

Package: chatgpt (via r-universe)

October 6, 2026

Type Package

Title Interface to 'ChatGPT' from R

Version 0.2.4

Maintainer Juan Cruz Rodriguez <jcrodriguez@unc.edu.ar>

Description 'OpenAI's 'ChatGPT' <<https://chat.openai.com/>> coding assistant for 'RStudio'. A set of functions and 'RStudio' addins that aim to help the R developer in tedious coding tasks.

License GPL (>= 3)

URL <https://github.com/jcrodriguez1989/chatgpt>

BugReports <https://github.com/jcrodriguez1989/chatgpt/issues>

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Imports clipr, http, jsonlite, miniUI, rstudioapi, shiny, utils, xfun

Config/pak/sysreqs cmake make libuv1-dev libssl-dev libx11-dev zlib1g-dev

Repository <https://jcrodriguez1989.r-universe.dev>

Date/Publication 2024-12-05 17:59:18 UTC

RemoteUrl <https://github.com/jcrodriguez1989/chatgpt>

RemoteRef HEAD

RemoteSha 16b43dd2dbe418838b2972340f7181b346c86373

Contents

chatgpt-package	2
ask_chatgpt	3
build_prompt_content	3
comment_code	4
complete_code	4

create_unit_tests	5
create_variable_name	6
document_code	6
explain_code	7
find_issues_in_code	7
generate_image	8
get_chat_session	9
get_verbosity	9
gpt_get_completions	9
list_models	10
optimize_code	10
parse_response	11
refactor_code	11
reset_chat_session	12
run_addin	12
run_addin_ask_chatgpt	13

Index	14
--------------	-----------

chatgpt-package	<i>'OpenAI's 'ChatGPT' <https://chat.openai.com/> coding assistant for 'RStudio'. A set of functions and 'RStudio' addins that aim to help the R developer in tedious coding tasks.</i>
-----------------	--

Description

'OpenAI's 'ChatGPT' <https://chat.openai.com/> coding assistant for 'RStudio'. A set of functions and 'RStudio' addins that aim to help the R developer in tedious coding tasks.

Author(s)

Maintainer: Juan Cruz Rodriguez <jcrodriguez@unc.edu.ar>

See Also

Useful links:

- <https://github.com/jcrodriguez1989/chatgpt>
- Report bugs at <https://github.com/jcrodriguez1989/chatgpt/issues>

ask_chatgpt	<i>Ask ChatGPT</i>
-------------	--------------------

Description

Note: See also ‘reset_chat_session‘.

Usage

```
ask_chatgpt(  
  question,  
  session_id = "1",  
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),  
  images = NULL  
)
```

Arguments

question	The question to ask ChatGPT.
session_id	The ID of the session to be used. We can have different conversations by using different session IDs.
openai_api_key	OpenAI’s API key.
images	A list of images to attach to the question. It could be a list of URLs or paths.

Value

A character value with the response generated by ChatGPT.

Examples

```
## Not run:  
cat(ask_chatgpt("What do you think about R language?"))  
  
## End(Not run)
```

build_prompt_content	<i>Build Prompt Content</i>
----------------------	-----------------------------

Description

Build Prompt Content

Usage

```
build_prompt_content(question, images)
```

Arguments

question	The question to ask ChatGPT.
images	A list of images to attach to the question. It could be a list of URLs or paths.

comment_code	<i>ChatGPT: Comment Code</i>
--------------	------------------------------

Description

ChatGPT: Comment Code

Usage

```
comment_code(code = clipr::read_clip(allow_non_interactive = TRUE))
```

Arguments

code	The code to be commented by ChatGPT. If not provided, it will use what's copied on the clipboard.
------	---

Value

A character value with the response generated by ChatGPT.

Examples

```
## Not run:
cat(comment_code("for (i in 1:10) {\n  print(i ** 2)\n}"))

## End(Not run)
```

complete_code	<i>ChatGPT: Complete Code</i>
---------------	-------------------------------

Description

ChatGPT: Complete Code

Usage

```
complete_code(code = clipr::read_clip(allow_non_interactive = TRUE))
```

Arguments

code	The code to be completed by ChatGPT. If not provided, it will use what's copied on the clipboard.
------	---

Value

A character value with the response generated by ChatGPT.

