

Package: rgemini (via r-universe)

May 13, 2026

Type Package

Title R interface to Google Gemini API

Version 0.1.0

Author Sebastian Kranz

Maintainer Sebastian Kranz <sebastian.kranz@uni-ulm.de>

Description In development

License GPL>=2.0

Depends jsonlite, httr, restorepoint, dplyr

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Config/pak/sysreqs libssl-dev

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

Date/Publication 2025-07-13 16:43:31 UTC

RemoteUrl <https://github.com/skranz/rgemini>

RemoteRef main

RemoteSha 26d3006eba64ec3063c8a66a289db8d5bbab2717

Contents

arr_resp	2
gemini_cache_context	2
gemini_content	3
gemini_context	3
gemini_context_cache_expire_sec	4
gemini_delete_all_context_caches	5
gemini_delete_context_cache	5
gemini_list_context_caches	6
gemini_media_upload	6
gemini_update_context_cache	7

infer_response_type	7
obj_resp	8
response_schema	9
run_gemini	9
set_gemini_api_key	11

Index 12

arr_resp	<i>Create an Array Response Template</i>
----------	--

Description

This function creates a template for an array response consisting of JSON objects. Each element (object) in the array is expected to have the provided fields.

Usage

```
arr_resp(...)
```

Arguments

... Named elements that define the fields for each JSON object in the array.

Value

A list with class ‘arr_resp’ that serves as a template for an array response.

Examples

```
# Create an array response template with two fields.
arr_template <- arr_resp(city = "Paris", country = "France")
str(arr_template)
```

gemini_cache_context	<i>Cache a Gemini context</i>
----------------------	-------------------------------

Description

Caches a Gemini context to the API.

Usage

```
gemini_cache_context(
  context,
  verbose = TRUE,
  api_key = getOption("gemini_api_key")
)
```

Arguments

context	A 'gemini_context' object.
verbose	Logical; if 'TRUE', prints additional logging information.
api_key	The API key for authentication (default is taken from 'getOption("gemini_api_key")').

Value

The updated 'gemini_context' object with caching details.

gemini_content	<i>Extract content from a more detailed run_gemini response</i>
----------------	---

Description

Extract content from a more detailed run_gemini response

Usage

```
gemini_content(res_df)
```

gemini_context	<i>Create a Gemini context</i>
----------------	--------------------------------

Description

Initializes a Gemini context object with caching options.

Usage

```
gemini_context(
  prompt = NULL,
  model = "gemini-1.5-flash-001",
  media = NULL,
  ttl_sec = 10 * 60,
  role = "user",
  do_cache = TRUE,
  api_key = getOption("gemini_api_key")
)
```

Arguments

<code>prompt</code>	The input prompt for the model.
<code>model</code>	The model to use (default: "gemini-1.5-flash-001").
<code>media</code>	Optional media input.
<code>ttl_sec</code>	Time-to-live for cache in seconds (default: '60*60').
<code>role</code>	The role of the request (default: "user").
<code>do_cache</code>	Logical; if 'TRUE', attempts to cache the context.
<code>api_key</code>	The API key for authentication (default is taken from 'getOption("gemini_api_key")').

Value

A 'gemini_context' object.

`gemini_context_cache_expire_sec`

Get remaining context cache expiration time

Description

Computes the remaining seconds until a cached context expires.

Usage

```
gemini_context_cache_expire_sec(context)
```

Arguments

<code>context</code>	A 'gemini_context' object.
----------------------	----------------------------

Value

The remaining seconds until expiration, or 'NA' if not cached.

```
gemini_delete_all_context_caches
    Delete all cached contexts
```

Description

Deletes all cached contexts from the Gemini API.

Usage

```
gemini_delete_all_context_caches(api_key = getOption("gemini_api_key"))
```

Arguments

`api_key` The API key for authentication (default is taken from `'getOption("gemini_api_key")'`).

Value

The number of caches deleted.

```
gemini_delete_context_cache
    Delete a cached context
```

Description

Deletes a cached context from the Gemini API.

Usage

```
gemini_delete_context_cache(
  context = NULL,
  cache_name = context$cache_name,
  api_key = getOption("gemini_api_key"),
  verbose = FALSE
)
```

Arguments

`context` An optional context object containing the cache name.

`cache_name` The name of the cache to delete (default is extracted from `'context'`).

`api_key` The API key for authentication (default is taken from `'getOption("gemini_api_key")'`).

`verbose` Logical; if TRUE, prints messages about the deletion process.

Value

Returns 'TRUE' if deletion is successful, 'FALSE' otherwise. If 'context' is provided, returns the updated context.

gemini_list_context_caches
List all cached contexts

Description

Retrieves a list of all cached contexts from the Gemini API.

Usage

```
gemini_list_context_caches(api_key = getOption("gemini_api_key"))
```

Arguments

api_key The API key for authentication (default is taken from 'getOption("gemini_api_key")').

Value

A data frame with cached context details, or 'NULL' if no caches are found.

gemini_media_upload *Upload a document or image to Gemini API.*

Description

Returned object or a list of such objects can then be passed to gemini_run. Media should be available for around an hour.

Usage

```
gemini_media_upload(  
  file_path,  
  mime_type = NULL,  
  display_name = NULL,  
  api_key = getOption("gemini_api_key"),  
  verbose = FALSE  
)
```

gemini_update_context_cache
Update cached context expiration

Description

Updates the TTL of a cached Gemini context.

