

Package: nrouter (via r-universe)

October 5, 2026

Type Package

Title OpenAI-Compatible SDK for the nRouter LLM Gateway

Version 3.0.0

Description A client for the nRouter LLM gateway ([\(<https://nrouter.ai>](https://nrouter.ai)), an OpenAI-compatible API giving one key access to models across six provider clouds. No official OpenAI SDK exists for R, so this package calls the HTTP API directly via 'httr'. Gateway refusals are raised as classed conditions, and per-request cost and token metadata are returned alongside every response.

License MIT + file LICENSE

URL <https://nrouter.ai>, <https://github.com/nRouterAI/nrouter-sdk>

BugReports <https://github.com/nRouterAI/nrouter-sdk/issues>

Encoding UTF-8

Imports curl, httr, jsonlite

Suggests testthat (>= 3.0.0), webfakes

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

Config/pak/sysreqs libssl-dev

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

Date/Publication 2026-09-05 06:04:48 UTC

RemoteUrl <https://github.com/nRouterAI/nrouter-sdk>

RemoteRef HEAD

RemoteSha 5b62e4b978d59815b853ee34c8b6304007be5502

RemoteSubdir sdks/r

Contents

nrouter-errors	3
nrouter_audio_speech	3
nrouter_audio_transcriptions	4
nrouter_audio_translations	4
nrouter_bytes	5
nrouter_chat	5
nrouter_chat_completions	6
nrouter_chat_completions_stream	7
nrouter_client	7
nrouter_completions	8
nrouter_completions_stream	9
nrouter_condition	9
nrouter_configuration_condition	10
nrouter_count_tokens	10
nrouter_create_video	11
nrouter_default_base_url	11
nrouter_default_connect_timeout_seconds	12
nrouter_default_stream_idle_seconds	12
nrouter_default_timeout_seconds	13
nrouter_download_video_content	13
nrouter_embeddings	14
nrouter_env_key	14
nrouter_error_from_payload	15
nrouter_header_names	15
nrouter_images_generations	16
nrouter_is_priced	16
nrouter_is_retryable	17
nrouter_key_prefix	17
nrouter_messages	18
nrouter_messages_stream	18
nrouter_meta	19
nrouter_model	19
nrouter_models	20
nrouter_multipart	20
nrouter_request	21
nrouter_resolve_api_key	21
nrouter_responses	22
nrouter_responses_stream	22
nrouter_retrieve_video	23
nrouter_sse_parser	23
nrouter_stream	24
nrouter_transport_condition	24
print.nrouter_client	25

nrouter-errors	<i>nRouter error conditions</i>
----------------	---------------------------------

Description

Every gateway refusal is raised as a classed R condition. The class vector runs specific-to-general — e.g. `c("nrouter_rate_limit_error", "nrouter_error", "error", "condition")` — so `tryCatch(nrouter_error = ...)` catches everything while `tryCatch(nrouter_rate_limit_error = ...)` catches only that.

Details

The gateway's stable code decides the class, not the HTTP status: status alone cannot separate `invalid_request` from `guardrail_blocked` (both 400), nor `rate_limit_exceeded` from `tpm_limit_exceeded` (both 429).

nrouter_audio_speech	<i>POST /audio/speech — generated audio bytes plus response metadata</i>
----------------------	--

Description

POST /audio/speech — generated audio bytes plus response metadata

Usage

```
nrouter_audio_speech(client, body)
```

Arguments

client	An <code>nrouter_client</code> .
body	Named list forming the request body.

Value

A list with `bytes`, `meta` and `status_code`.

`nrouter_audio_transcriptions`*POST /audio/transcriptions - Whisper-style speech to text*

Description

POST /audio/transcriptions - Whisper-style speech to text

Usage

```
nrouter_audio_transcriptions(client, file_path, fields = list())
```

Arguments

<code>client</code>	An <code>nrouter_client</code> .
<code>file_path</code>	Path to the file to upload. Its EXTENSION is load-bearing — the upstream providers pick their decoder from it.
<code>fields</code>	Named list of additional form fields, e.g. <code>model</code> .

