

Package ‘tts.api’

July 24, 2026

Title 'OpenAI'-Compatible Text-to-Speech API Client

Version 0.3.0

Description A minimal-dependency R client for 'OpenAI'-compatible text-to-speech APIs (see <https://developers.openai.com/api/reference/resources/audio>). Supports 'OpenAI', 'ElevenLabs', local servers ('Chatterbox', 'LM Studio', 'OpenWebUI', 'AnythingLLM'), and the 'chatterbox' package for in-process synthesis. Provides a unified interface for speech synthesis, voice cloning, and voice design with backend-specific parameters.

License MIT + file LICENSE

Encoding UTF-8

URL <https://github.com/cornball-ai/tts.api>

BugReports <https://github.com/cornball-ai/tts.api/issues>

Imports curl, jsonlite, tools, utils

Suggests chatterbox, tinytest, torch

NeedsCompilation no

Author Troy Hernandez [aut, cre] (ORCID: <https://orcid.org/0009-0005-4248-604X>), cornball.ai [cph]

Maintainer Troy Hernandez <troy@cornball.ai>

Repository CRAN

Date/Publication 2026-07-24 09:40:02 UTC

Contents

chatterbox_available	2
chatterbox_voice_upload	3
clear_native_chatterbox_cache	3
elevenlabs_voices	4
elevenlabs_voice_delete	5
elevenlabs_voice_upload	5

languages	6
qwen3_available	7
set_elevenlabs_key	7
set_tts_base	8
set_tts_key	8
speech_clone	9
speech_design	10
tts	11
tts_health	13
tts_providers	14
tts_voices	15
voices	15
voice_ensure	16
voice_file	17
voice_library	17
voice_upload	18
Index	20

chatterbox_available	<i>Check if Chatterbox Service is Available</i>
----------------------	---

Description

Quick check if the local Chatterbox TTS API is reachable. Returns TRUE/FALSE without throwing errors.

Usage

```
chatterbox_available(port = NULL, timeout = 2)
```

Arguments

port	Port to check (default from CHATTERBOX_PORT env var or 7810)
timeout	Timeout in seconds (default 2)

Value

TRUE if service is available, FALSE otherwise

Examples

```
## Not run:
  if (chatterbox_available()) {
    tts("Hello", voice = "default", backend = "chatterbox")
  }

## End(Not run)
```

`chatterbox_voice_upload`*Upload Voice to Chatterbox*

Description

Uploads a voice sample directly to a local Chatterbox instance. Does not require setting the API base URL first.

Usage

```
chatterbox_voice_upload(voice_file, voice_name, port = NULL, timeout = 30)
```

Arguments

<code>voice_file</code>	Character. Path to the voice sample file (mp3, wav).
<code>voice_name</code>	Character. Name to save the voice as.
<code>port</code>	Port for Chatterbox API (default from CHATTERBOX_PORT env var or 7810)
<code>timeout</code>	Timeout in seconds (default 30)

Value

TRUE if upload succeeded, FALSE otherwise. Does not throw errors.

Examples

```
## Not run:
# Upload a voice sample
success <- chatterbox_voice_upload("my_voice.wav", "my-voice")

if (success) {
  tts("Hello!", voice = "my-voice", backend = "chatterbox")
}

## End(Not run)
```

`clear_native_chatterbox_cache`*Clear native chatterbox model cache*

Description

Removes cached native chatterbox models from memory. Call this to free GPU/RAM after batch processing is complete.

Usage

```
clear_native_chatterbox_cache()
```

Value

Invisibly returns NULL, called for its side effect.

Examples

```
## Not run:  
clear_native_chatterbox_cache()  
  
## End(Not run)
```

elevenlabs_voices	<i>List ElevenLabs Voices</i>
-------------------	-------------------------------

Description

Get all voices available in your ElevenLabs account.

Usage

```
elevenlabs_voices()
```

Value

Data frame with voice_id, name, and category columns.

Examples

```
## Not run:  
voices <- elevenlabs_voices()  
print(voices)  
  
## End(Not run)
```

`elevenlabs_voice_delete`*Delete ElevenLabs Voice*

Description

Delete a voice from your ElevenLabs account.

Usage

```
elevenlabs_voice_delete(voice_id)
```

Arguments

`voice_id` Character. The voice ID to delete.

Value

Invisible TRUE on success.

