Package ‘openai’

July 22, 2025

Type Package

Title R Wrapper for OpenAI API

Version 0.4.1

Date 2023-03-14

Description An R wrapper of OpenAI API endpoints (see <https://platform.openai.com/docs/introduction> for details). This package covers Models, Completions, Chat, Edits, Images, Embeddings, Audio, Files, Fine-tunes, Moderations, and legacy Engines endpoints.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.1

URL https://github.com/irudnyts/openai,
https://irudnyts.github.io/openai/

BugReports https://github.com/irudnyts/openai/issues

Depends R (>= 3.5)

Imports assertthat (>= 0.2.1), glue (>= 1.6.2), httr (>= 1.4.3), jsonlite (>= 1.8.0), lifecycle (>= 1.0.1), magrittr (>= 2.0.3)

Suggests testthat (>= 3.0.0), purrr (>= 0.3.4), covr (>= 3.5.1)

Config/testthat/edition 3

NeedsCompilation no

Author Iegor Rudnytskyi [aut, cre]

Maintainer Iegor Rudnytskyi <iegor.rudnytskyi@gmail.com>

Repository CRAN

Date/Publication 2023-03-15 00:20:02 UTC

Contents

`cancel_fine_tune` *Cancel fine-tune*

Description

Cancel a running fine-tune job. See this page for details.

Usage

```
cancel_fine_tune(
  fine_tune_id,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

fine_tune_id — required; a length one character vector.

openai_api_key — required; defaults to Sys.getenv("OPENAI_API_KEY") (i.e., the value is retrieved from the .Renviron file); a length one character vector. Specifies OpenAI API key.

openai_organization — optional; defaults to NULL; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, elements of which contains information about the cancelled fine-tune.

See Also

Other fine-tune functions: create_fine_tune(), delete_fine_tune_model(), list_fine_tune_events(), list_fine_tunes(), retrieve_fine_tune()

Examples

```
## Not run:
training_file <- system.file(
    "extdata", "sport_prepared_train.jsonl", package = "openai"
)
validation_file <- system.file(
    "extdata", "sport_prepared_train.jsonl", package = "openai"
)

training_info <- upload_file(training_file, "fine-tune")
validation_info <- upload_file(validation_file, "fine-tune")

info <- create_fine_tune(
    training_file = training_info$id,
    validation_file = validation_info$id,
    model = "ada",
    compute_classification_metrics = TRUE,
    classification_positive_class = " baseball" # Mind space in front
)

id <- ifelse(
    length(info$data$id) > 1,
    info$data$id[length(info$data$id)],
    info$data$id
)

cancel_fine_tune(fine_tune_id = id)
```

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

`create_chat_completion`

Create chat completion

Description

Creates a completion for the chat message. See this page for details.

Usage

```
create_chat_completion(
  model,
  messages = NULL,
  temperature = 1,
  top_p = 1,
  n = 1,
  stream = FALSE,
  stop = NULL,
  max_tokens = NULL,
  presence_penalty = 0,
  frequency_penalty = 0,
  logit_bias = NULL,
  user = NULL,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`model` required; a length one character vector.

`messages` required; defaults to `NULL`; a list in the following format: `list(list("role" = "user", "content" = "H`

`temperature` required; defaults to `1`; a length one numeric vector with the value between `0` and `2`.

`top_p` required; defaults to `1`; a length one numeric vector with the value between `0` and `1`.

`n` required; defaults to `1`; a length one numeric vector with the integer value greater than `0`.

`stream` required; defaults to `FALSE`; a length one logical vector. **Currently is not implemented.**

`stop` optional; defaults to `NULL`; a character vector of length between one and four.

`max_tokens` required; defaults to `(4096 - prompt tokens)`; a length one numeric vector with the integer value greater than `0`.

`presence_penalty`
: required; defaults to `0`; a length one numeric vector with a value between `-2` and `2`.

`frequency_penalty`
: required; defaults to `0`; a length one numeric vector with a value between `-2` and `2`.

`logit_bias`
: optional; defaults to `NULL`; a named list.

`user`
: optional; defaults to `NULL`; a length one character vector.

`openai_api_key`
: required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.

`openai_organization`
: optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, elements of which contain chat completion(s) and supplementary information.

