

Package ‘inferencer’

August 4, 2026

Type Package

Title Simple Unified Wrappers for Hosted Foundation Model Inference APIs

Version 0.2.0

Author Oliver Zhou [aut, cre]

Maintainer Oliver Zhou <oliver.yxzhou@gmail.com>

Description Provides lightweight R wrappers for querying and listing models from several hosted foundation model inference platforms, currently including 'OpenAI' <<https://developers.openai.com/api/docs/>>, 'Google Gemini' <<https://ai.google.dev/gemini-api/docs>>, 'Groq' <<https://console.groq.com/docs/>>, 'OpenRouter' <<https://openrouter.ai/docs/>>, 'Cerebras' <<https://inference-docs.cerebras.ai/>>, and 'Ollama Cloud' <<https://docs.ollama.com/cloud>>. The package is designed for simple inference workflows and quick experimentation, with minimal abstraction and a consistent interface across providers. It includes helper functions for model discovery, text generation, embeddings, image generation, and multimodal inputs, while leaving room for future support of provider-specific parameters and advanced options.

License MIT + file LICENSE

URL <https://github.com/OliverLDS/inferencer>

BugReports <https://github.com/OliverLDS/inferencer/issues>

Depends R (>= 4.1.0)

Encoding UTF-8

RoxygenNote 7.2.3

Imports data.table, httr2, jsonlite

Suggests knitr, rmarkdown, testthat

Config/testthat/edition 3

NeedsCompilation no

Repository CRAN

Date/Publication 2026-08-04 06:50:02 UTC

Contents

embed_gemini	2
embed_openrouter	3
extract_openrouter_benchmarks	4
generate_image_gemini	5
generate_image_openrouter	6
list_cerebras_models	7
list_gemini_models	7
list_groq_models	8
list_ollama_models	8
list_openai_models	9
list_openrouter_audio_models	9
list_openrouter_benchmarks	10
list_openrouter_embedding_models	11
list_openrouter_image_models	11
list_openrouter_models	12
list_openrouter_multimodal_models	13
list_openrouter_video_models	13
query_cerebras	14
query_fallback	15
query_gemini	15
query_gemini_content	16
query_groq	18
query_ollama	19
query_openai	19
query_openrouter	20
query_openrouter_content	21
shell-scripts	23
write_gemini_audio	23
Index	25

embed_gemini	<i>Request Gemini Embeddings</i>
--------------	----------------------------------

Description

Calls Gemini embedding endpoints for one or more text inputs.

Usage

```
embed_gemini(  
    input,  
    api_key = Sys.getenv("GEMINI_API_KEY"),  
    model = "gemini-embedding-001",  
    url0 = Sys.getenv("GEMINI_API_URL", unset =  
        "https://generativelanguage.googleapis.com/v1beta/models"),
```

```
task_type = NULL,  
title = NULL,  
output_dimensionality = NULL,  
json_list = FALSE  
)
```

Arguments

- input A non-empty character vector.
- api_key Gemini API key. Defaults to Sys.getenv("GEMINI_API_KEY").
- model Embedding model identifier.
- url0 Base Gemini models URL.
- task_type Optional Gemini embedding taskType.
- title Optional embedding title, used with document-style inputs.
- output_dimensionality
 Optional embedding dimensionality override.
- json_list If TRUE, return the parsed JSON response as a list.

Value

A numeric matrix by default, or the parsed JSON response when json_list = TRUE.

embed_openrouter	<i>Request OpenRouter Embeddings</i>
------------------	--------------------------------------

Description

Calls the OpenRouter embeddings endpoint for one or more text inputs.

Usage

```
embed_openrouter(  
  input,  
  model = "openai/text-embedding-3-small",  
  dimensions = NULL,  
  encoding_format = c("float", "base64"),  
  api_key = Sys.getenv("OPENROUTER_API_KEY"),  
  url = "https://openrouter.ai/api/v1/embeddings",  
  json_list = FALSE  
)
```

Arguments

input	A non-empty character vector.
model	Model identifier.
dimensions	Optional embedding dimensionality override.
encoding_format	Embedding encoding. Defaults to "float".
api_key	OpenRouter API key. Defaults to <code>Sys.getenv("OPENROUTER_API_KEY")</code> .
url	OpenRouter embeddings endpoint.
json_list	If TRUE, return the parsed JSON response as a list.

