Package 'openai'

June 7, 2022

Type Package

Title R Wrapper for OpenAI API

Version 0.1.0

Date 2022-06-06

Description An R wrapper of OpenAI API endpoints (see <https://beta.openai.com/docs/introduction> for details). This package covers Engines, Completions, Edits, Files, Fine-tunes, Embeddings and legacy Searches, Classifications, and Answers endpoints.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.2

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

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

Depends R (>= 4.2)

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 2022-06-07 08:30:06 UTC

R topics documented:

`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_answer` *Create answer*

Description

[Deprecated]

Note: This endpoint is deprecated and will be removed on December 3, 2022. Please see Answers Transition Guide for details.

Answers the specified question based on the documents and examples. See this page for details.

Usage

```
create_answer(
  model = c("ada", "babbage", "curie", "davinci"),
  question,
  examples,
  examples_context,
  documents = NULL,
  file = NULL,
  search_model = c("ada", "babbage", "curie", "davinci"),
  max_rerank = 200,
  temperature = 0,
  logprobs = NULL,
  max_tokens = 16,
  stop = NULL,
  n = 1,
  logit_bias = NULL,
  return_metadata = FALSE,
  return_prompt = FALSE,
  expand = NULL,
  user = NULL,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`model`	required; defaults to `"ada"`; a length one character vector, one among `"ada"`, `"babbage"`, `"curie"`, and `"davinci"`.
`question`	required; a length one character vector.
`examples`	required; a list.
`examples_context`	required; a length one character vector.
`documents`	optional; defaults to `NULL`; an arbitrary length character vector.
`file`	optional; defaults to `NULL`; a length one character vector.
`search_model`	required; defaults to `ada`; a length one character vector, one among `"ada"`, `"babbage"`, `"curie"`, and `"davinci"`.
`max_rerank`	required; defaults to `200`; a length one numeric vector with the integer value greater than `0`.

temperature	required; defaults to 0; a length one numeric vector with the value between 0 and 2.
logprobs	optional; defaults to NULL; a length one numeric vector with the integer value between 0 and 5.
max_tokens	required; defaults to 16; a length one numeric vector with the integer value greater than 0.
stop	optional; defaults to NULL; a character vector of length between one and four.
n	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.
return_metadata	required; defaults to FALSE; a length one logical vector.
return_prompt	required; defaults to FALSE; a length one logical vector.
expand	optional; defaults to NULL; a list elements of which are among completion and file.
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 an answer and other supplementary information.

Examples

```
## Not run:
create_answer(
    search_model = "ada",
    model = "curie",
    question = "How many red apples do I have?",
    documents = c("I have five green apples.", "I love oranges."),
    examples_context = "Jack has three brothers and one sister. His sister is sad",
    examples = list(
        c("How many siblings has Jack?", "Three"),
        c("Who is sad?", "Jack's sister is.")
    ),
    max_tokens = 5,
    stop = c("\n", "<|endoftext|>"),
)

## End(Not run)
```

`create_classification` *Create classification*

Description

[Deprecated]

Note: This endpoint is deprecated and will be removed on December 3, 2022. Please see Classifications Transition Guide for details.

Classifies the query based on the provided examples. See this page for details.

Usage

```
create_classification(
  model = c("ada", "babbage", "curie", "davinci"),
  query,
  examples = NULL,
  file = NULL,
  labels = NULL,
  search_model = c("ada", "babbage", "curie", "davinci"),
  temperature = 0,
  logprobs = NULL,
  max_examples = 200,
  logit_bias = NULL,
  return_prompt = FALSE,
  return_metadata = FALSE,
  expand = NULL,
  user = NULL,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`model`	required; defaults to `"ada"`; a length one character vector, one among `"ada"`, `"babbage"`, `"curie"`, and `"davinci"`.
`query`	required; a length one character vector.
`examples`	optional; defaults to `NULL`; a list. A list of examples with labels, in the following format: `list(c("The movie is so interesting.", "Positive"), c("It is quite boring.", "Negative"), ...)`.
`file`	optional; defaults to `NULL`; a length one character vector.
`labels`	optional; defaults to `NULL`; an arbitrary length character vector.
`search_model`	required; defaults to `ada`; a length one character vector, one among `"ada"`, `"babbage"`, `"curie"`, and `"davinci"`.
`temperature`	required; defaults to `0`; a length one numeric vector with the value between `0` and `2`.

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

`max_examples` required; defaults to `200`; a length one numeric vector with the integer value greater than `0`.

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

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

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

`expand` optional; defaults to `NULL`; a list elements of which are among `completion` and `file`.

`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 label and other supplementary information.

Examples

```
## Not run:
create_classification(
    search_model = "ada",
    model = "curie",
    examples = list(
        c("A happy moment", "Positive"),
        c("I am sad.", "Negative"),
        c("I am feeling awesome", "Positive")
    ),
    query = "I'm ok",
    labels = c("Positive", "Negative", "Neutral")
)

## 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,
  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`	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(
    engine = "text-davinci-002",
    prompt = "Say this is a test",
    max_tokens = 5
)

logit_bias <- list(
    "11" = -100,
    "13" = -100
)
create_completion(
    engine_id = "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,
  input = "\"",
  instruction,
  temperature = 1,
  top_p = 1,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

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

Arguments

`engine_id`	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(
    engine_id = "text-similarity-babbage-001",
    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 = c("curie", "ada", "babbage", "davinci"),
  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`	model required; defaults to `"curie"`; a length one character vector, one among `"ada"`, `"babbage"`, `"curie"`, and `"davinci"`.
`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_search` *Create search*

Description

[Deprecated]

Note: This endpoint is deprecated and will be removed on December 3, 2022. Please see Search Transition Guide for details.

Computes similarity scores between the query and provided documents. See this page for details.

Usage

```
create_search(
  engine_id = c("ada", "babbage", "curie", "davinci"),
  query,
  documents = NULL,
  file = NULL,
  max_rerank = 200,
  return_metadata = FALSE,
  user = NULL,
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),
  openai_organization = NULL
)
```

Arguments

`engine_id`	required; defaults to `"ada"`; a length one character vector, one among `"ada"`, `"babbage"`, `"curie"`, and `"davinci"`.
`query`	required; length one character vector.
`documents`	optional; defaults to `NULL`; an arbitrary length character vector.
`file`	optional; defaults to `NULL`; a length one character vector.
`max_rerank`	required; defaults to `200`; a length one numeric vector with the integer value greater than `0`.

`return_metadata`
: required; defaults to `FALSE`; a length one logical 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, elements of which contain score of each document and supplementary information.

Examples

```
## Not run:
create_search(
    documents = c("White House", "hospital", "school"),
    query = "the president"
)

## 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_engines *List engines*

Description

Lists available engines and provides basic information about such engines. See this page for details.

Usage

```
list_engines(
  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 engines.

See Also

Other engine functions: retrieve_engine()

Examples

```
## Not run:
list_engines()

## 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 specidied 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)
```

---

retrieve_engine *Retrieve engine*

Description

Provides information about a specified engine. See this page for details.

Usage

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

Arguments

`engine_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 contain information about the engine.

See Also

Other engine functions: `list_engines()`

Examples

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

## 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)
```

---

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 "search"; a length one character vector, one among "search", "answers", "classifications", and "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