Examples

```
## Not run:  
elevenlabs_voice_delete("abc123")  
  
## End(Not run)
```

`elevenlabs_voice_upload`*Upload Voice to ElevenLabs (Instant Voice Clone)*

Description

Create an instant voice clone on ElevenLabs from audio samples.

Usage

```
elevenlabs_voice_upload(files, name, description = NULL,  
                        remove_background_noise = FALSE, labels = NULL)
```

Arguments

`files` Character vector. Paths to audio files (1-25 files, each up to 10MB).
`name` Character. Name for the cloned voice.
`description` Character or NULL. Optional description.
`remove_background_noise` Logical. Remove background noise from samples. Default FALSE.
`labels` Named list or NULL. Optional labels (e.g., `list(accent = "british")`).

Value

List with voice_id and name on success.

Examples

```
## Not run:
# Clone from a single file
voice <- elevenlabs_voice_upload(
  files = "my_voice.mp3",
  name = "My Clone"
)

# Use the cloned voice
tts("Hello!", voice = voice$voice_id, backend = "elevenlabs")

# Clone from multiple samples
voice <- elevenlabs_voice_upload(
  files = c("sample1.mp3", "sample2.mp3"),
  name = "Better Clone",
  remove_background_noise = TRUE
)

## End(Not run)
```

languages

List Supported Languages

Description

Returns a list of languages supported by the TTS server.

Usage

```
languages()
```

Value

A character vector or list of supported language codes.

Examples

```
## Not run:
set_tts_base("http://localhost:7810")
langs <- languages()
print(langs)

## End(Not run)
```

qwen3_available	<i>Check if Qwen3-TTS Service is Available</i>
-----------------	--

Description

Quick check if Qwen3-TTS API is reachable. Distinguishes from Chatterbox by checking for the qwen3-specific /v1/audio/speech/design endpoint.

Usage

```
qwen3_available(port = NULL, timeout = 2)
```

Arguments

port	Port to check (default from QWEN3_TTS_PORT env var or 7811)
timeout	Timeout in seconds (default 2)

Value

TRUE if Qwen3-TTS is available, FALSE otherwise

Examples

```
## Not run:  
if (qwen3_available()) {  
  tts("Hello", voice = "Vivian", backend = "qwen3")  
}  
  
## End(Not run)
```

set_elevenlabs_key	<i>Set ElevenLabs API Key</i>
--------------------	-------------------------------

Description

Configure the API key for ElevenLabs TTS backend.

Usage

```
set_elevenlabs_key(api_key)
```

Arguments

api_key	Character. Your ElevenLabs API key. Alternatively, set the ELEVENLABS_API_KEY environment variable.
---------	---

Value

Invisibly returns the API key.

Examples

```
set_elevenlabs_key("example-api-key")
getOption("tts.elevenlabs_key")
```

set_tts_base	<i>Set the TTS API Base URL</i>
--------------	---------------------------------

Description

Sets the base URL for the TTS API. This should be the root URL without trailing slash (e.g., "http://localhost:7810" or "https://api.openai.com").

Usage

```
set_tts_base(url)
```

Arguments

url	Character string. The base URL for the API.
-----	---

Value

Invisibly returns the previous value.

Examples

```
set_tts_base("http://localhost:7810")
getOption("tts.api_base")
```

set_tts_key	<i>Set the TTS API Key</i>
-------------	----------------------------

Description

Sets the API key for authentication. Required for cloud services like OpenAI, typically ignored by local servers.

Usage

```
set_tts_key(key)
```

Arguments

key Character string. The API key.

Value

Invisibly returns the previous value.

Examples

```
set_tts_key("sk-example-key")
getOption("tts.api_key")
```

speech_clone	<i>Generate Speech with Voice Cloning</i>
--------------	---

Description

Uploads a voice sample and generates speech in that voice using the /v1/audio/speech/upload endpoint.