Examples

```
## Not run:
create_chat_completion(
   model = "gpt-3.5-turbo",
   messages = list(
       list(
           "role" = "system",
           "content" = "You are a helpful assistant."
       ),
       list(
           "role" = "user",
           "content" = "Who won the world series in 2020?"
       ),
       list(
           "role" = "assistant",
           "content" = "The Los Angeles Dodgers won the World Series in 2020."
       ),
       list(
           "role" = "user",
           "content" = "Where was it played?"
       )
   )
)

## End(Not run)
```

create_completion *Create completion*

Description

Creates a completion based on the provided prompt and parameters. See this page for details.

Usage

```
create_completion(
  engine_id = deprecated(),
  model,
  prompt = "<|endoftext|>",
  suffix = NULL,
  max_tokens = 16,
  temperature = 1,
  top_p = 1,
  n = 1,
  stream = FALSE,
  logprobs = NULL,
  echo = FALSE,
  stop = NULL,
  presence_penalty = 0,
  frequency_penalty = 0,
  best_of = 1,
  logit_bias = NULL,
  user = NULL,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

engine_id	**[Deprecated]**
model	required; a length one character vector.
prompt	required; defaults to "<|endoftext|>"; an arbitrary length character vector.
suffix	optional; defaults to NULL; a length one character vector.
max_tokens	required; defaults to 16; a length one numeric vector with the integer value greater than 0.
temperature	required; defaults to 1; a length one numeric vector with the value between 0 and 2.
top_p	required; defaults to 1; a length one numeric vector with the value between 0 and 1.
n	required; defaults to 1; a length one numeric vector with the integer value greater than 0.

`stream` required; defaults to `FALSE`; a length one logical vector. **Currently is not implemented.**

`logprobs` optional; defaults to `NULL`; a length one numeric vector with the integer value between `0` and `5`.

`echo` required; defaults to `FALSE`; a length one logical vector.

`stop` optional; defaults to `NULL`; a character vector of length between one and four.

`presence_penalty` required; defaults to `0`; a length one numeric vector with a value between `-2` and `2`.

`frequency_penalty` required; defaults to `0`; a length one numeric vector with a value between `-2` and `2`.

`best_of` required; defaults to `1`; a length one numeric vector with the integer value greater than `0`.

`logit_bias` optional; defaults to `NULL`; a named list.

`user` optional; defaults to `NULL`; a length one character vector.

`openai_api_key` required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.

`openai_organization` optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, elements of which contain completion(s) and supplementary information.

Examples

```
## Not run:
create_completion(
    model = "text-davinci-002",
    prompt = "Say this is a test",
    max_tokens = 5
)

logit_bias <- list(
    "11" = -100,
    "13" = -100
)
create_completion(
    model = "ada",
    prompt = "Generate a question and an answer",
    n = 4,
    best_of = 4,
    logit_bias = logit_bias
)

## End(Not run)
```

create_edit *Create edit*

Description

Creates an edit based on the provided input, instruction, and parameters. See this page for details.

Usage

```
create_edit(
  engine_id = deprecated(),
  model,
  input = "\"",
  instruction,
  temperature = 1,
  top_p = 1,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`engine_id`	**[Deprecated]**
`model`	required; a length one character vector.
`input`	required; defaults to `'"'`; a length one character vector.
`instruction`	required; a length one character vector.
`temperature`	required; defaults to `1`; a length one numeric vector with the value between `0` and `2`.
`top_p`	required; defaults to `1`; a length one numeric vector with the value between `0` and `1`.
`openai_api_key`	required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.
`openai_organization`	optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, elements of which contain edited version of prompt and supplementary information.

Examples

```
## Not run:
create_edit(
    model = "text-davinci-edit-001",
    input = "What day of the wek is it?",
    instruction = "Fix the spelling mistakes"
)

## End(Not run)
```

`create_embedding` *Create embeddings*

Description

Creates an embedding vector that represents the provided input. See this page for details.