Value

A numeric matrix by default when `encoding_format = "float"`, or the parsed JSON response when `json_list = TRUE`.

extract_openrouter_benchmarks

Extract Benchmark Scores from OpenRouter Model Metadata

Description

Scans OpenRouter model metadata for benchmark payloads, including possible Artificial Analysis benchmark fields when they are present in the models response.

Usage

```
extract_openrouter_benchmarks(
  models,
  benchmark_fields = c("benchmarks", "benchmark_scores", "artificial_analysis",
    "artificial_analysis_benchmarks", "benchmark_data")
)
```

Arguments

models	Parsed JSON from <code>list_openrouter_models(json_list = TRUE)</code> , a list of OpenRouter model objects, or a <code>data.frame</code> returned by <code>list_openrouter_models()</code> .
benchmark_fields	Candidate field names to inspect on each model.

Value

A `data.table` with one row per extracted benchmark metric. Returns an empty `data.table` when no benchmark fields are found.

generate_image_gemini *Generate Images with Gemini*

Description

Calls Gemini image-generation capable models and returns the generated image payloads as base64 strings.

Usage

```
generate_image_gemini(  
  prompt,  
  api_key = Sys.getenv("GEMINI_API_KEY"),  
  model = "gemini-2.5-flash-image",  
  url0 = Sys.getenv("GEMINI_API_URL", unset =  
    "https://generativelanguage.googleapis.com/v1beta/models"),  
  temperature = 0.7,  
  top_p = 1,  
  top_k = 40,  
  max_tokens = NULL,  
  response_modalities = c("TEXT", "IMAGE"),  
  json_list = FALSE  
)
```

Arguments

prompt	A non-empty character string.
api_key	Gemini API key. Defaults to Sys.getenv("GEMINI_API_KEY").
model	Image-capable Gemini model identifier.
url0	Base Gemini models URL.
temperature	Sampling temperature.
top_p	Nucleus sampling parameter.
top_k	Top-k sampling parameter.
max_tokens	Optional maximum number of output tokens.
response_modalities	Modalities to request. Defaults to c("TEXT", "IMAGE").
json_list	If TRUE, return the parsed JSON response as a list.

Value

A character vector of base64-encoded image payloads by default, or the parsed JSON response when json_list = TRUE.

`generate_image_openrouter`*Generate Images with OpenRouter*

Description

Calls the OpenRouter chat completions API for image-capable models and returns the first available image payload or URL.

Usage

```
generate_image_openrouter(  
    prompt,  
    model = "google/gemini-2.5-flash-image-preview",  
    temperature = 0,  
    top_p = 1,  
    max_tokens = 2048L,  
    api_key = Sys.getenv("OPENROUTER_API_KEY"),  
    url = Sys.getenv("OPENROUTER_API_URL", unset =  
        "https://openrouter.ai/api/v1/chat/completions"),  
    json_list = FALSE  
)
```

Arguments

<code>prompt</code>	A non-empty character string.
<code>model</code>	Model identifier.
<code>temperature</code>	Sampling temperature.
<code>top_p</code>	Nucleus sampling parameter.
<code>max_tokens</code>	Maximum number of output tokens.
<code>api_key</code>	OpenRouter API key. Defaults to <code>Sys.getenv("OPENROUTER_API_KEY")</code> .
<code>url</code>	OpenRouter chat completions endpoint.
<code>json_list</code>	If TRUE, return the parsed JSON response as a list.

Value

A character string containing an image URL or base64 payload by default, or the parsed JSON response when `json_list = TRUE`.

list_cerebras_models	<i>List Cerebras Models</i>
----------------------	-----------------------------

Description

Retrieves available models from the public Cerebras models endpoint.

