

Package: ehallm (via r-universe)

September 12, 2024

Type Package

Title What the Package Does (Title Case)

Version 0.1.0

Author Who wrote it

Maintainer The package maintainer <yourself@somewhere.net>

Description More about what it does (maybe more than one line) Use four spaces when indenting paragraphs within the Description.

License What license is it under?

Encoding UTF-8

LazyData true

URL <https://github.com/ecohealthalliance/ehallm>

BugReports <https://github.com/ecohealthalliance/ehallm/issues>

RoxygenNote 7.2.3

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

RemoteUrl <https://github.com/ecohealthalliance/ehallm>

RemoteRef HEAD

RemoteSha 460411c24189570a766d1556e3f532060949a6d9

Contents

format_function_call_df	2
format_openai_results	2
get_function_call_parameter	3
get_function_call_params	4
get_openai_function_call	4
hello	5
openai_chat	6

Index	7
--------------	----------

format_function_call_df

Format the function call dataframe. Nest parameters based on the group column and format the list structure for conversion to JSON

Description

Format the function call dataframe. Nest parameters based on the group column and format the list structure for conversion to JSON

Usage

```
format_function_call_df(function_call_df)
```

Arguments

function_call_df

Value

A formatted tibble

Examples

```
function_call_df <- get_function_call_params() |> format_function_call_df()
```

format_openai_results *Convert the JSON formatted openAI responses into a more human readable nested tibble.*

Description

Convert the JSON formatted openAI responses into a more human readable nested tibble.

Usage

```
format_openai_results(model_response)
```

Arguments

model_response

Value

A nested tibble

Examples

```
#' specific_context <- "Chronic wasting disease (CWD) is a fatal, prion disease of cervids that was first detected in  
common_context <- "Please use the extract_outbreak_data function to extract outbreak details from the given abstract  
chat_results <- openai_chat(specific_context, common_context)  
formatted_chat_results <- format_openai_results(chat_results)
```

`get_function_call_parameter`

Arrange parameter details in JSON schema format

Description

Arrange parameter details in JSON schema format

Usage

```
get_function_call_parameter(  
  parameter_name,  
  description,  
  type = c("string", "null"),  
  enum = NULL  
)
```

Arguments

<code>parameter_name</code>	The name of the parameter openAI should return
<code>description</code>	A description of the parameter. OpenAI will use this description to construct it's response
<code>type</code>	What type or types are allowed in OpenAI's response. i.e. <code>c("string", "null")</code> constrains openAI to returning a character string or NULL.
<code>enum</code>	A list of the return values that OpenAI should adhere to when choosing a response.

Value

A formatted and named parameter list

Examples

```
function_call_parameter <- get_function_call_parameter("disease", "The outbreak event disease name")
```

get_function_call_params

Construct a tibble of parameters that openAI will return and specify constraints such as type and enumeration

Description

Construct a tibble of parameters that openAI will return and specify constraints such as type and enumeration

Usage

```
get_function_call_params(country = NA, disease = NA)
```

Arguments

country	Country context can be added to parameter descriptions by modifying the tribble to include glue or paste statements
disease	Disease context can be added to parameter descriptions by modifying the tribble to include glue or paste statements

Value

A tibble outlining the parameters that openAI should use in it's reply. Will be further processed by `format_function_call_df`

Examples

```
function_call_params <- get_function_call_params()
```

get_openai_function_call

context aware openai function call.

Description

context aware openai function call.

Usage

```
get_openai_function_call(  
  country = NA,  
  disease = NA,  
  function_name = "extract_outbreak_data",  
  function_description = "Extract outbreak details from provided abstract"  
)
```

Arguments

country	Passed to <code>get_function_call_params</code> if parameter descriptions need to include context
disease	Passed to <code>get_function_call_params</code> if parameter descriptions need to include context
function_name	The name of the imaginary function openAI will be formatting it's response for. Should be descriptive of the desired task it will perform.
function_description	# A description of the what the function would accomplish

Value

Returns a list containing a formatted function call as an R list, as a compact JSON, a pretty JSON, and a tibble of the parameters used in constructing the call.

Examples

```
function_call <- get_openai_function_call()
```

hello	<i>Hello, World!</i>
-------	----------------------

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

openai_chat	<i>Initiate a conversation with openAI using the provided model and system, common, and specific context. Note: to use these functions you need to provide an openai key in the .env file as in:</i>
-------------	--

Description

Initiate a conversation with openAI using the provided model and system, common, and specific context. Note: to use these functions you need to provide an openai key in the .env file as in:

Usage

```
openai_chat(
  specific_context,
  common_context,
  hint = NA,
  model = "gpt-3.5-turbo",
  system_context =
    "You act as a function to extract outbreak information from a provided abstract."
)
```

Arguments

specific_context	This is the text specific to each query you would like OpenAI to consider
common_context	This is the text OpenAI should consider in every query
hint	An optional hint to get OpenAI on the right track.
model	The identity of the model you would like to submit the context to.
system_context	Text explaining to the model what role you would like it to play. In example, "Respond as if you an 18th century naturalist."

Value

A tibble containing the responses from openAI for each query

Examples

```
specific_context <- "Chronic wasting disease (CWD) is a fatal, prion disease of cervids that was first detected in A
common_context <- "Please use the extract_outbreak_data function to extract outbreak details from the given abstract
chat_results <- openai_chat(specific_context, common_context)
```

Index

[format_function_call_df](#), 2
[format_openai_results](#), 2

[get_function_call_parameter](#), 3
[get_function_call_params](#), 4
[get_openai_function_call](#), 4

[hello](#), 5

[openai_chat](#), 6