Usage

```
create_embedding(
  engine_id = deprecated(),
  model,
  input,
  user = NULL,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`engine_id` **[Deprecated]**

`model` required; a length one character vector.

`input` required; an arbitrary length character vector.

`user` optional; defaults to `NULL`; a length one character vector.

`openai_api_key` required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.

`openai_organization` optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, an element of which contains embedding vector(s) for a given input.

Examples

```
## Not run:
create_embedding(
    model = "text-embedding-ada-002",
    input = c(
        "Ah, it is so boring to write documentation",
        "But examples are really crucial"
    )
)

## End(Not run)
```

`create_fine_tune` *Create fine-tune*

Description

Creates a job that fine-tunes a specified model based on a given dataset. See this page for details.

Usage

```
create_fine_tune(
  training_file,
  validation_file = NULL,
  model,
  n_epochs = 4,
  batch_size = NULL,
  learning_rate_multiplier = NULL,
  prompt_loss_weight = 0.1,
  compute_classification_metrics = FALSE,
  classification_n_classes = NULL,
  classification_positive_class = NULL,
  classification_betas = NULL,
  suffix = NULL,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`training_file` required; a length one character vector.

`validation_file`
optional; defaults to `NULL`; a length one character vector.

`model` required; a length one character vector.

`n_epochs` required; defaults to `4`; a length one numeric vector with the integer value greater than `0`.

`batch_size` optional; defaults to `NULL`; a length one numeric vector with the integer value greater than `0`.

`learning_rate_multiplier`
optional; defaults to `NULL`; a length one numeric vector with the value greater than `0`.

`prompt_loss_weight`
required; defaults to `0.1`; a length one numeric vector.

`compute_classification_metrics`
required; defaults to `FLASE`; a length one logical vector.

`classification_n_classes`
optional; defaults to `NULL`; a length one numeric vector with the value greater than `0`.

`classification_positive_class`
optional; defaults to `NULL`; a length one character vector.

`classification_betas`
optional; defaults to `NULL`; a list elements of which are numeric values greater than `0`.

`suffix` optional; defaults to `NULL`; a length one character vector.

`openai_api_key` required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.

`openai_organization`
optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, elements of which contain information about the fine-tune.

See Also

Other fine-tune functions: `cancel_fine_tune()`, `delete_fine_tune_model()`, `list_fine_tune_events()`, `list_fine_tunes()`, `retrieve_fine_tune()`

Examples

```
## Not run:
training_file <- system.file(
    "extdata", "sport_prepared_train.jsonl", package = "openai"
)
validation_file <- system.file(
    "extdata", "sport_prepared_train.jsonl", package = "openai"
)

training_info <- upload_file(training_file, "fine-tune")
validation_info <- upload_file(validation_file, "fine-tune")

info <- create_fine_tune(
    training_file = training_info$id,
    validation_file = validation_info$id,
    model = "ada",
    compute_classification_metrics = TRUE,
    classification_positive_class = " baseball" # Mind space in front
)

## End(Not run)
```

`create_image` *Create image*

Description

Creates an image given a prompt. See this page for details.

Usage

```
create_image(
  prompt,
  n = 1,
  size = c("1024x1024", "256x256", "512x512"),
  response_format = c("url", "b64_json"),
  user = NULL,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`prompt`	required; a length one character vector.
`n`	required; defaults to `1`; a length one numeric vector with the integer value greater than `0`.
`size`	required; defaults to `"1024x1024"`; a length one character vector, one among `"256x256"`, `"512x512"`, and `"1024x1024"`.

`response_format`
: required; defaults to `"url"`; a length one character vector, one among `"url"` and `"b64_json"`.

`user`
: optional; defaults to `NULL`; a length one character vector.

`openai_api_key`
: required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.

`openai_organization`
: optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, an element of which contain either a link to the generated image or the generated image decoded in Base64.

See Also

Other image functions: `create_image_edit()`, `create_image_variation()`

Examples

```
## Not run:
create_image("An astronaut riding a horse in a photorealistic style")

## End(Not run)
```

`create_image_edit` *Create image edit*

Description

Creates an edited or extended image given an original image and a prompt. See this page for details.

Usage

```
create_image_edit(
  image,
  mask,
  prompt,
  n = 1,
  size = c("1024x1024", "256x256", "512x512"),
  response_format = c("url", "b64_json"),
  user = NULL,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

image required; a length one character vector.

mask required; a length one character vector.

prompt required; a length one character vector.

n required; defaults to 1; a length one numeric vector with the integer value greater than 0.

size required; defaults to "1024x1024"; a length one character vector, one among "256x256", "512x512", and "1024x1024".

response_format
required; defaults to "url"; a length one character vector, one among "url" and "b64_json".

user optional; defaults to NULL; a length one character vector.

openai_api_key required; defaults to Sys.getenv("OPENAI_API_KEY") (i.e., the value is retrieved from the .Renviron file); a length one character vector. Specifies OpenAI API key.

openai_organization
optional; defaults to NULL; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, an element of which contain either a link to the edited image or the edited image decoded in Base64.