Usage

```
list_cerebras_models(  
  url = "https://api.cerebras.ai/public/v1/models",  
  json_list = FALSE  
)
```

Arguments

- url Cerebras public models endpoint.
- json_list If TRUE, return the parsed JSON response as a list.

Value

A data.table by default, or a parsed JSON list when json_list = TRUE.

list_gemini_models	<i>List Gemini Models</i>
--------------------	---------------------------

Description

Retrieves available models from the Gemini models endpoint.

Usage

```
list_gemini_models(  
  api_key = Sys.getenv("GEMINI_API_KEY"),  
  url = "https://generativelanguage.googleapis.com/v1beta/models",  
  json_list = FALSE  
)
```

Arguments

- api_key Gemini API key. Defaults to Sys.getenv("GEMINI_API_KEY").
- url Gemini models endpoint.
- json_list If TRUE, return the parsed JSON response as a list.

Value

A data.table by default, or a parsed JSON list when json_list = TRUE.

list_groq_models	List Groq Models
------------------	------------------

Description

Retrieves available models from the Groq models endpoint.

Usage

```
list_groq_models(  
  api_key = Sys.getenv("GROQ_API_KEY"),  
  url = "https://api.groq.com/openai/v1/models",  
  json_list = FALSE  
)
```

Arguments

- api_key Groq API key. Defaults to Sys.getenv("GROQ_API_KEY").
- url Groq models endpoint.
- json_list If TRUE, return the parsed JSON response as a list.

Value

A data.table by default, or a parsed JSON list when json_list = TRUE.

list_ollama_models	List Ollama Cloud Models
--------------------	--------------------------

Description

Retrieves available models from the Ollama Cloud tags endpoint.

Usage

```
list_ollama_models(  
  api_key = Sys.getenv("OLLAMA_API_KEY"),  
  url = "https://ollama.com/api/tags",  
  json_list = FALSE  
)
```

Arguments

- api_key Ollama API key. Defaults to Sys.getenv("OLLAMA_API_KEY").
- url Ollama Cloud tags endpoint.
- json_list If TRUE, return the parsed JSON response as a list.

Value

A data.table by default, or a parsed JSON list when json_list = TRUE.

list_openai_models	<i>List OpenAI Models</i>
--------------------	---------------------------

Description

Retrieves models available to the current OpenAI API key.

Usage

```
list_openai_models(  
  api_key = Sys.getenv("OPENAI_API_KEY"),  
  url = "https://api.openai.com/v1/models",  
  json_list = FALSE  
)
```

Arguments

api_key	OpenAI API key. Defaults to Sys.getenv("OPENAI_API_KEY").
url	OpenAI models endpoint.
json_list	If TRUE, return the complete parsed JSON response as a list.

Value

A data.table by default, or the complete parsed JSON response when json_list = TRUE.

list_openrouter_audio_models	<i>List OpenRouter Audio Models</i>
------------------------------	-------------------------------------

Description

Filters the general OpenRouter models catalog to models with audio input or output modalities.

Usage

```
list_openrouter_audio_models(  
  api_key = Sys.getenv("OPENROUTER_API_KEY"),  
  url = "https://openrouter.ai/api/v1/models",  
  json_list = FALSE  
)
```

Arguments

api_key	OpenRouter API key. Defaults to <code>Sys.getenv("OPENROUTER_API_KEY")</code> .
url	OpenRouter models endpoint.
json_list	If TRUE, return the filtered parsed JSON response as a list.

Value

A filtered `data.table` by default, or a filtered parsed JSON list when `json_list = TRUE`.

list_openrouter_benchmarks

List OpenRouter Benchmarks

Description

Retrieves benchmark results directly from OpenRouter's unified benchmarks endpoint. Unlike [extract_openrouter_benchmarks\(\)](#), this function does not inspect model metadata and is intended for benchmark refresh workflows.

