

# Package ‘text2speech’

June 24, 2019

**Type** Package

**Title** Text to Speech

**Version** 0.2.5

**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.

**License** GPL-3

**Suggests** covr, knitr, rmarkdown, testthat, stringi

**Encoding** UTF-8

**LazyData** true

**VignetteBuilder** knitr

**RoxygenNote** 6.1.1

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

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

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

**NeedsCompilation** no

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

**Maintainer** John Muschelli <[muschellij2@gmail.com](mailto:muschellij2@gmail.com)>

**Repository** CRAN

**Date/Publication** 2019-06-24 20:50:27 UTC

## R topics documented:

tts_amazon_auth	2
tts_bind_wav	3
tts_google	3
tts_voices	4

<b>Index</b>	<b>6</b>
--------------	----------

---

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 [aws.signature::use_credentials()] or [mscsstts::ms_get_tts_token()]
service	type of synthesis engine

**Value**

A logical indicator of authorization

**Examples**

```
tts_auth("google")
tts_auth("amazon")
tts_auth("microsoft")

tts_google_authenticated()
tts_microsoft_authenticated()
tts_amazon_authenticated()

tts_google_auth()
tts_microsoft_auth()
tts_amazon_auth()
```

---

tts_bind_wav	<i>Bind Ways together</i>
--------------	---------------------------

---

**Description**

Bind Ways 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	<i>Text to Speech</i>
------------	-----------------------

---

**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(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 [aws.polly::get_synthesis], or [mscstts::ms_synthesize()] or the ‘name’ argument for [googleLanguageR::gl_talk()]
bind_audio	Should the [text2speech::tts_bind_wav()] 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 [text2speech::tts_google()], [text2speech::tts_amazon()], or [text2speech::tts_microsoft()]
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 ‘voice’ for [aws.polly::get\_synthesis]

**Examples**

```
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\_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 (tts_amazon_auth()) {  
  tts_voices(service = "amazon")  
}
```

# Index

tts (tts\_google), 3  
tts\_amazon (tts\_google), 3  
tts\_amazon\_auth, 2  
tts\_amazon\_authenticated  
    (tts\_amazon\_auth), 2  
tts\_amazon\_voices (tts\_voices), 4  
tts\_auth (tts\_amazon\_auth), 2  
tts\_bind\_wav, 3  
tts\_google, 3  
tts\_google\_auth (tts\_amazon\_auth), 2  
tts\_google\_authenticated  
    (tts\_amazon\_auth), 2  
tts\_google\_voices (tts\_voices), 4  
tts\_microsoft (tts\_google), 3  
tts\_microsoft\_auth (tts\_amazon\_auth), 2  
tts\_microsoft\_authenticated  
    (tts\_amazon\_auth), 2  
tts\_microsoft\_voices (tts\_voices), 4  
tts\_voices, 4