See Also

Other image functions: create_image_variation(), create_image()

Examples

```
## Not run:
image <- system.file("extdata", "astronaut.png", package = "openai")
mask <- system.file("extdata", "mask.png", package = "openai")
create_image_edit(
    image = image,
    mask = mask,
    prompt = "goat",
    n = 1,
    response_format = "url"
)

## End(Not run)
```

`create_image_variation`
Create image variation

Description

Creates a variation of a given image. See this page for details.

Usage

```
create_image_variation(
  image,
  n = 1,
  size = c("1024x1024", "256x256", "512x512"),
  response_format = c("url", "b64_json"),
  user = NULL,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`image`	required; a length one character vector.
`n`	required; defaults to `1`; a length one numeric vector with the integer value greater than `0`.
`size`	required; defaults to `"1024x1024"`; a length one character vector, one among `"256x256"`, `"512x512"`, and `"1024x1024"`.
`response_format`	required; defaults to `"url"`; a length one character vector, one among `"url"` and `"b64_json"`.
`user`	optional; defaults to `NULL`; a length one character vector.
`openai_api_key`	required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.
`openai_organization`	optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, an element of which contain either a link to the image variation or the image variation decoded in Base64.

See Also

Other image functions: `create_image_edit()`, `create_image()`

Examples

```
## Not run:
image <- system.file("extdata", "astronaut.png", package = "openai")
create_image_variation(
    image = image,
    n = 1,
    size = "256x256",
    response_format = "url"
)

## End(Not run)
```

---

create_moderation   *Create moderation*

Description

Classifies if text violates OpenAI's Content Policy. See this page for details.

Usage

```
create_moderation(
  input,
  model,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`input` required; an arbitrary length character vector.

`model` required; a length one character vector.

`openai_api_key` required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.

`openai_organization`
optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, elements of which contain information about the model.

Examples

```
## Not run:
create_moderation(
    input = "I want to kill them all.",
    model = "text-moderation-stable"
)

## End(Not run)
```

`create_transcription` *Create transcription*

Description

Transcribes audio into the input language. See this page for details.

Usage

```
create_transcription(
  file,
  model,
  prompt = NULL,
  response_format = "json",
  temperature = 0,
  language = NULL,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`file`	required; a length one character vector.
`model`	required; a length one character vector.
`prompt`	optional; defaults to `NULL`; a length one character vector.
`response_format`	required; defaults to `"json"`; length one character vector equals to `"json"`. **Currently only `"json"` is implemented.**
`temperature`	required; defaults to `1`; a length one numeric vector with the value between `0` and `2`.

`language` optional; defaults to `NULL`; a length one character vector.

`openai_api_key` required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.

`openai_organization`
optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, elements of which contain a transcription and supplementary information.

See Also

Other audio functions: `create_translation()`

Examples

```
## Not run:
voice_sample_en <- system.file(
    "extdata", "sample-en.m4a", package = "openai"
)
create_transcription(file = voice_sample_en, model = "whisper-1")

## End(Not run)
```

`create_translation` *Create translation*

Description

Translates audio into into English. See this page for details.

Usage

```
create_translation(
  file,
  model,
  prompt = NULL,
  response_format = "json",
  temperature = 0,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`file` required; a length one character vector.

`model` required; a length one character vector.

`prompt` optional; defaults to `NULL`; a length one character vector.

`response_format` required; defaults to `"json"`; length one character vector equals to `"json"`. **Currently only `"json"` is implemented.**

`temperature` required; defaults to `1`; a length one numeric vector with the value between `0` and `2`.

`openai_api_key` required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.

`openai_organization` optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, elements of which contain a transcription and supplementary information.