Usage

```
list_openrouter_benchmarks(
  api_key = Sys.getenv("OPENROUTER_API_KEY"),
  source = NULL,
  task_type = NULL,
  url = "https://openrouter.ai/api/v1/benchmarks"
)
```

Arguments

api_key	OpenRouter API key. Defaults to <code>Sys.getenv("OPENROUTER_API_KEY")</code> .
source	Optional benchmark source: "artificial-analysis", "design-arena", or "openrouter".
task_type	Optional workload filter: "intelligence", "coding", or "agentic".
url	OpenRouter benchmarks endpoint.

Value

A long `data.table` with one row per model and benchmark metric. The returned scores retain the source-native scale. Provider-supplied source timestamps and URLs, plus the queried endpoint, are retained for provenance. `price_basis` identifies provider-reported per-token pricing and is distinct from `avg_cost_per_task`.

`list_openrouter_embedding_models`*List OpenRouter Embedding Models*

Description

Filters the general OpenRouter models catalog to models with embedding output modalities.

Usage

```
list_openrouter_embedding_models(  
    api_key = Sys.getenv("OPENROUTER_API_KEY"),  
    url = "https://openrouter.ai/api/v1/models",  
    json_list = FALSE  
)
```

Arguments

<code>api_key</code>	OpenRouter API key. Defaults to <code>Sys.getenv("OPENROUTER_API_KEY")</code> .
<code>url</code>	OpenRouter models endpoint.
<code>json_list</code>	If TRUE, return the filtered parsed JSON response as a list.

Value

A filtered `data.table` by default, or a filtered parsed JSON list when `json_list = TRUE`.

`list_openrouter_image_models`*List OpenRouter Image Models*

Description

Filters the general OpenRouter models catalog to models with image output modalities.

Usage

```
list_openrouter_image_models(  
    api_key = Sys.getenv("OPENROUTER_API_KEY"),  
    url = "https://openrouter.ai/api/v1/models",  
    json_list = FALSE  
)
```

Arguments

<code>api_key</code>	OpenRouter API key. Defaults to <code>Sys.getenv("OPENROUTER_API_KEY")</code> .
<code>url</code>	OpenRouter models endpoint.
<code>json_list</code>	If TRUE, return the filtered parsed JSON response as a list.

Value

A filtered `data.table` by default, or a filtered parsed JSON list when `json_list = TRUE`.

<code>list_openrouter_models</code>
<i>List OpenRouter Models</i>

Description

Retrieves available models from the OpenRouter models endpoint.

Usage

```
list_openrouter_models(  
  api_key = Sys.getenv("OPENROUTER_API_KEY"),  
  url = "https://openrouter.ai/api/v1/models",  
  json_list = FALSE  
)
```

Arguments

<code>api_key</code>	OpenRouter API key. Defaults to <code>Sys.getenv("OPENROUTER_API_KEY")</code> .
<code>url</code>	OpenRouter models endpoint.
<code>json_list</code>	If TRUE, return the parsed JSON response as a list.

Value

A `data.table` by default, or a parsed JSON list when `json_list = TRUE`.

`list_openrouter_multimodal_models`*List OpenRouter Multimodal Models*

Description

Filters the general OpenRouter models catalog to models that support multiple input modalities.

Usage

```
list_openrouter_multimodal_models(  
    api_key = Sys.getenv("OPENROUTER_API_KEY"),  
    url = "https://openrouter.ai/api/v1/models",  
    json_list = FALSE  
)
```

Arguments

<code>api_key</code>	OpenRouter API key. Defaults to <code>Sys.getenv("OPENROUTER_API_KEY")</code> .
<code>url</code>	OpenRouter models endpoint.
<code>json_list</code>	If TRUE, return the filtered parsed JSON response as a list.

Value

A filtered `data.table` by default, or a filtered parsed JSON list when `json_list = TRUE`.

`list_openrouter_video_models`*List OpenRouter Video Generation Models*

Description

Retrieves available video generation models and their normalized feature metadata from the OpenRouter video models endpoint.

Usage

```
list_openrouter_video_models(  
    api_key = Sys.getenv("OPENROUTER_API_KEY"),  
    url = "https://openrouter.ai/api/v1/videos/models",  
    json_list = FALSE  
)
```

Arguments

api_key	OpenRouter API key. Defaults to <code>Sys.getenv("OPENROUTER_API_KEY")</code> .
url	OpenRouter video models endpoint.
json_list	If TRUE, return the parsed JSON response as a list.

