

Package: quackingllama (via r-universe)

March 28, 2025

Title Process text with Ollama, retrieve structured results, cache them locally in DuckDB

Version 0.0.0.9013

Description Process text with Ollama, store results in DuckDB.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports base64enc, cli, dplyr, duckdb, fs, httpuv, httr2, jsonlite, purrr, rlang, stringr, tibble, yyjsonr

Depends R (>= 3.5)

LazyData true

URL <https://giocomai.github.io/quackingllama/>

Config/pak/sysreqs make libicu-dev libssl-dev xz-utils zlib1g-dev

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

RemoteUrl <https://github.com/giocomai/quackingllama>

RemoteRef HEAD

RemoteSha aa6e5075214c36bd5f5c4bdc549fcc53499fbb

Contents

ql_disable_db	2
ql_enable_db	2
ql_generate	3
ql_get_db_options	4
ql_get_models	4
ql_get_options	5
ql_hash	6
ql_prompt	7
ql_read_images	8

ql_request	9
ql_set_db_options	9
ql_set_options	10

Index	12
--------------	-----------

ql_disable_db	<i>Disable caching for the current session</i>
---------------	------------------------------------------------

Description

Disable caching for the current session

Usage

```
ql_disable_db()
```

Value

Nothing, used for its side effects.

See Also

Other database: [ql_enable_db\(\)](#), [ql_set_db_options\(\)](#)

Examples

```
ql_disable_db()
```

ql_enable_db	<i>Enable storing data in a database for the current session</i>
--------------	------------------------------------------------------------------

Description

Enable storing data in a database for the current session

Usage

```
ql_enable_db(db_type = "DuckDB")
```

Arguments

db_type	Defaults to DuckDB.
---------	---------------------

Value

Nothing, used for its side effects.

See Also

Other database: [ql_disable_db\(\)](#), [ql_set_db_options\(\)](#)

Examples

```
ql_enable_db()
```

ql_generate	<i>Generate a response and return the result in a data frame</i>
-------------	------------------------------------------------------------------

Description

Generate a response and return the result in a data frame

Usage

```
ql_generate(
  prompt_df,
  only_cached = FALSE,
  host = NULL,
  message = NULL,
  timeout = NULL,
  error = c("fail", "warn")
)
```

Arguments

prompt_df	A data frame with all inputs passed to the LLM, typically created with ql_prompt() .
only_cached	Defaults to FALSE. If TRUE, only cached responses are returned.
host	The address where the Ollama API can be reached, e.g. <code>http://localhost:11434</code> for locally deployed Ollama.
timeout	If not set with ql_set_options() , defaults to 300 seconds (5 minutes).
error	Defines how errors should be handled, defaults to "fail", i.e. if an error emerges while querying the LLM, the function stops. If set to "warn", it sets the response to <code>NA_character_</code> and stores it in database. This can be useful e.g. for proceed if the prompts include a request that routinely time outs without giving a response. This does not imply that the model would never give a responses, e.g. re-running the same query with longer time out may work.

Value

A data frame, including a response column, as well as other information returned by the model.

Examples

```
## Not run:
ql_prompt("a haiku") |>
  ql_generate()

## End(Not run)
```

ql_get_db_options	<i>Retrieve</i>
-------------------	-----------------

Description

Retrieve

Usage

```
ql_get_db_options(options = c("db", "db_type", "db_folder", "db_filename"))
```

Arguments

options Available options that

Value

A list with the selected options.

Examples

```
ql_get_db_options()

## Retrieve only selected option
ql_get_db_options("db_type")
```

ql_get_models	<i>Get available models</i>
---------------	-----------------------------

Description

Get available models

Usage

```
ql_get_models(host = "http://localhost:11434")
```

Arguments

host Defaults to "http://localhost:11434", where locally deployed Ollama usually responds.

Value

A data frame (a tibble) with details on all locally available models.

Examples

```
## Not run:
ql_get_models()

## End(Not run)
```

ql_get_options	<i>Get options</i>
----------------	--------------------

Description

Get options

Usage

```
ql_get_options(
  options = c("system", "model", "host", "temperature", "seed", "keep_alive", "timeout"),
  system = NULL,
  model = NULL,
  host = NULL,
  temperature = NULL,
  seed = NULL,
  keep_alive = NULL,
  timeout = NULL
)
```

Arguments

options A character vector used to filter which options should effectively be returned. Defaults to all available.

Value

A list with all available options (or those selected with options)

Examples

```
ql_set_options(  
  model = "llama3.2",  
  host = "http://localhost:11434",  
  system = "You are a helpful assistant.",  
  temperature = 0,  
  seed = 42,  
  keep_alive = "5m"  
)  
  
ql_get_options()
```

ql_hash	<i>Hash all inputs relevant to the call to the LLM and create a hash to be used for caching.</i>
---------	--------------------------------------------------------------------------------------------------

Description

Mostly used internally.

Usage

```
ql_hash(prompt_df)
```

Arguments

prompt_df A data frame with all inputs passed to the LLM, typically created with [ql_prompt\(\)](#).

Value

A tibble, such as those returned by [ql_prompt\(\)](#), but always including a hash column.

Examples

```
ql_prompt("a haiku", hash = FALSE) |> ql_hash()
```

`ql_prompt`*Generate a data frame with all relevant inputs for the LLM.*

Description

Typically passed to `ql_generate()`.