Usage

```
speech_clone(input, voice_file, file = NULL,
              backend = c("auto", "chatterbox", "qwen3"), ref_text = NULL,
              x_vector_only = FALSE, language = NULL, exaggeration = NULL,
              temperature = NULL, cfg_weight = NULL, speed = NULL, seed = NULL)
```

Arguments

input	Character. The text to convert to speech.
voice_file	Character. Path to the voice sample file (mp3, wav, etc.).
file	Character or NULL. Output file path. If NULL, returns raw bytes.
backend	Character. Backend to use: "auto" to detect qwen3 vs chatterbox, "qwen3" for Qwen3-TTS, or "chatterbox" for Chatterbox.
ref_text	Character or NULL. Transcript of the reference audio. Required by qwen3-tts for high-quality cloning (ICL mode). If NULL and backend requires it, use x_vector_only = TRUE for faster but lower-quality cloning.
x_vector_only	Logical. If TRUE, use only speaker embedding for cloning (faster but lower quality). Useful when ref_text is not available.
language	Character or NULL. Language for synthesis (qwen3-tts specific).
exaggeration	Numeric or NULL. Exaggeration parameter (Chatterbox-specific).
temperature	Numeric or NULL. Sampling temperature for generation.
cfg_weight	Numeric or NULL. CFG weight parameter (Chatterbox-specific).
speed	Numeric or NULL. Speed multiplier for the audio.
seed	Integer or NULL. Random seed for reproducible output.

Value

If file is provided, invisibly returns the file path. If file is NULL, returns raw audio bytes.

Examples

```
## Not run:
set_tts_base("http://localhost:7811")

# Clone voice with transcript (high quality, qwen3-tts)
speech_clone(
  input = "Hello with my custom voice!",
  voice_file = "my_voice.wav",
  ref_text = "This is what I said in the recording.",
  file = "output.wav",
  backend = "qwen3"
)

# Clone voice without transcript (faster, lower quality)
speech_clone(
  input = "Hello with my custom voice!",
  voice_file = "my_voice.wav",
  x_vector_only = TRUE,
  file = "output.wav"
)

# Chatterbox-style cloning
speech_clone(
  input = "Hello with my custom voice!",
  voice_file = "my_voice.mp3",
  file = "output.wav",
  exaggeration = 0.8,
  backend = "chatterbox"
)

## End(Not run)
```

speech_design

Generate Speech with Designed Voice

Description

Create a voice from a natural language description and generate speech. This endpoint requires the qwen3-tts-api backend with the voice design model.

Usage

```
speech_design(input, voice_description, file = NULL, language = "English")
```

Arguments

input	Character. The text to convert to speech.
voice_description	Character. Natural language description of the desired voice (e.g., "A warm, friendly female voice with a slight British accent").
file	Character or NULL. Output file path. If NULL, returns raw bytes.
language	Character. Language for synthesis. Supported: Chinese, English, Japanese, Korean, German, French, Russian, Portuguese, Spanish, Italian.

Details

This function uses the `/v1/audio/speech/design` endpoint which is specific to `qwen3-tts-api`. It requires the `VoiceDesign` model to be loaded (loaded on first use).

Value

If `file` is provided, invisibly returns the file path. If `file` is `NULL`, returns raw audio bytes.

Examples

```
## Not run:
set_tts_base("http://localhost:7811")

# Generate speech with a custom designed voice
speech_design(
  input = "Hello, I am your AI assistant.",
  voice_description = "A professional male voice, clear and authoritative",
  file = "assistant.wav"
)

# Playful voice
speech_design(
  input = "Let's have some fun!",
  voice_description = "An energetic young female voice, cheerful and playful",
  file = "playful.wav"
)

## End(Not run)
```

tts

Generate Speech from Text

Description

Generate audio from text using various TTS backends: native R `chatterbox`, local `Chatterbox` container, `OpenAI TTS API`, or `ElevenLabs API`.

Usage

```
tts(input, voice, file = NULL,
    backend = c("auto", "chatterbox", "qwen3", "openai", "elevenlabs"),
    source = c("api", "auto", "package"), model = NULL, temperature = NULL,
    speed = NULL, exaggeration = NULL, cfg_weight = NULL, stability = NULL,
    similarity_boost = NULL, seed = NULL, response_format = NULL,
    instructions = NULL, language = NULL, device = "cuda")
```