Examples

```
## Not run:
cat(complete_code("# A function to square each element of a vector\nsquare_each <- function("))

## End(Not run)
```

create_unit_tests	<i>ChatGPT: Create Unit Tests</i>
-------------------	-----------------------------------

Description

Create ‘testthat’ test cases for the code.

Usage

```
create_unit_tests(code = clipr::read_clip(allow_non_interactive = TRUE))
```

Arguments

code	The code for which to create unit tests by ChatGPT. If not provided, it will use what’s copied on the clipboard.
------	--

Value

A character value with the response generated by ChatGPT.

Examples

```
## Not run:
cat(create_unit_tests("squared_numbers <- function(numbers) {\n  numbers ^ 2\n}"))

## End(Not run)
```

create_variable_name	<i>ChatGPT: Create Variable Name</i>
----------------------	--------------------------------------

Description

ChatGPT: Create Variable Name

Usage

```
create_variable_name(code = clipr::read_clip(allow_non_interactive = TRUE))
```

Arguments

code	The code for which to give a variable name to its result. If not provided, it will use what's copied on the clipboard.
------	--

Value

A character value with the response generated by ChatGPT.

Examples

```
## Not run:  
cat(create_variable_name("sapply(1:10, function(i) i ** 2)"))  
  
## End(Not run)
```

document_code	<i>ChatGPT: Document Code (in roxygen2 format)</i>
---------------	--

Description

ChatGPT: Document Code (in roxygen2 format)

Usage

```
document_code(code = clipr::read_clip(allow_non_interactive = TRUE))
```

Arguments

code	The code to be documented by ChatGPT. If not provided, it will use what's copied on the clipboard.
------	--

Value

A character value with the response generated by ChatGPT.

Examples

```
## Not run:
cat(document_code("square_numbers <- function(numbers) numbers ** 2"))

## End(Not run)
```

`explain_code`*ChatGPT: Explain Code*

Description

ChatGPT: Explain Code

Usage

```
explain_code(code = clipr::read_clip(allow_non_interactive = TRUE))
```

Arguments

<code>code</code>	The code to be explained by ChatGPT. If not provided, it will use what's copied on the clipboard.
-------------------	---

Value

A character value with the response generated by ChatGPT.

Examples

```
## Not run:
cat(explain_code("for (i in 1:10) {\n  print(i ** 2)\n}"))

## End(Not run)
```

`find_issues_in_code`*ChatGPT: Find Issues in Code*

Description

ChatGPT: Find Issues in Code

Usage

```
find_issues_in_code(code = clipr::read_clip(allow_non_interactive = TRUE))
```

Arguments

code	The code to be analyzed by ChatGPT. If not provided, it will use what's copied on the clipboard.
------	--

Value

A character value with the response generated by ChatGPT.

Examples

```
## Not run:
cat(find_issues_in_code("i <- 0\nwhile (i < 0) {\n  i <- i - 1\n}"))

## End(Not run)
```

generate_image

Generate an Image With DALL-E 3

Description

Generate an Image With DALL-E 3

Usage

```
generate_image(
  prompt,
  out_file = tempfile(fileext = ".png"),
  openai_api_key = Sys.getenv("OPENAI_API_KEY")
)
```

Arguments

prompt	The prompt for image generation.
out_file	The path where to save the generated image.
openai_api_key	OpenAI's API key.

get_chat_session	<i>Get Chat Session</i>
------------------	-------------------------

Description

Get Chat Session

Usage

```
get_chat_session(session_id = "1")
```

Arguments

session_id	The ID of the session to be used. If 'NULL', it will return an empty session.
------------	---

get_verbosity	<i>Get Verbosity Level</i>
---------------	----------------------------

Description

Get Verbosity Level

Usage

```
get_verbosity()
```

gpt_get_completions	<i>Get GPT Completions Endpoint</i>
---------------------	-------------------------------------

Description

Get GPT Completions Endpoint

Usage

```
gpt_get_completions(  
  prompt,  
  openai_api_key = Sys.getenv("OPENAI_API_KEY"),  
  messages = NULL  
)
```

Arguments

prompt	The prompt to generate completions for.
openai_api_key	OpenAI's API key.
messages	Available variable, to send the needed messages list to ChatGPT.