Value

A `data.table` by default, or a parsed JSON list when `json_list = TRUE`.

query_cerebras	<i>Query a Cerebras Chat Model</i>
----------------	------------------------------------

Description

Sends a single user prompt to the Cerebras chat completions API.

Usage

```
query_cerebras(  
  prompt,  
  model = "gpt-oss-120b",  
  api_key = Sys.getenv("CEREBRAS_API_KEY"),  
  url = "https://api.cerebras.ai/v1/chat/completions",  
  json_list = FALSE  
)
```

Arguments

prompt	A non-empty character string.
model	Model identifier.
api_key	Cerebras API key. Defaults to <code>Sys.getenv("CEREBRAS_API_KEY")</code> .
url	Cerebras chat completions endpoint.
json_list	If TRUE, return the parsed JSON response as a list.

Value

A character string by default, or a parsed JSON list when `json_list = TRUE`.

See Also

[list_cerebras_models\(\)](#)

query_fallback	<i>Query Providers with Ordered Fallback</i>
----------------	--

Description

Tries the package’s text-query wrappers in order until one succeeds: query_gemini(), then query_openrouter(), then query_groq().

Usage

```
query_fallback(  
  prompt,  
  json_list = FALSE,  
  api_key_gemini = Sys.getenv("GEMINI_API_KEY"),  
  api_key_openrouter = Sys.getenv("OPENROUTER_API_KEY"),  
  api_key_groq = Sys.getenv("GROQ_API_KEY")  
)
```

Arguments

- prompt A non-empty character string.
- json_list If TRUE, return a list containing the successful provider name and parsed response object.
- api_key_gemini Gemini API key. Defaults to Sys.getenv("GEMINI_API_KEY").
- api_key_openrouter OpenRouter API key. Defaults to Sys.getenv("OPENROUTER_API_KEY").
- api_key_groq Groq API key. Defaults to Sys.getenv("GROQ_API_KEY").

Value

A character string by default. When json_list = TRUE, returns a list with elements provider and response.

query_gemini	<i>Query a Gemini Model</i>
--------------	-----------------------------

Description

Sends a single user prompt to the Gemini generateContent API.

Usage

```
query_gemini(
  prompt,
  api_key = Sys.getenv("GEMINI_API_KEY"),
  model = "gemini-2.5-flash",
  url0 = Sys.getenv("GEMINI_API_URL", unset =
    "https://generativelanguage.googleapis.com/v1beta/models"),
  temperature = 0.7,
  top_p = 1,
  top_k = 40,
  max_tokens = NULL,
  response_modalities = NULL,
  speech_config = NULL,
  json_list = FALSE
)
```

Arguments

<code>prompt</code>	A non-empty character string.
<code>api_key</code>	Gemini API key. Defaults to <code>Sys.getenv("GEMINI_API_KEY")</code> .
<code>model</code>	Model identifier.
<code>url0</code>	Base Gemini models URL.
<code>temperature</code>	Sampling temperature.
<code>top_p</code>	Nucleus sampling parameter.
<code>top_k</code>	Top-k sampling parameter.
<code>max_tokens</code>	Optional maximum number of output tokens.
<code>response_modalities</code>	Optional response modalities, for example <code>c("TEXT")</code> or <code>c("AUDIO")</code> .
<code>speech_config</code>	Optional Gemini <code>speechConfig</code> object supplied as an R list.
<code>json_list</code>	If TRUE, return the parsed JSON response as a list.

Value

A character string by default. For audio responses, returns the base64-encoded audio data from `inlineData$data`. Returns the parsed JSON list when `json_list = TRUE`.

<code>query_gemini_content</code>	<i>Query a Gemini Multimodal Model</i>
-----------------------------------	--

Description

Sends Gemini `generateContent` requests using either a simple text prompt or an explicit list of content parts. This allows text, image, audio, PDF, and other supported non-text inputs to be passed through the same wrapper.

