Package 'chatcache'

November 18, 2026

Type Package

Title Persistent Cache for 'ellmer' and 'shinychat' Large Language Model Interactions

Version 0.1.0

Description Provides a persistent SQLite-based cache for user prompts and large language model (LLM) responses in applications built with the 'ellmer' and 'shinychat' packages. Reduces API (application programming interface) costs and latency by storing and retrieving identical or similar chat interactions across sessions and users.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports digest, jsonlite, DBI, RSQLite, cli, rlang (>= 1.1.0)

Suggests ellmer, shiny, shinychat, testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.2

URL https://github.com/shekharchalwadigrepo-hub/chatcache

BugReports https://github.com/shekharchalwadigrepo-hub/chatcache/issues

Language en-US

NeedsCompilation no

Author Shekhar Chalwadi [aut, cre]

Maintainer Shekhar Chalwadi <skrchalwadicg@gmail.com>

Repository CRAN

Date/Publication 2026-11-18 21:30:07 UTC

Contents

cached_chat 2
cached_shinychat_respond 3
chatcache_clear 3
chatcache_get 4
chatcache_init 4
chatcache_list 5
chatcache_set 5
make_cache_key 6

Index **7**

`cached_chat` *Cached ellmer chat call*

Description

Checks the cache before calling an ellmer Chat object. On a miss, the model is queried and the result is stored.

Usage

```
cached_chat(chat, prompt, ..., use_history = TRUE)
```

Arguments

`chat`	An ellmer Chat object.
`prompt`	Character. User prompt.
`...`	Additional arguments passed to `chat$chat()`.
`use_history`	Logical. Whether to include conversation history in the cache key. Default is `TRUE`.

Value

Character string containing the response.

Examples

```
## Not run:
library(ellmer)
chat <- chat_openai(system_prompt = "Be concise.")
answer <- cached_chat(chat, "What is the capital of France?")

## End(Not run)
```

cached_shinychat_respond *Cached response helper for shinychat*

Description

Convenience wrapper that retrieves or generates a response and appends it to a shinychat UI.

Usage

```
cached_shinychat_respond(chat, user_input, chat_id = "chat")
```

Arguments

chat	An ellmer Chat object.
user_input	Character. The user message (typically input$chat_user_input).
chat_id	Character. The shinychat UI id. Default is "chat".

Value

Invisible character response.

chatcache_clear *Clear the cache*

Description

Deletes all stored interactions.

Usage

```
chatcache_clear(confirm = TRUE)
```

Arguments

confirm	Logical. If TRUE (default) and running interactively, asks for confirmation.

Value

Invisible logical indicating whether the cache was cleared.

chatcache_get *Retrieve a cached response*

Description

Looks up a previously stored response by cache key and increments the hit count.

Usage

```
chatcache_get(cache_key, con = NULL)
```

Arguments

cache_key	Character. The key returned by make_cache_key().
con	Optional DBI connection. Uses the global connection if NULL.

Value

A list with elements response, metadata, and hit_count, or NULL if the key is not found.

chatcache_init *Initialize the chat cache*

Description

Creates or connects to the SQLite database used to store prompts and responses.

Usage

```
chatcache_init(cache_dir = NULL)
```

Arguments

cache_dir	Character string. Directory in which to store the cache database. Defaults to the user cache directory via tools::R_user_dir().

Value

An invisible DBI connection object.

Examples

```
## Not run:
chatcache_init()
chatcache_init(cache_dir = tempdir())

## End(Not run)
```

chatcache_list *List cached interactions*

Description

Returns a data frame of recent cached prompt-response pairs.

Usage

```
chatcache_list(limit = 100, con = NULL)
```

Arguments

limit	Integer. Maximum number of rows to return. Default is 100.
con	Optional DBI connection.

Value

A data frame.

chatcache_set *Store a response in the cache*

Description

Inserts or replaces a prompt-response pair in the cache.

Usage

```
chatcache_set(
  cache_key,
  prompt,
  response,
  system_prompt = NULL,
  model = NULL,
  provider = NULL,
  metadata = NULL,
  con = NULL
)
```

Arguments

cache_key	Character. Cache key.
prompt	Character. Original user prompt.
response	Character. LLM response text.
system_prompt	Character or NULL.
model	Character or NULL.
provider	Character or NULL.
metadata	List or NULL. Additional metadata to store as JSON.
con	Optional DBI connection.

Value

Invisible TRUE.

---

make_cache_key *Generate a cache key*

Description

Creates a stable SHA-256 hash from the prompt and relevant context.

Usage

```
make_cache_key(prompt, system_prompt = NULL, model = NULL, history = NULL)
```

Arguments

prompt	Character. The user prompt.
system_prompt	Character or NULL. The system prompt.
model	Character or NULL. The model name.
history	Character vector or NULL. Simplified conversation history.

Value

A character string (SHA-256 hash).

Index

cached_chat, 2
cached_shinychat_respond, 3
chatcache_clear, 3
chatcache_get, 4
chatcache_init, 4
chatcache_list, 5
chatcache_set, 5

make_cache_key, 6
make_cache_key(), *4*