Value

A list with `body`, `meta` and `status_code`.

`nrouter_audio_translations`*POST /audio/translations - speech in any language to English text*

Description

POST /audio/translations - speech in any language to English text

Usage

```
nrouter_audio_translations(client, file_path, fields = list())
```

Arguments

<code>client</code>	An <code>nrouter_client</code> .
<code>file_path</code>	Path to the file to upload. Its EXTENSION is load-bearing — the upstream providers pick their decoder from it.
<code>fields</code>	Named list of additional form fields, e.g. <code>model</code> .

Value

A list with `body`, `meta` and `status_code`.

nrouter_bytes	<i>Raw bytes plus metadata, for the endpoints that do not return JSON</i>
---------------	---

Description

/v1/audio/speech returns audio, /v1/videos/{id}/content returns a video, and stream = TRUE returns SSE. The JSON helpers refuse those rather than handing back an empty body for a request you were billed for; this is the function that returns them.

Usage

```
nrouter_bytes(client, path, body = NULL)
```

Arguments

client	An nrouter_client.
path	Path under the gateway's /v1 root.
body	Named list to send as JSON, or NULL for a GET.

Value

A list with bytes, meta and status_code.

nrouter_chat	<i>Send a chat completion request (convenience wrapper)</i>
--------------	---

Description

Kept as the package's original one-call entry point. Prefer nrouter_chat_completions() when you need the response metadata.

Usage

```
nrouter_chat(  
  messages,  
  model = "gpt-5.4-mini",  
  api_key = NULL,  
  base_url = nrouter_default_base_url()  
)
```

Arguments

messages	List of message objects, e.g. <code>list(list(role = "user", content = "Hello!"))</code> .
model	Model name. The default MUST be a model the gateway serves on <code>/v1/chat/completions</code> , because that is the only wire this wrapper posts to. The gateway resolves a provider endpoint PER WIRE and answers 404 <code>model_unavailable_on_route</code> when the provider declares none, so an Anthropic id (Messages-only) was a 404 out of the box for anyone calling <code>nrouter_chat(messages)</code> without naming a model. Pinned by <code>tests/testthat/test-contract.R</code> and <code>conformance/source_defaults.py</code> .
api_key	nRouter API key. Defaults to <code>NROUTER_API_KEY</code> .
base_url	Gateway base URL.

Value

The parsed JSON response as an R list.

```
nrouter_chat_completions
      POST /chat/completions
```

Description

POST /chat/completions

Usage

```
nrouter_chat_completions(client, body)
```

Arguments

client	An <code>nrouter_client</code> .
body	Named list forming the request body.

Value

A list with `body`, `meta` and `status_code`.

nrouter_chat_completions_stream
Stream POST /chat/completions

Description

Stream POST /chat/completions

Usage

```
nrouter_chat_completions_stream(client, body, on_chunk)
```

Arguments

client	An nrouter_client.
body	Named list forming the request body. stream = TRUE is set in a copy and the caller's list is not mutated.
on_chunk	Function called incrementally with event, delta, and provider-native data. Return FALSE to cancel early.

nrouter_client *Create an nRouter client*

Description

The gateway speaks the OpenAI wire format, so request and response bodies are the shapes you already know. There is no official OpenAI SDK for R, so this package calls the HTTP API directly via **httr**.

Usage

```
nrouter_client(  
  api_key = NULL,  
  base_url = nrouter_default_base_url(),  
  timeout_seconds = nrouter_default_timeout_seconds(),  
  connect_timeout_seconds = nrouter_default_connect_timeout_seconds(),  
  stream_idle_seconds = nrouter_default_stream_idle_seconds()  
)
```

Arguments

api_key nRouter API key. Defaults to NROUTER_API_KEY.
base_url Gateway base URL.
timeout_seconds Whole-request ceiling for buffered calls. See [nrouter_default_timeout_seconds](#).
connect_timeout_seconds Connect-phase ceiling. See [nrouter_default_connect_timeout_seconds](#).
stream_idle_seconds Stall ceiling for streaming and binary transfers, which carry no whole-request ceiling. See [nrouter_default_stream_idle_seconds](#).