Usage

```

query_gemini_content(
  prompt = NULL,
  parts = NULL,
  api_key = Sys.getenv("GEMINI_API_KEY"),
  model = "gemini-2.5-flash",
  url0 = Sys.getenv("GEMINI_API_URL", unset =
    "https://generativelanguage.googleapis.com/v1beta/models/"),
  temperature = 0.7,
  top_p = 1,
  top_k = 40,
  max_tokens = NULL,
  response_modalities = NULL,
  speech_config = NULL,
  json_list = FALSE
)

```

Arguments

<code>prompt</code>	A non-empty character string. Ignored when <code>parts</code> is supplied.
<code>parts</code>	Optional Gemini parts payload supplied as a list. Use this for multimodal inputs such as <code>inlineData</code> or <code>fileData</code> .
<code>api_key</code>	Gemini API key. Defaults to <code>Sys.getenv("GEMINI_API_KEY")</code> .
<code>model</code>	Model identifier.
<code>url0</code>	Base Gemini models URL.
<code>temperature</code>	Sampling temperature.
<code>top_p</code>	Nucleus sampling parameter.
<code>top_k</code>	Top-k sampling parameter.
<code>max_tokens</code>	Optional maximum number of output tokens.
<code>response_modalities</code>	Optional response modalities, for example <code>c("TEXT")</code> , <code>c("AUDIO")</code> , or <code>c("TEXT", "IMAGE")</code> .
<code>speech_config</code>	Optional Gemini <code>speechConfig</code> object supplied as an R list.
<code>json_list</code>	If TRUE, return the parsed JSON response as a list.

Value

A character string when Gemini returns text, a base64 string when the first returned part is inline binary data, or the parsed JSON response when `json_list = TRUE`.

query_groq	<i>Query a Groq Chat Model</i>
------------	--------------------------------

Description

Sends a single user prompt to the Groq chat completions API.

Usage

```
query_groq(  
    prompt,  
    api_key = Sys.getenv("GROQ_API_KEY"),  
    url = Sys.getenv("GROQ_API_URL", unset =  
        "https://api.groq.com/openai/v1/chat/completions"),  
    model = c("groq/compound", "allam-2-7b", "groq/compound-mini", "qwen/qwen3-32b",  
        "openai/gpt-oss-20b", "canopylabs/orpheus-v1-english", "openai/gpt-oss-120b",  
        "whisper-large-v3", "llama-3.3-70b-versatile", "moonshotai/kimi-k2-instruct-0905",  
        "whisper-large-v3-turbo", "meta-llama/llama-prompt-guard-2-86m",  
        "moonshotai/kimi-k2-instruct", "meta-llama/llama-prompt-guard-2-22m",  
        "meta-llama/llama-4-scout-17b-16e-instruct", "openai/gpt-oss-safeguard-20b",  
        "llama-3.1-8b-instant", "canopylabs/orpheus-arabic-saudi"),  
    temperature = 1,  
    top_p = 1,  
    max_tokens = 1024,  
    stream = FALSE,  
    json_list = FALSE  
)
```

Arguments

prompt	A non-empty character string.
api_key	Groq API key. Defaults to Sys.getenv("GROQ_API_KEY").
url	Groq chat completions endpoint.
model	Model identifier.
temperature	Sampling temperature.
top_p	Nucleus sampling parameter.
max_tokens	Maximum number of output tokens.
stream	Whether to request an SSE response. The complete response is buffered and assembled before the function returns; tokens are not printed incrementally to the terminal.
json_list	If TRUE, return the parsed JSON response as a list.

Value

A character string by default. With json_list = TRUE, returns a parsed JSON response for non-streaming requests or a list of parsed SSE chunks for streaming requests.

query_ollama	<i>Query an Ollama Cloud Chat Model</i>
--------------	---

Description

Sends a single user prompt to the Ollama Cloud chat API.

