon_authenticated
 (tts_amazon_auth), 3

tts_amazon_voices (tts_voices), 7

tts_auth (tts_amazon_auth), 3

tts_bind_wav, 4

tts_default_voice (tts_google), 4

tts_google, 4

tts_google_auth (tts_amazon_auth), 3

tts_google_authenticated
 (tts_amazon_auth), 3

tts_google_voices (tts_voices), 7

tts_microsoft (tts_google), 4

tts_microsoft_auth (tts_amazon_auth), 3

tts_microsoft_authenticated
 (tts_amazon_auth), 3

tts_microsoft_voices (tts_voices), 7

tts_speak_engine, 6

tts_voices, 7

tuneR::Wave, 2

tuneR::writeWave, 2

use_credentials, 3