Chat with an OpenAI model

Source: R/provider-openai.R

This is the main interface to OpenAI's models, using the **responses API**. You can use this to access OpenAI's latest models and features like image generation and web search. If you need to use an OpenAI-compatible API from another provider, or the **chat completions** API with OpenAI, use `chat_openai_compatible()` instead.

Note that a ChatGPT Plus membership does not grant access to the API. You will need to sign up for a developer account (and pay for it) at the developer platform.

Usage

```r
chat_openai(
  system_prompt = NULL,
  base_url = "https://api.openai.com/v1",
  api_key = NULL,
  credentials = NULL,
  model = NULL,
  params = NULL,
  api_args = list(),
  api_headers = character(),
  service_tier = c("auto", "default", "flex", "prior
  echo = c("none", "output", "all")
)

models_openai(
```

)

Arguments

system_prompt

A system prompt to set the behavior of the assistant.

base_url

The base URL to the endpoint; the default is OpenAI's public API.

api_key

`lifecycle` `deprecated` Use `credentials` instead.

credentials

Override the default credentials. You generally should not need this argument; instead set the `OPENAI_API_KEY` environment variable. The best place to set this is in `.Renviron`, which you can easily edit by calling `usethis::edit_r_environ()`.

If you do need additional control, this argument takes a zero-argument function that returns either a string (the API key), or a named list (added as additional headers to every request).

model

The model to use for the chat (defaults to "gpt-4.1"). We regularly update the default, so we strongly recommend explicitly specifying a model for anything other than casual use. Use `models_openai()` to see all options.

params

api_args

Named list of arbitrary extra arguments appended to the body of every chat API call. Combined with the body object generated by ellmer with `modifyList()`.

api_headers

Named character vector of arbitrary extra headers appended to every chat API call.

service_tier

Request a specific service tier. There are four options:

- `"auto"` (default): uses the service tier configured in Project settings.

- `"default"` : standard pricing and performance.

- `"flex"` : slower and cheaper.

- `"priority"` : faster and more expensive.

echo

One of the following options:

- `none` : don't emit any output (default when running in a function).

- `output` : echo text and tool-calling output as it streams in (default when running at the console).

- `all` : echo all input and output.

Note this only affects the `chat()` method.

Value

A [Chat](#) object.

chat_aws_bedrock(), chat_azure_openai(),
chat_cloudflare(), chat_databricks(),
chat_deepseek(), chat_github(),
chat_google_gemini(), chat_groq(),
chat_huggingface(), chat_mistral(),
chat_ollama(), chat_openai_compatible(),
chat_openrouter(), chat_perplexity(),
chat_portkey()

Examples

```
chat <- chat_openai()
#> Using model = "gpt-4.1".
chat$chat("
  What is the difference between a tibble and a data
  Answer with a bulleted list
")
#> - **Origin:**
#>   - **Data Frame:** Base R object for storing tab
#>   - **Tibble:** Modern reimagining of data frames
#> tidyverse (tibble package).
#>
#> - **Printing:**
#>   - **Data Frame:** Prints the entire object, pos
#> console.
#>   - **Tibble:** Prints a preview (first 10 rows a
#> screen), making output more readable.
#>
#> - **Subsetting:**
#>   - **Data Frame:** May simplify to a vector when
#> column.
#>   - **Tibble:** Always returns a tibble when subs
#> `[`.
#>
#> - **Column Names:**
#>   - **Data Frame:** Allows non-syntactic names (w
```

```
#>
#> - **Data Types:**
#>   - **Data Frame:** Converts strings to factors b
#> specified otherwise).
#>   - **Tibble:** Does **not** convert strings to †
#>
#> - **Partial Matching:**
#>   - **Data Frame:** Allows partial matching of co
#>   - **Tibble:** Does **not** allow partial matchî
#> names.
#>
#> - **Performance:**
#>   - **Data Frame:** Slightly faster for basic ope
#> simpler structure.
#>   - **Tibble:** Slightly slower but offers better
#> especially for big data analysis.
#>
#> - **Use in Tidyverse:**
#>   - **Data Frame:** Used in base R workflows.
#>   - **Tibble:** Default in tidyverse packages (gg
#>
#> - **Row Names:**
#>   - **Data Frame:** Supports row names.
#>   - **Tibble:** Does not support row names; store
#> needed.
#>
#> In summary, tibbles are a modern and tidyverse-fr
#> data frames with improved usability and printing

chat$chat("Tell me three funny jokes about statistic
#> Absolutely! Here are three funny jokes about stat
#>
#> 1. **Why did the statistician bring a ladder to †
#>    Because they heard the drinks were on the hous
#>
#> 2. **How does a statistician catch a lion?**
#>    They build a cage, label it as "population," a
#> to walk right in—it's all about sampling!
#>
```

ellmer 0.4.0