Value

An object of class `nrouter_client`.

Examples

```
## Not run:
client <- nrouter_client()
result <- nrouter_chat_completions(client, list(
  model = "gpt-5.4-mini",
  messages = list(list(role = "user", content = "Hello!"))
))
# Unpriced is unknown, not free. Never render a NULL cost as 0.
print(result$meta)

## End(Not run)
```

nrouter_completions *POST /completions — the legacy text-completions wire*

Description

POST /completions — the legacy text-completions wire

Usage

```
nrouter_completions(client, body)
```

Arguments

client An `nrouter_client`.
body Named list forming the request body.

Value

A list with `body`, `meta` and `status_code`.

```
nrouter_completions_stream
    Stream POST /completions
```

Description

Stream POST /completions

Usage

```
nrouter_completions_stream(client, body, on_chunk)
```

Arguments

client	An nrouter_client.
body	Named list forming the request body. stream = TRUE is set in a copy and the caller's list is not mutated.
on_chunk	Function called incrementally with event, delta, and provider-native data. Return FALSE to cancel early.

```
nrouter_condition    Build (but do not signal) an nRouter condition
```

Description

Build (but do not signal) an nRouter condition

Usage

```
nrouter_condition(
  message,
  code = NULL,
  status = NULL,
  request_id = NULL,
  limit_source = NULL,
  auth_reason = NULL
)
```

Arguments

message	Human-readable message.
code	The gateway's stable error code, or NULL.
status	HTTP status, or NULL.
request_id	Value of x-nr-request-id, or NULL.
limit_source	Value of x-nr-limit-source, or NULL.
auth_reason	Value of x-nr-auth-reason, or NULL.

Value

A condition object.

nrouter_configuration_condition	<i>A configuration failure: the SDK refused before sending anything</i>
---------------------------------	---

Description

Separate from nrouter_transport_condition on purpose. Both are raised locally, but this one is PERMANENT — a caller retrying on nrouter_is_retryable would spin forever without making a request.

Usage

nrouter_configuration_condition(message)

Arguments

message Human-readable message.

Value

A condition object.

nrouter_count_tokens	<i>POST /messages/count_tokens — counts input without generating</i>
----------------------	--

Description

POST /messages/count_tokens — counts input without generating

Usage

nrouter_count_tokens(client, body)

Arguments

client An nrouter_client.
body Named list forming the request body.

Value

A list with body, meta and status_code.

nrouter_create_video *POST /videos — starts a video generation job*

Description

POST /videos — starts a video generation job

Usage

```
nrouter_create_video(client, body)
```

Arguments

client	An nrouter_client.
body	Named list forming the request body.

Value

A list with body, meta and status_code.

nrouter_default_base_url
The gateway's customer surface

Description

A dynamic value: override it for stage.

Usage

```
nrouter_default_base_url()
```

Value

The default base URL.

nrouter_default_connect_timeout_seconds
Connect-phase ceiling, in seconds

Description

Bounds DNS resolution plus the TCP handshake and nothing after it, so a black-holed gateway address is reported rather than waited on. Matches the gateway's own 10 s provider connect timeout.

Usage

nrouter_default_connect_timeout_seconds()

Value

The default connect timeout in seconds.

nrouter_default_stream_idle_seconds
Stall ceiling for streaming and binary transfers, in seconds

Description

nrouter_stream() and nrouter_bytes() deliberately set NO whole-request timeout: a whole-request ceiling severs an SSE stream mid-generation and truncates a long /v1/videos/{id}/content download, both of them already billed. They bound the transfer with libcurl's low-speed limit instead, so a transfer that STOPS producing bytes for this long is abandoned while a working one runs as long as it needs. 180 s sits above the gateway's own 120 s between-bytes read timeout, so the gateway's honest error reaches the caller instead of a client-side abort racing it.