Usage

```
query_ollama(  
    prompt,  
    model = "gpt-oss:120b",  
    api_key = Sys.getenv("OLLAMA_API_KEY"),  
    url = "https://ollama.com/api/chat",  
    json_list = FALSE  
)
```

Arguments

- | | |
|-----------|---|
| prompt | A non-empty character string. |
| model | Model identifier. |
| api_key | Ollama API key. Defaults to Sys.getenv("OLLAMA_API_KEY"). |
| url | Ollama Cloud chat endpoint. |
| json_list | If TRUE, return the parsed JSON response as a list. |

Value

A character string by default, or a parsed JSON list when json_list = TRUE.

query_openai	<i>Query an OpenAI Model with the Responses API</i>
--------------	---

Description

Sends a request directly to OpenAI’s Responses API. The input may be a simple text prompt or a structured Responses API input list for multimodal and multi-message workflows.

Usage

```
query_openai(  
  input,  
  model = "gpt-5.6",  
  instructions = NULL,  
  max_output_tokens = NULL,  
  temperature = NULL,  
  top_p = NULL,  
  tools = NULL,  
  api_key = Sys.getenv("OPENAI_API_KEY"),  
  url = Sys.getenv("OPENAI_API_URL", unset = "https://api.openai.com/v1/responses"),  
  json_list = FALSE  
)
```

Arguments

input	A non-empty character string or structured Responses API input list.
model	Model identifier.
instructions	Optional system or developer instructions.
max_output_tokens	Optional maximum number of generated tokens, including visible output and reasoning tokens.
temperature	Optional sampling temperature.
top_p	Optional nucleus sampling parameter.
tools	Optional list of Responses API tool definitions.
api_key	OpenAI API key. Defaults to Sys.getenv("OPENAI_API_KEY").
url	OpenAI Responses endpoint.
json_list	If TRUE, return the complete parsed JSON response as a list.

Value

A character string containing the response text by default, or the complete parsed JSON response when json_list = TRUE.

query_openrouter	<i>Query an OpenRouter Chat Model</i>
------------------	---------------------------------------

Description

Sends a single user prompt to the OpenRouter chat completions API.

Usage

```
query_openrouter(  
  prompt,  
  model = "openrouter/free",  
  temperature = 0,  
  top_p = 1,  
  max_tokens = 2048L,  
  reasoning = TRUE,  
  logprobs = FALSE,  
  top_logprobs = NULL,  
  api_key = Sys.getenv("OPENROUTER_API_KEY"),  
  url = Sys.getenv("OPENROUTER_API_URL", unset =  
    "https://openrouter.ai/api/v1/chat/completions"),  
  json_list = FALSE  
)
```

Arguments

prompt	A non-empty character string.
model	Model identifier.
temperature	Sampling temperature.
top_p	Nucleus sampling parameter.
max_tokens	Maximum number of output tokens.
reasoning	Whether to enable reasoning mode.
logprobs	Whether to return log probabilities for output tokens.
top_logprobs	Optional number of most likely tokens to return at each position. Must be an integer from 0 through 20 and requires logprobs = TRUE.
api_key	OpenRouter API key. Defaults to Sys.getenv("OPENROUTER_API_KEY").
url	OpenRouter chat completions endpoint.
json_list	If TRUE, return the parsed JSON response as a list.

Value

A character string by default, or a parsed JSON list when json_list = TRUE.

query_openrouter_content
<i>Query an OpenRouter Multimodal Model</i>

Description

Sends a single OpenRouter chat-completions request using either simple text content or an explicit multimodal content block list.

Usage

```

query_openrouter_content(
    content,
    model = "openrouter/free",
    temperature = 0,
    top_p = 1,
    max_tokens = 2048L,
    reasoning = TRUE,
    logprobs = FALSE,
    top_logprobs = NULL,
    modalities = NULL,
    api_key = Sys.getenv("OPENROUTER_API_KEY"),
    url = Sys.getenv("OPENROUTER_API_URL", unset =
        "https://openrouter.ai/api/v1/chat/completions"),
    json_list = FALSE
)