See Also

Other audio functions: `create_transcription()`

Examples

```
## Not run:
voice_sample_ua <- system.file(
    "extdata", "sample-ua.m4a", package = "openai"
)
create_translation(file = voice_sample_ua, model = "whisper-1")

## End(Not run)
```

`delete_file` *Delete file*

Description

Deletes a file. See this page for details.

Usage

```
delete_file(
  file_id,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`file_id` required; a length one character vector.

`openai_api_key` required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.

`openai_organization`
optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, elements of which contains ID of the deleted file and status whether the file is deleted.

See Also

Other file functions: `list_files()`, `retrieve_file_content()`, `retrieve_file()`, `upload_file()`

Examples

```
## Not run:
file <- system.file("extdata", "classification-file.jsonl", package = "openai")
file_info <- upload_file(file = file, purpose = "classification")
delete_file(file_info$id)

## End(Not run)
```

`delete_fine_tune_model` *Delete fine_tune model*

Description

Deletes a fine-tuned model. See this page for details.

Usage

```
delete_fine_tune_model(
  model,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`model`	required; a length one character vector.
`openai_api_key`	required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.
`openai_organization`	optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, elements of which contains information about the deleted model.

See Also

Other fine-tune functions: `cancel_fine_tune()`, `create_fine_tune()`, `list_fine_tune_events()`, `list_fine_tunes()`, `retrieve_fine_tune()`

Examples

```
## Not run:
fine_tunes <- list_fine_tunes()

fine_tunes <- fine_tunes$data

id <- fine_tunes[!is.na(fine_tunes[, "fine_tuned_model"]), "fine_tuned_model"]

delete_fine_tune_model(model = id[1])

## End(Not run)
```

list_files *List files*

Description

Lists files uploaded by user's organization. See this page for details.

Usage

```
list_files(
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`openai_api_key`	required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.
`openai_organization`	optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, an element of which is a data frame containing information about files.

See Also

Other file functions: `delete_file()`, `retrieve_file_content()`, `retrieve_file()`, `upload_file()`

Examples

```
## Not run:
list_files()

## End(Not run)
```

list_fine_tunes *Lists fine-tunes*

Description

Lists organization's fine-tuning jobs. See this page for details.

Usage

```
list_fine_tunes(
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`openai_api_key` required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.

`openai_organization` optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, an element of which is a data frame containing information about fine-tunes.

See Also

Other fine-tune functions: `cancel_fine_tune()`, `create_fine_tune()`, `delete_fine_tune_model()`, `list_fine_tune_events()`, `retrieve_fine_tune()`

Examples

```
## Not run:
list_fine_tunes()

## End(Not run)
```

list_fine_tune_events *List fine-tune events*

Description

Returns events related to a specified fine-tune job. See this page for details.

Usage

```
list_fine_tune_events(
  fine_tune_id,
  stream = FALSE,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

fine_tune_id	required; a length one character vector.
stream	required; defaults to FALSE; a length one logical vector. **Currently is not implemented.**
openai_api_key	required; defaults to Sys.getenv("OPENAI_API_KEY") (i.e., the value is retrieved from the .Renviron file); a length one character vector. Specifies OpenAI API key.
openai_organization	optional; defaults to NULL; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, elements of which contains information about the fine-tune events.

See Also

Other fine-tune functions: cancel_fine_tune(), create_fine_tune(), delete_fine_tune_model(), list_fine_tunes(), retrieve_fine_tune()

Examples

```
## Not run:
training_file <- system.file(
    "extdata", "sport_prepared_train.jsonl", package = "openai"
)
```

```
validation_file <- system.file(
    "extdata", "sport_prepared_train.jsonl", package = "openai"
)

training_info <- upload_file(training_file, "fine-tune")
validation_info <- upload_file(validation_file, "fine-tune")

info <- create_fine_tune(
    training_file = training_info$id,
    validation_file = validation_info$id,
    model = "ada",
    compute_classification_metrics = TRUE,
    classification_positive_class = " baseball" # Mind space in front
)

id <- ifelse(
    length(info$data$id) > 1,
    info$data$id[length(info$data$id)],
    info$data$id
)

list_fine_tune_events(fine_tune_id = id)

## End(Not run)
```

list_models   *List models*

Description

Lists the currently available models, and provides basic information about each one such as the owner and availability. See this page for details.

Usage

```
list_models(
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

openai_api_key required; defaults to Sys.getenv("OPENAI_API_KEY") (i.e., the value is retrieved from the .Renviron file); a length one character vector. Specifies OpenAI API key.

openai_organization
optional; defaults to NULL; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, an element of which is a data frame containing information about models.