`list_models`*ChatGPT: List Models*

Description

ChatGPT: List Models

Usage

```
list_models(openai_api_key = Sys.getenv("OPENAI_API_KEY"))
```

Arguments

`openai_api_key` OpenAI's API key.

Value

A data.frame with the available models to be used by OpenAI's API.

Examples

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

`optimize_code`*ChatGPT: Optimize Code*

Description

ChatGPT: Optimize Code

Usage

```
optimize_code(code = clipr::read_clip(allow_non_interactive = TRUE))
```

Arguments

`code` The code to be optimized by ChatGPT. If not provided, it will use what's copied on the clipboard.

Value

A character value with the response generated by ChatGPT.

Examples

```
## Not run:
cat(optimize_code("i <- 10\nwhile (i > 0) {\n  i <- i - 1\n  print(i)\n}"))

## End(Not run)
```

parse_response	<i>Parse OpenAI API Response</i>
----------------	----------------------------------

Description

Takes the raw response from the OpenAI API and extracts the text content from it.

Usage

```
parse_response(raw_responses, verbosity = get_verbosity())
```

Arguments

`raw_responses` The raw response object returned by the OpenAI API.
`verbosity` The verbosity level for this function.

Value

Returns a character vector containing the text content of the response.

refactor_code	<i>ChatGPT: Refactor Code</i>
---------------	-------------------------------

Description

ChatGPT: Refactor Code

Usage

```
refactor_code(code = clipr::read_clip(allow_non_interactive = TRUE))
```

Arguments

`code` The code to be refactored by ChatGPT. If not provided, it will use what's copied on the clipboard.

Value

A character value with the response generated by ChatGPT.

Examples

```
## Not run:
cat(refactor_code("i <- 10\nwhile (i > 0) {\n  i <- i - 1\n  print(i)\n}"))

## End(Not run)
```

reset_chat_session	<i>Reset Chat Session</i>
--------------------	---------------------------

Description

This function is intended to be used with ‘ask_chatgpt’. If we are using ‘ask_chatgpt’ to chat with ChatGPT, and we want to start a new conversation, we must call ‘reset_chat_session’.

Usage

```
reset_chat_session(
  system_role = "You are a helpful assistant.",
  session_id = "1"
)
```

Arguments

system_role	ChatGPT’s role as an AI assistant.
session_id	The ID of the session to be used. If ‘NULL’, this function will have no effect.

run_addin	<i>Run a ChatGPT RStudio Addin</i>
-----------	------------------------------------

Description

Run a ChatGPT RStudio Addin

Usage

```
run_addin(addin_name)
```

Arguments

addin_name	The name of the adding to execute.
------------	------------------------------------

run_addin_ask_chatgpt *Ask ChatGPT*

Description

Opens an interactive chat session with ChatGPT

Usage

`run_addin_ask_chatgpt()`

Index

ask_chatgpt, [3](#)

build_prompt_content, [3](#)

chatgpt (chatgpt-package), [2](#)

chatgpt-package, [2](#)

comment_code, [4](#)

complete_code, [4](#)

create_unit_tests, [5](#)

create_variable_name, [6](#)

document_code, [6](#)

explain_code, [7](#)

find_issues_in_code, [7](#)

generate_image, [8](#)

get_chat_session, [9](#)

get_verbosity, [9](#)

gpt_get_completions, [9](#)

list_models, [10](#)

optimize_code, [10](#)

parse_response, [11](#)

refactor_code, [11](#)

reset_chat_session, [12](#)

run_addin, [12](#)

run_addin_ask_chatgpt, [13](#)