Usage

nrouter_default_stream_idle_seconds()

Value

The default stall timeout in seconds.

<code>nrouter_default_timeout_seconds</code>
<i>Whole-request ceiling for buffered calls, in seconds</i>

Description

Applied to `nrouter_request()` and `nrouter_multipart()` as `http::timeout()`. 600 s is above the gateway's ~410 s worst honest case with margin, and the same order as the OpenAI and Anthropic clients' own 600 s defaults.

Usage

`nrouter_default_timeout_seconds()`

Value

The default request timeout in seconds.

<code>nrouter_download_video_content</code>
<i>GET /videos/(id)/content — downloads generated video bytes</i>

Description

GET /videos/(id)/content — downloads generated video bytes

Usage

`nrouter_download_video_content(client, video_id)`

Arguments

- `client` An `nrouter_client`.
- `video_id` Video job identifier. It is encoded as one URL path segment.

Value

A list with bytes, meta and status_code.

nrouter_embeddings	<i>POST /embeddings</i>
--------------------	-------------------------

Description

POST /embeddings

Usage

```
nrouter_embeddings(client, body)
```

Arguments

client	An nrouter_client.
body	Named list forming the request body.

Value

A list with body, meta and status_code.

nrouter_env_key	<i>The one environment variable this package reads</i>
-----------------	--

Description

The one environment variable this package reads

Usage

```
nrouter_env_key()
```

Value

The variable name.

`nrouter_error_from_payload`*Pull the gateway's code and message out of an error payload*

Description

The gateway nests them under `error`; a bare object is accepted too, so a proxy that reshapes the envelope cannot downgrade a typed error into a generic one.

Usage

```
nrouter_error_from_payload(status, payload, meta)
```

Arguments

<code>status</code>	HTTP status.
<code>payload</code>	Parsed response body.
<code>meta</code>	An <code>nrouter_meta</code> object.

Value

A condition object.

`nrouter_header_names` *Every response header this SDK reads*

Description

Exactly the names in `spec/nrouter-sdk-spec.json`.

Usage

```
nrouter_header_names()
```

Value

A character vector of 15 header names.

nrouter_images_generations	<i>POST /images/generations</i>
----------------------------	---------------------------------

Description

POST /images/generations

Usage

nrouter_images_generations(client, body)

Arguments

- | | |
|--------|--------------------------------------|
| client | An nrouter_client. |
| body | Named list forming the request body. |

Value

A list with body, meta and status_code.

nrouter_is_priced	<i>Did the gateway price this request exactly?</i>
-------------------	--

Description

Did the gateway price this request exactly?

Usage

nrouter_is_priced(meta)

Arguments

- | | |
|------|-------------------------|
| meta | An nrouter_meta object. |
|------|-------------------------|

Value

TRUE only when the cost is both present and exact.

nrouter_is_retryable	<i>Is this nRouter condition worth retrying?</i>
----------------------	--

Description

Is this nRouter condition worth retrying?

Usage

```
nrouter_is_retryable(cond)
```

Arguments

cond	A condition raised by this package.
------	-------------------------------------

Value

TRUE when retrying the identical request could succeed.

nrouter_key_prefix	<i>The prefix every nRouter customer key carries</i>
--------------------	--

Description

The prefix every nRouter customer key carries

Usage

```
nrouter_key_prefix()
```

Value

The key prefix.

nrouter_messages	<i>POST /messages (the Anthropic wire format the gateway also serves)</i>
------------------	---

Description

POST /messages (the Anthropic wire format the gateway also serves)

Usage

nrouter_messages(client, body)

Arguments

- | | |
|--------|--------------------------------------|
| client | An nrouter_client. |
| body | Named list forming the request body. |

Value

A list with body, meta and status_code.

nrouter_messages_stream	<i>Stream POST /messages</i>
-------------------------	------------------------------

Description

Stream POST /messages

Usage

nrouter_messages_stream(client, body, on_chunk)