Usage

```
ql_prompt(  
  prompt,  
  system = NULL,  
  format = NULL,  
  model = NULL,  
  images = NULL,  
  temperature = NULL,  
  seed = NULL,  
  host = NULL,  
  hash = TRUE  
)
```

Arguments

<code>prompt</code>	A prompt for the LLM.
<code>system</code>	System message to pass to the model. See official documentation for details. For example: "You are a helpful assistant."
<code>model</code>	The name of the model, e.g. llama3.2 or phi3.5:3.8b. Run <code>ollama list</code> from the command line to see a list of locally available models.
<code>temperature</code>	Numeric value comprised between 0 and 1 passed to the model. When set to 0 and with the same seed, the response to the same prompt is always exactly the same. When closer to one, the response is more variable and creative. Use 0 for consistent responses. Setting this to 0.7 is a common choice for creative or interactive tasks.
<code>seed</code>	An integer. When temperature is set to 0 and the seed is constant, the model consistently returns the same response to the same prompt.
<code>host</code>	The address where the Ollama API can be reached, e.g. <code>http://localhost:11434</code> for locally deployed Ollama.
<code>hash</code>	Defaults to TRUE. If TRUE, adds a column with the hash of all other components of the prompt. Used internally for caching. Can be added separately with <code>ql_hash()</code> .

Details

For more details and context about each parameter, see <https://github.com/ollama/ollama/blob/main/docs/api.md>.

Value

A tibble with all main components of a query, to be passed to `ql_generate()`.

Examples

```
ql_prompt("a haiku")
```

ql_read_images	<i>Read image in order to pass it to multimodal models</i>
----------------	------------------------------------------------------------

Description

Read image in order to pass it to multimodal models

Usage

```
ql_read_images(path)
```

Arguments

path Path to image file.

Value

A list object of character vectors of base 64 encoded images.

Examples

```
if (interactive()) {
  library("quackingllama")

  img_path <- fs::file_temp(ext = "png")

  download.file(
    url = "https://ollama.com/public/ollama.png",
    destfile = img_path
  )

  resp_df <- ql_prompt(
    prompt = "what is this?",
    images = img_path,
    model = "llama3.2-vision"
  ) |>
  ql_generate()

  resp_df

  resp_df$response
}
```

ql_request	<i>Create http2 request for both generate and chat endpoints</i>
------------	------------------------------------------------------------------

Description

Create http2 request for both generate and chat endpoints

Usage

```
ql_request(
  prompt_df,
  endpoint = "generate",
  host = NULL,
  message = NULL,
  timeout = NULL
)
```

Arguments

endpoint	Defaults to "generate". Must be either "generate" or "chat".
host	The address where the Ollama API can be reached, e.g. <code>http://localhost:11434</code> for locally deployed Ollama.
timeout	If not set with <code>ql_set_options()</code> , defaults to 300 seconds (5 minutes).

Value

A http2 request object.

Examples

```
ql_prompt(prompt = "a haiku")

ql_prompt(prompt = "a haiku") |>
  ql_request() |>
  http2::req_dry_run()
```

ql_set_db_options	<i>Set options for the local database and enables caching</i>
-------------------	---------------------------------------------------------------

Description

Set options for the local database and enables caching

Usage

```
ql_set_db_options(db_filename = NULL, db_type = "DuckDB", db_folder = ".")
```

Arguments

db_filename	Defaults NULL. Internally, defaults to a combination of quackingllama, followed by the name of the model used. Name given to the local database file. Useful for differentiating among different approaches or projects when storing multiple database files in the same folder.
db_type	Defaults to DuckDB.
db_folder	Defaults to ., i.e., to the current working directory.

Value

Nothing, used for its side effects.

See Also

Other database: [ql_disable_db\(\)](#), [ql_enable_db\(\)](#)

Examples

```
ql_set_db_options(db_filename = "testing_ground")
```

ql_set_options	<i>Set basic options for the current session.</i>
----------------	---------------------------------------------------

Description

Set basic options for the current session.

Usage

```
ql_set_options(
  system = NULL,
  model = NULL,
  host = NULL,
  temperature = NULL,
  seed = NULL,
  keep_alive = NULL,
  timeout = NULL
)
```

Arguments

system	System message to pass to the model. See official documentation for details. For example: "You are a helpful assistant."
model	The name of the model, e.g. llama3.2 or phi3.5:3.8b. Run <code>ollama list</code> from the command line to see a list of locally available models.

host	The address where the Ollama API can be reached, e.g. <code>http://localhost:11434</code> for locally deployed Ollama.
temperature	Numeric value comprised between 0 and 1 passed to the model. When set to 0 and with the same seed, the response to the same prompt is always exactly the same. When closer to one, the response is more variable and creative. Use 0 for consistent responses. Setting this to 0.7 is a common choice for creative or interactive tasks.
seed	An integer. When temperature is set to 0 and the seed is constant, the model consistently returns the same response to the same prompt.
keep_alive	Defaults to "5m". Controls controls how long the model will stay loaded into memory following the request.
timeout	Time in seconds before the request times out. Defaults to 300 (corresponding to 5 minutes).

Value

Nothing, used for its side effects. Options can be retrieved with `ql_get_db_options()`

Examples

```
ql_set_options(  
  model = "llama3.2",  
  host = "http://localhost:11434",  
  system = "You are a helpful assistant.",  
  temperature = 0,  
  seed = 42  
)  
  
ql_get_options()
```

Index

* database

- ql_disable_db, 2
- ql_enable_db, 2
- ql_set_db_options, 9

- ql_disable_db, 2, 3, 10
- ql_enable_db, 2, 2, 10
- ql_generate, 3
- ql_generate(), 7, 8
- ql_get_db_options, 4
- ql_get_models, 4
- ql_get_options, 5
- ql_hash, 6
- ql_hash(), 7
- ql_prompt, 7
- ql_prompt(), 3, 6
- ql_read_images, 8
- ql_request, 9
- ql_set_db_options, 2, 3, 9
- ql_set_options, 10
- ql_set_options(), 3, 9