Arguments

input	Character. The text to convert to speech.
voice	Character. The voice to use for synthesis. For Chatterbox (package or container) and Qwen3: a voice-library name (e.g. "FatherChristmas") or a path to a reference audio file. The package source resolves names via voice_file ; containers resolve them server-side. For OpenAI: "alloy", "echo", "fable", "onyx", "nova", "shimmer". For ElevenLabs: voice ID (e.g., "XpDLYThV0yUAFjVTok7m").
file	Character or NULL. Output file path. If NULL, returns raw bytes.
backend	Character. The TTS engine: "chatterbox", "qwen3", "openai", "elevenlabs", or "auto" (defaults to "chatterbox"). This selects <i>what</i> synthesizes; see source for <i>where</i> it runs.
source	Character. Where the engine runs: "api" (default) for an HTTP service (container or hosted package server), "package" for the in-process R chatterbox package, or "auto" which uses the package for chatterbox when it is installed and the API otherwise. Only "chatterbox" has a package source; the other engines are API-only.
model	Character or NULL. The sub-model to use. For Chatterbox package source: "turbo" loads Chatterbox Turbo. For OpenAI: "tts-1" or "tts-1-hd". For ElevenLabs: "eleven_multilingual_v2" (default), "eleven_turbo_v2_5", etc.
temperature	Numeric or NULL. Sampling temperature for generation.
speed	Numeric or NULL. Speed multiplier for the audio.
exaggeration	Numeric or NULL. Exaggeration parameter (Chatterbox-specific).
cfg_weight	Numeric or NULL. CFG weight parameter (Chatterbox-specific).
stability	Numeric or NULL. Voice stability 0-1 (ElevenLabs-specific). Default 0.5.
similarity_boost	Numeric or NULL. Similarity boost 0-1 (ElevenLabs-specific). Default 0.75.
seed	Integer or NULL. Random seed for reproducible output.
response_format	Character or NULL. Audio format (e.g., "wav", "mp3"). If NULL and file is provided, inferred from file extension.
instructions	Character or NULL. Instructions for how the voice should speak (OpenAI/Qwen3, e.g., "Speak in a cheerful and positive tone.").
language	Character or NULL. Language for synthesis (Qwen3-specific). Options: "English", "Chinese", "Japanese", "Korean", "French", "German", "Spanish", "Italian", "Portuguese", "Russian".
device	Character. Device for native backend: "cuda", "cpu", or "mps". Default "cuda".

Value

If file is provided, invisibly returns the file path. If file is NULL, returns raw audio bytes (or list for native backend).

Examples

```
## Not run:
# Native R chatterbox package, no container (source resolves to "package")
tts("Hello, world!", voice = "FatherChristmas", file = "hello.mp3")

# Chatterbox Turbo via the package
tts("Hello, world!", voice = "FatherChristmas", file = "hello.mp3",
    source = "package", model = "turbo")

# Chatterbox over an HTTP container
set_tts_base("http://troy-g5:7810")
tts("Hello, world!", voice = "FatherChristmas", file = "hello.wav",
    source = "api")

# Using OpenAI TTS
tts("Hello, world!", voice = "nova", file = "hello.mp3", backend = "openai")

# Using ElevenLabs
tts("Hello, world!", voice = "XpDLYThV0yUAFjVTok7m",
    file = "hello.mp3", backend = "elevenlabs")

## End(Not run)
```

tts_health

Check TTS API Health

Description

Checks whether the TTS backend is reachable by trying common health endpoints.

Usage

```
tts_health()
```

Value

A list with components:

ok Logical. TRUE if the server is reachable.

status Character. A human-readable status message.

raw The raw response from the server (if any).

Examples

```
## Not run:
set_tts_base("http://localhost:4123")
h <- tts_health()
if (h$ok) {
  message("Server is ready!")
}

## End(Not run)
```

tts_providers	<i>TTS Provider Configurations</i>
---------------	------------------------------------

Description

A list of known TTS providers with their default voices and configuration.

Usage

```
tts_providers
```

Format

An object of class list of length 4.

Details

A named list where each element contains:

- voices** Character vector of default/known voice names, or NULL if dynamic
- env_var** Name of environment variable for API key, or NULL if not needed
- base_url** Default API base URL, or NULL

Examples

```
names(tts_providers)
tts_providers[["OpenAI"]$voices
```

tts_voices	<i>Get Voices for a TTS Provider</i>
------------	--------------------------------------

Description

Returns available voices for a given provider. Attempts to fetch dynamically from the API first, falls back to static list if unavailable.