Arguments

- | | |
|----------|---|
| client | An nrouter_client. |
| body | Named list forming the request body. stream = TRUE is set in a copy and the caller's list is not mutated. |
| on_chunk | Function called incrementally with event, delta, and provider-native data. Return FALSE to cancel early. |

nrouter_meta	<i>Per-request metadata from the nRouter response headers</i>
--------------	---

Description

Every field is NULL when the gateway did not send it. The gateway omits a header rather than sending a placeholder, and the two omissions that matter most are `x-nr-request-cost` — ABSENT when the model is unpriced, never `0` — and `x-nr-limit-source`, absent when nothing measured a refusal.

Usage

```
nrouter_meta(headers = list())
```

Arguments

`headers` Named character vector or list of response headers, keyed by lowercase name.

Value

An object of class `nrouter_meta`.

nrouter_model	<i>GET /models/(model_id)</i>
---------------	-------------------------------

Description

GET /models/(model_id)

Usage

```
nrouter_model(client, model_id)
```

Arguments

`client` An `nrouter_client`.
`model_id` Model identifier. It is encoded as one URL path segment.

Value

A list with `body`, `meta` and `status_code`.

nrouter_models	<i>GET /models — what this key is allowed to route to</i>
----------------	---

Description

GET /models — what this key is allowed to route to

Usage

```
nrouter_models(client)
```

Arguments

client	An nrouter_client.
--------	--------------------

Value

A list with body, meta and status_code.

nrouter_multipart	<i>Send a multipart request to the nRouter gateway</i>
-------------------	--

Description

multipart/form-data, not JSON: /v1/audio/transcriptions and /v1/audio/translations require a binary file part, so the JSON helpers cannot reach those endpoints at all.

Usage

```
nrouter_multipart(client, path, file_path, fields = list())
```

Arguments

client	An nrouter_client.
path	Path under the gateway's /v1 root.
file_path	Path to the file to upload. Its EXTENSION is load-bearing — the upstream providers pick their decoder from it.
fields	Named list of additional form fields, e.g. model.

Value

A list with body, meta and status_code.

nrouter_request	<i>Send a request to the nRouter gateway</i>
-----------------	--

Description

Send a request to the nRouter gateway

Usage

```
nrouter_request(  
    client,  
    path,  
    body = NULL,  
    method = if (is.null(body)) "GET" else "POST"  
)
```

Arguments

client	An nrouter_client.
path	Path under the gateway's /v1 root, e.g. "/models".
body	Named list to send as JSON, or NULL for a GET.
method	HTTP method; inferred from body when not given.

Value

A list with body, meta and status_code.

nrouter_resolve_api_key	<i>Resolve and validate an nRouter API key</i>
-------------------------	--

Description

Explicit argument first, then NROUTER_API_KEY. Validation happens before any request so a malformed key fails here rather than as a 401 that reads like a revoked credential.

Usage

```
nrouter_resolve_api_key(api_key = NULL)
```

Arguments

api_key	Explicit key, or NULL to read the environment.
---------	--

Value

The validated key.

nrouter_responses	<i>POST /responses</i>
-------------------	------------------------

Description

POST /responses

Usage

nrouter_responses(client, body)

Arguments

- | | |
|--------|--------------------------------------|
| client | An nrouter_client. |
| body | Named list forming the request body. |

Value

A list with body, meta and status_code.

nrouter_responses_stream	<i>Stream POST /responses</i>
--------------------------	-------------------------------

Description

Stream POST /responses

Usage

nrouter_responses_stream(client, body, on_chunk)

Arguments

- | | |
|----------|---|
| client | An nrouter_client. |
| body | Named list forming the request body. stream = TRUE is set in a copy and the caller's list is not mutated. |
| on_chunk | Function called incrementally with event, delta, and provider-native data. Return FALSE to cancel early. |

`nrouter_retrieve_video`*GET /videos/(id) — polls a video generation job*

Description

GET /videos/(id) — polls a video generation job

Usage

```
nrouter_retrieve_video(client, video_id)
```

Arguments

<code>client</code>	An <code>nrouter_client</code> .
<code>video_id</code>	Video job identifier. It is encoded as one URL path segment.