Usage

```
gemini_update_context_cache(
  context,
  ttl_sec = context$ttl_sec,
  only_if_worked = TRUE,
  only_if_expired = TRUE,
  expired_sec_margin = 10,
  verbose = TRUE,
  api_key = getOption("gemini_api_key")
)
```

Arguments

context	A 'gemini_context' object.
ttl_sec	New TTL value (default: existing TTL).
only_if_worked	Logical; if 'TRUE', only updates if caching previously worked.
only_if_expired	Logical; if 'TRUE', only updates if cache is expired.
expired_sec_margin	Margin in seconds before cache is considered expired (default: '10').
api_key	The API key for authentication (default is taken from 'getOption("gemini_api_key")').

Value

The updated 'gemini_context' object.

infer_response_type *Infer the JSON Response Type*

Description

This function inspects the provided value and returns a string representing the JSON type. It distinguishes between objects, arrays, and primitive types.

Usage

```
infer_response_type(x)
```

Arguments

x The value to inspect.

Value

A character string: one of "null", "object", "array", "string", "integer", "number", or "boolean".

obj_resp

Create an Object Response Template

Description

This function creates a template for a single JSON object response. The response is expected to be a JSON object with the provided fields.

Usage

```
obj_resp(...)
```

Arguments

... Named elements that define the fields for the JSON object.

Value

A list with class 'obj_resp' that serves as a template for an object response.

Examples

```
# Create an object response template with three fields.
obj_template <- obj_resp(city = "Paris", country = "France", population = 5.2)
str(obj_template)
```

response_schema	<i>Generate a JSON Schema from a Response Template</i>
-----------------	--

Description

This function recursively generates a JSON Schema based on a response template. The template must be created using either `obj_resp` (for a single object response) or `arr_resp` (for an array response). The function infers the JSON Schema by inspecting the data types of the provided fields without including the example values.

Usage

```
response_schema(example)
```

Arguments

example	A response template created by <code>obj_resp</code> or <code>arr_resp</code> .
---------	---

Value

A list representing the inferred JSON Schema for the provided response template.

run_gemini	<i>Generate Content with Gemini API</i>
------------	---

Description

Sends a text prompt (and optionally one or more media objects) to the Gemini API to generate content.

Usage

```
run_gemini(  
  prompt,  
  model = "gemini-2.0-flash",  
  media = NULL,  
  json_mode = !is.null(response_schema),  
  response_schema = NULL,  
  temperature = 0.1,  
  context = NULL,  
  detailed_results = FALSE,  
  make_content = TRUE,  
  add_prompt = FALSE,  
  verbose = FALSE,  
  api_key = getOption("gemini_api_key")  
)
```

Arguments

prompt	A character string containing the text prompt to be sent to the Gemini API.
model	A character string specifying the Gemini model to use. Defaults to "gemini-2.0-flash".
media	Either a single media object or a list of media objects to be included in the prompt. Call gemini_media_upload to upload the document or media.
json_mode	Logical. If TRUE, expects the response in JSON format. Defaults to !is.null(response_schema).
response_schema	Optional structured output schema for the response. Defaults to NULL. Even if you want well structures JSON output, you may not always want to provide a response schema. I experienced several cases in which the actual output becomes worse if a schema is provided.
temperature	A numeric value controlling the randomness of the output. Defaults to 0.1.
context	A context object generated by <code>gemini_context</code> . Allows to cache long prompt prefixes or uploaded documents.
detailed_results	Logical. If TRUE, returns detailed results of different retrieval steps. Contains also an element <code>has_error</code> indicating if an error occurred and an <code>err_msg</code> . Use this if you want error handling or debugging. If FALSE just returns finally parsed content and throws an error if somewhere in the retrieval process and error occurred.
make_content	Only relevant if <code>detailed_results=TRUE</code> . If set FALSE don't attempt last step to generate main content. If <code>json_mode = TRUE</code> the last content generation step is a source of error with models that don't reliably generate valid json.
add_prompt	Only relevant if <code>detailed_results=TRUE</code> . Then Logical. If TRUE, includes the prompt in the returned <code>res_df</code> . Defaults to FALSE.
verbose	Logical. If TRUE, prints debugging and request information. Defaults to FALSE.
api_key	A character string containing your Gemini API key. Defaults to the value obtained from <code>getOption("gemini_api_key")</code> .

Details

The function builds a JSON payload that includes the provided text prompt as well as any media objects supplied via the `media` parameter. Each media object is appended to the parts of the request under the `file_data` key. The payload is then sent via a POST request to the Gemini API endpoint for content generation.

Value

A list containing the Gemini API response. If `as_data_frame` is TRUE, the response is converted to a data frame.

See Also

[POST, toJSONN](#)

Examples

```
## Not run:  
# Example using only a text prompt:  
result <- run_gemini(prompt = "Tell a joke", verbose = TRUE)  
  
## End(Not run)
```

set_gemini_api_key *Set your Gemini API*

Description

Set your Gemini API

Usage

```
set_gemini_api_key(key = NULL, file = NULL)
```

Index

[arr_resp](#), [2](#)

[gemini_cache_context](#), [2](#)

[gemini_content](#), [3](#)

[gemini_context](#), [3](#)

[gemini_context_cache_expire_sec](#), [4](#)

[gemini_delete_all_context_caches](#), [5](#)

[gemini_delete_context_cache](#), [5](#)

[gemini_list_context_caches](#), [6](#)

[gemini_media_upload](#), [6](#), [10](#)

[gemini_update_context_cache](#), [7](#)

[infer_response_type](#), [7](#)

[obj_resp](#), [8](#)

[POST](#), [10](#)

[response_schema](#), [9](#)

[run_gemini](#), [9](#)

[set_gemini_api_key](#), [11](#)

[toJSON](#), [10](#)