Usage

```
tts_voices(provider, base_url = NULL, timeout = 2)
```

Arguments

provider	Character string naming the provider (e.g., "OpenAI", "Chatterbox (Local)")
base_url	Optional base URL override for the provider's API
timeout	Timeout in seconds for API requests (default 2)

Value

Character vector of voice names

Examples

```
## Not run:  
tts_voices("OpenAI")  
tts_voices("Chatterbox (Local)")  
  
## End(Not run)
```

voices	<i>List Available Voices</i>
--------	------------------------------

Description

Returns a list of available voices from the TTS server. Tries the /v1/audio/voices endpoint first, then falls back to /voices.

Usage

```
voices()
```

Value

A character vector of voice names, or a list/data.frame depending on the server's response format.

Examples

```
## Not run:
set_tts_base("http://localhost:7810")
v <- voices()
print(v)

## End(Not run)
```

voice_ensure

Ensure Voice is Available on TTS Server

Description

Checks if a voice exists on the current TTS server. If not, finds the voice file in the library and uploads it.

Usage

```
voice_ensure(voice_name, voices_dir = NULL)
```

Arguments

voice_name	Character. Name of the voice.
voices_dir	Path to voice library directory (see voice_library).

Value

Invisible TRUE if the voice is available (already present or successfully uploaded).

Examples

```
## Not run:
set_tts_base("http://localhost:4123")
voice_ensure("BigCasey")
tts("Hello!", voice = "BigCasey", file = "out.mp3")

## End(Not run)
```

voice_file	<i>Resolve Voice Name to File Path</i>
------------	--

Description

Looks up a voice by name in the voice library directory. Case-insensitive matching is attempted if exact match fails.

Usage

```
voice_file(voice_name, voices_dir = NULL)
```

Arguments

voice_name	Character. Name of the voice to find.
voices_dir	Path to voice library directory (see voice_library).

Value

Character string: full path to the voice file. Stops with an error if the voice is not found.

Examples

```
## Not run:  
voice_file("BigCasey")  
voice_file("delicatecasey")  
  
## End(Not run)
```

voice_library	<i>List Voice Files in Library</i>
---------------	------------------------------------

Description

Scans a directory for voice sample files and returns their paths.

Usage

```
voice_library(voices_dir = NULL)
```

Arguments

voices_dir	Path to voice library directory. Defaults to TTS_VOICES_DIR env var, then ~/ .cornball/voices.
------------	--

Value

Named character vector: names are voice names (filename without extension), values are full file paths. Returns empty named character vector if directory doesn't exist or contains no voice files.

Examples

```
## Not run:
voice_library()
voice_library("~/my-voices")

## End(Not run)
```

voice_upload	<i>Upload Voice to Library</i>
--------------	--------------------------------

Description

Uploads a voice sample to the server's voice library for reuse. Once uploaded, the voice can be used by name in tts().

Usage

```
voice_upload(voice_file, voice_name, language = NULL)
```

Arguments

voice_file	Character. Path to the voice sample file (mp3, wav, etc.).
voice_name	Character. Name to save the voice as.
language	Character or NULL. Language code (e.g., "en", "fr", "es").

Value

Invisibly returns the response from the server.

Examples

```
## Not run:
set_tts_base("http://localhost:7810")

# Upload a voice
voice_upload(
  voice_file = "my_voice.wav",
  voice_name = "my-custom-voice"
)

# Upload with language
voice_upload(
  voice_file = "french_voice.wav",
```

```
    voice_name = "french-speaker",
    language = "fr"
)

# Then use it by name
tts(
  input = "Hello with my custom voice!",
  voice = "my-custom-voice",
  file = "output.wav"
)

## End(Not run)
```

Index

* datasets

tts_providers, [14](#)

chatterbox_available, [2](#)

chatterbox_voice_upload, [3](#)

clear_native_chatterbox_cache, [3](#)

elevenlabs_voice_delete, [5](#)

elevenlabs_voice_upload, [5](#)

elevenlabs_voices, [4](#)

languages, [6](#)

qwen3_available, [7](#)

set_elevenlabs_key, [7](#)

set_tts_base, [8](#)

set_tts_key, [8](#)

speech_clone, [9](#)

speech_design, [10](#)

tts, [11](#)

tts_health, [13](#)

tts_providers, [14](#)

tts_voices, [15](#)

voice_ensure, [16](#)

voice_file, [12](#), [17](#)

voice_library, [16](#), [17](#), [17](#)

voice_upload, [18](#)

voices, [15](#)