Value

A list with `body`, `meta` and `status_code`.

`nrouter_sse_parser` *Build an incremental server-sent event parser*

Description

The parser is public for advanced transports and deterministic testing. Most callers should use one of the named streaming helpers.

Usage

```
nrouter_sse_parser(on_chunk)
```

Arguments

<code>on_chunk</code>	Function called with each decoded event. Return <code>FALSE</code> to cancel the request early.
-----------------------	---

Value

A list with `feed(raw)` and `finish()` functions.

nrouter_stream	<i>Stream a JSON request from the nRouter gateway</i>
----------------	---

Description

Stream a JSON request from the nRouter gateway

Usage

```
nrouter_stream(client, path, body, on_chunk)
```

Arguments

client	An nrouter_client.
path	Path under the gateway's /v1 root.
body	Named list forming the request body. stream = TRUE is set in a copy and the caller's list is not mutated.
on_chunk	Function called incrementally with event, delta, and provider-native data. Return FALSE to cancel early.

Value

A list with final meta, status_code, and cancelled.

nrouter_transport_condition	<i>A transport failure: the request left this process and got no answer</i>
-----------------------------	---

Description

A transport failure: the request left this process and got no answer

Usage

```
nrouter_transport_condition(message)
```

Arguments

message	Human-readable message.
---------	-------------------------

Value

A condition object.

print.nrouter_client	<i>Print an nRouter client without disclosing its key</i>
----------------------	---

Description

R's default list printer shows every element, so an interactive client or a `print(client)` would display the full API key — a credential that spends real credits, leaked by an ordinary session transcript. Only the prefix and last four characters are shown.

Usage

```
## S3 method for class 'nrouter_client'
print(x, ...)
```

Arguments

x	An nrouter_client.
...	Ignored.

Value

x, invisibly.

Index

`nrouter-errors`, [3](#)
`nrouter_audio_speech`, [3](#)
`nrouter_audio_transcriptions`, [4](#)
`nrouter_audio_translations`, [4](#)
`nrouter_bytes`, [5](#)
`nrouter_chat`, [5](#)
`nrouter_chat_completions`, [6](#)
`nrouter_chat_completions_stream`, [7](#)
`nrouter_client`, [7](#)
`nrouter_completions`, [8](#)
`nrouter_completions_stream`, [9](#)
`nrouter_condition`, [9](#)
`nrouter_configuration_condition`, [10](#)
`nrouter_count_tokens`, [10](#)
`nrouter_create_video`, [11](#)
`nrouter_default_base_url`, [11](#)
`nrouter_default_connect_timeout_seconds`,
 [8](#), [12](#)
`nrouter_default_stream_idle_seconds`, [8](#),
 [12](#)
`nrouter_default_timeout_seconds`, [8](#), [13](#)
`nrouter_download_video_content`, [13](#)
`nrouter_embeddings`, [14](#)
`nrouter_env_key`, [14](#)
`nrouter_error_from_payload`, [15](#)
`nrouter_header_names`, [15](#)
`nrouter_images_generations`, [16](#)
`nrouter_is_priced`, [16](#)
`nrouter_is_retryable`, [17](#)
`nrouter_key_prefix`, [17](#)
`nrouter_messages`, [18](#)
`nrouter_messages_stream`, [18](#)
`nrouter_meta`, [19](#)
`nrouter_model`, [19](#)
`nrouter_models`, [20](#)
`nrouter_multipart`, [20](#)
`nrouter_request`, [21](#)
`nrouter_resolve_api_key`, [21](#)
`nrouter_responses`, [22](#)
`nrouter_responses_stream`, [22](#)
`nrouter_retrieve_video`, [23](#)
`nrouter_sse_parser`, [23](#)
`nrouter_stream`, [24](#)
`nrouter_transport_condition`, [24](#)

`print.nrouter_client`, [25](#)