See Also

Other model functions: `retrieve_model()`

Examples

```
## Not run:
list_models()

## End(Not run)
```

`retrieve_file` *Retrieve file*

Description

Provides information about a specific file. See this page for details.

Usage

```
retrieve_file(
  file_id,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`file_id` required; a length one character vector.

`openai_api_key` required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.

`openai_organization` optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, elements of which contains information about the file.

See Also

Other file functions: `delete_file()`, `list_files()`, `retrieve_file_content()`, `upload_file()`

Examples

```
## Not run:
file <- system.file("extdata", "classification-file.jsonl", package = "openai")
file_info <- upload_file(file = file, purpose = "classification")
retrieve_file(file_info$id)

## End(Not run)
```

`retrieve_file_content` *Retrieve file content*

Description

Returns the content of the specified file. See this page for details. Please note that only output files are allowed to be downloaded, not the input ones.

Usage

```
retrieve_file_content(
  file_id,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`file_id` required; a length one character vector.

`openai_api_key` required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.

`openai_organization` optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, an element of which contains the content of the file.

See Also

Other file functions: `delete_file()`, `list_files()`, `retrieve_file()`, `upload_file()`

`retrieve_fine_tune` *Retrieve fine-tune*

Description

Returns information about the specified fine-tune job. See this page for details.

Usage

```
retrieve_fine_tune(
  fine_tune_id,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`fine_tune_id`	required; a length one character vector.
`openai_api_key`	required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.
`openai_organization`	optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, elements of which contains information about the fine-tune.

See Also

Other fine-tune functions: `cancel_fine_tune()`, `create_fine_tune()`, `delete_fine_tune_model()`, `list_fine_tune_events()`, `list_fine_tunes()`

Examples

```
## Not run:
training_file <- system.file(
    "extdata", "sport_prepared_train.jsonl", package = "openai"
)
validation_file <- system.file(
    "extdata", "sport_prepared_train.jsonl", package = "openai"
)

training_info <- upload_file(training_file, "fine-tune")
validation_info <- upload_file(validation_file, "fine-tune")

info <- create_fine_tune(
    training_file = training_info$id,
    validation_file = validation_info$id,
    model = "ada",
    compute_classification_metrics = TRUE,
    classification_positive_class = " baseball" # Mind space in front
)

id <- ifelse(
    length(info$data$id) > 1,
    info$data$id[length(info$data$id)],
    info$data$id
)

retrieve_fine_tune(fine_tune_id = id)

## End(Not run)
```

retrieve_model *Retrieve model*

Description

Retrieves a model instance, providing basic information about the model such as the owner and permissioning. See this page for details.

Usage

```
retrieve_model(
  model,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

model
: required; a length one character vector.

openai_api_key
: required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.

openai_organization
: optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, elements of which contain information about the model.

See Also

Other model functions: `list_models()`

Examples

```
## Not run:
retrieve_model("text-davinci-002")

## End(Not run)
```

`upload_file` *Upload file*

Description

Uploads a file that will be used for various purposes. The size of the storage is limited to 1 Gb. See this page for details.

Usage

```
upload_file(
  file,
  purpose = c("search", "answers", "classifications", "fine-tune"),
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

file — required; a length one character vector.

purpose — required; defaults to `"fine-tune"`; a length one character vector equals to`"fine-tune"`.

openai_api_key — required; defaults to `Sys.getenv("OPENAI_API_KEY")` (i.e., the value is retrieved from the `.Renviron` file); a length one character vector. Specifies OpenAI API key.

openai_organization — optional; defaults to `NULL`; a length one character vector. Specifies OpenAI organization.

Details

For arguments description please refer to the official documentation.

Value

Returns a list, elements of which contains ID of the uploaded file and other supplementary information.

See Also

Other file functions: `delete_file()`, `list_files()`, `retrieve_file_content()`, `retrieve_file()`

Examples

```
## Not run:
file <- system.file("extdata", "classification-file.jsonl", package = "openai")
upload_file(file = file, purpose = "classification")

## End(Not run)
```

Index