```

Arguments

content	A non-empty character string or a non-empty list of OpenAI style content blocks.
model	Model identifier.
temperature	Sampling temperature.
top_p	Nucleus sampling parameter.
max_tokens	Maximum number of output tokens.
reasoning	Whether to enable reasoning mode.
logprobs	Whether to return log probabilities for output tokens.
top_logprobs	Optional number of most likely tokens to return at each position. Must be an integer from 0 through 20 and requires logprobs = TRUE.
modalities	Optional output modalities, for example "text" or c("text", "image").
api_key	OpenRouter API key. Defaults to Sys.getenv("OPENROUTER_API_KEY").
url	OpenRouter chat completions endpoint.
json_list	If TRUE, return the parsed JSON response as a list.

Value

A character string by default when the assistant returns text, the first image payload or URL when text is absent but images are returned, or the parsed JSON response when json_list = TRUE.

shell-scripts	<i>Shell Script Helpers</i>
---------------	-----------------------------

Description

inferencer also ships a small optional shell layer under `inst/shell`. These scripts are executable zsh command-line helpers that mirror common R wrapper workflows: `query_gemini`, `query_groq`, `query_openrouter`, `query_ollama`, `query_fallback`, provider-specific `list_*_models` scripts, and `render_markdown_terminal`.

Details

The shell scripts read API keys and defaults from shell environment variables, typically configured in `.zprofile`. The variable names are aligned with the R wrappers, such as `GEMINI_API_KEY`, `GROQ_API_KEY`, `OPENROUTER_API_KEY`, and `OLLAMA_API_KEY`.

They are intentionally minimal and depend on external `curl` and `jq` binaries. Query scripts print the model response text by default and return the full parsed JSON payload when called with `--json`. Pipe query output to `render_markdown_terminal` when terminal markdown rendering is desired. `query_fallback` tries `query_gemini`, then `query_openrouter`, then `query_groq` using each script's default model.

<code>write_gemini_audio</code>	<i>Write Gemini Audio to a File</i>
---------------------------------	-------------------------------------

Description

Decodes base64 audio returned by `query_gemini()` and writes it as raw PCM or a WAV file.

Usage

```
write_gemini_audio(  
  x,  
  path,  
  format = c("pcm", "wav"),  
  sample_rate = 24000L,  
  channels = 1L,  
  bits_per_sample = 16L  
)
```

Arguments

- | | |
|-------------------|---|
| <code>x</code> | A base64-encoded audio string, or a parsed JSON response returned by <code>query_gemini(..., json_list = TRUE)</code> . |
| <code>path</code> | Output file path. |

<code>format</code>	Output format: "pcm" or "wav".
<code>sample_rate</code>	Sample rate used when writing WAV output.
<code>channels</code>	Number of audio channels used when writing WAV output.
<code>bits_per_sample</code>	Bit depth used when writing WAV output.

Value

Invisibly returns path.

Index

`embed_gemini`, [2](#)
`embed_openrouter`, [3](#)
`extract_openrouter_benchmarks`, [4](#)
`extract_openrouter_benchmarks()`, [10](#)

`generate_image_gemini`, [5](#)
`generate_image_openrouter`, [6](#)

`list_cerebras_models`, [7](#)
`list_cerebras_models()`, [14](#)
`list_gemini_models`, [7](#)
`list_groq_models`, [8](#)
`list_ollama_models`, [8](#)
`list_openai_models`, [9](#)
`list_openrouter_audio_models`, [9](#)
`list_openrouter_benchmarks`, [10](#)
`list_openrouter_embedding_models`, [11](#)
`list_openrouter_image_models`, [11](#)
`list_openrouter_models`, [12](#)
`list_openrouter_multimodal_models`, [13](#)
`list_openrouter_video_models`, [13](#)

`query_cerebras`, [14](#)
`query_fallback`, [15](#)
`query_gemini`, [15](#)
`query_gemini_content`, [16](#)
`query_groq`, [18](#)
`query_ollama`, [19](#)
`query_openai`, [19](#)
`query_openrouter`, [20](#)
`query_openrouter_content`, [21](#)

`shell-scripts`, [23](#)

`write_gemini_audio`, [23](#)