Package: omophub (via r-universe)

October 22, 2026

Title R Client for the 'OMOPHub' Medical Vocabulary API

Version 1.10.0

Description Provides an R interface to the 'OMOPHub' API for accessing 'OHDSI ATHENA' standardized medical vocabularies. Supports concept search, semantic search using neural embeddings, concept similarity, vocabulary exploration, hierarchy navigation, relationship queries, concept mappings, and FHIR-to-OMOP concept resolution with automatic pagination.

License MIT + file LICENSE

URL https://github.com/omopHub/omophub-R, https://docs.omophub.com, https://omophub.github.io/omophub-R/, https://omophub.com

BugReports https://github.com/omopHub/omophub-R/issues

Depends R (>= 4.1)

Imports httr2 (>= 1.0.0), R6, rlang (>= 1.0.0), cli, tibble, purrr, glue, checkmate

Suggests testthat (>= 3.0.0), httptest2, httpuv, webmockr, knitr, rmarkdown, jsonlite, keyring, withr

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

VignetteBuilder knitr

Config/pak/sysreqs libssl-dev

Repository https://omophub.r-universe.dev

Date/Publication 2026-10-22 10:46:18 UTC

RemoteUrl https://github.com/omophub/omophub-r

RemoteRef HEAD

RemoteSha 433dfdb9159496966ab923383ee3e9532ddaaca0

Contents

fhir_resolve 2
fhir_resolve_batch 3
fhir_resolve_codeable_concept 4
get_api_key 5
has_api_key 5
omophub_fhir_url 6
OMOPHubClient 7
set_api_key 9

Index **10**

`fhir_resolve` *Resolve a FHIR Coding to an OMOP standard concept*

Description

Standalone wrapper for `client$fhir$resolve()`. See FhirResource for full parameter documentation.

Usage

```
fhir_resolve(client, ...)
```

Arguments

`client`	An OMOPHubClient instance.
`...`	Arguments passed to `FhirResource$resolve()`.

Value

A list with `input` and `resolution`.

See Also

FhirResource, `fhir_resolve_batch()`, `fhir_resolve_codeable_concept()`

Examples

```
## Not run:
client <- OMOPHubClient$new(api_key = Sys.getenv("OMOPHUB_API_KEY"))
fhir_resolve(
  client,
  system = "http://snomed.info/sct",
  code = "44054006",
  resource_type = "Condition"
)

## End(Not run)
```

`fhir_resolve_batch` *Batch-resolve FHIR Codings*

Description

Standalone wrapper for `client$fhir$resolve_batch()`. When `as_tibble = TRUE`, the result is a flat `tibble` suitable for `dplyr`/`tidyr` pipelines.

Usage

```
fhir_resolve_batch(client, ...)
```

Arguments

`client`	An OMOPHubClient instance.
`...`	Arguments passed to `FhirResource$resolve_batch()`.

Value

See `FhirResource$resolve_batch()`.

See Also

FhirResource, `fhir_resolve()`, `fhir_resolve_codeable_concept()`

Examples

```
## Not run:
client <- OMOPHubClient$new(api_key = Sys.getenv("OMOPHUB_API_KEY"))
tbl <- fhir_resolve_batch(
  client,
  codings = list(
    list(system = "http://snomed.info/sct", code = "44054006"),
    list(system = "http://loinc.org", code = "2339-0")
  ),
  as_tibble = TRUE
)
dplyr::filter(tbl, status == "resolved")

## End(Not run)
```

`fhir_resolve_codeable_concept` *Resolve a FHIR CodeableConcept with vocabulary preference*

Description

Standalone wrapper for `client$fhir$resolve_codeable_concept()`.

Usage

```
fhir_resolve_codeable_concept(client, ...)
```

Arguments

`client`	An OMOPHubClient instance.
`...`	Arguments passed to `FhirResource$resolve_codeable_concept()`.

Value

See `FhirResource$resolve_codeable_concept()`.

See Also

FhirResource, `fhir_resolve()`, `fhir_resolve_batch()`

Examples

```
## Not run:
client <- OMOPHubClient$new(api_key = Sys.getenv("OMOPHUB_API_KEY"))
fhir_resolve_codeable_concept(
  client,
  coding = list(
    list(system = "http://snomed.info/sct", code = "44054006"),
    list(system = "http://hl7.org/fhir/sid/icd-10-cm", code = "E11.9")
  ),
  resource_type = "Condition"
)

## End(Not run)
```

get_api_key *Get OMOPHub API Key*

Description

Retrieves the OMOPHub API key from multiple sources in priority order:

1. Explicit argument
2. `OMOPHUB_API_KEY` environment variable
3. System keyring (if `keyring` package is installed)

Usage

```
get_api_key(key = NULL)
```

Arguments

`key`	Optional explicit API key. If provided, this takes precedence.

Value

A character string containing the API key.

Examples

```
## Not run:
# From environment variable
Sys.setenv(OMOPHUB_API_KEY = "your_api_key")
key <- get_api_key()

# Explicit key
key <- get_api_key("your_api_key")

## End(Not run)
```

has_api_key *Check if API Key is Available*

Description

Checks whether an OMOPHub API key is available without throwing an error.

Usage

```
has_api_key()
```

Value

`TRUE` if an API key is available, `FALSE` otherwise.

omophub_fhir_url *OMOPHub FHIR Terminology Service URL*

Description

Convenience helper for constructing the OMOPHub FHIR Terminology Service base URL for a given FHIR version. Use it when configuring an external FHIR client library (`httr2`, `fhircrackr`, etc.) to talk to OMOPHub's FHIR endpoint directly.

Usage

```
omophub_fhir_url(version = c("r4", "r4b", "r5", "r6"))
```

Arguments

`version`	FHIR version prefix. One of `"r4"` (default), `"r4b"`, `"r5"`, or `"r6"`.

Value

A character scalar with the full FHIR base URL, e.g. `"https://fhir.omophub.com/fhir/r4"`.

Examples

```
omophub_fhir_url()
omophub_fhir_url("r5")

## Not run:
# Use with httr2 to call the $lookup operation directly
library(httr2)
req <- request(omophub_fhir_url()) |>
  req_url_path_append("CodeSystem/$lookup") |>
  req_url_query(
    system = "http://snomed.info/sct",
    code = "44054006"
  ) |>
  req_headers(Authorization = paste("Bearer", Sys.getenv("OMOPHUB_API_KEY")))
resp <- req_perform(req)

## End(Not run)
```

`OMOPHubClient` *OMOPHub API Client*

Description

R6 class for interacting with the OMOPHub vocabulary service. Provides access to OHDSI ATHENA standardized medical vocabularies.

Details

The client provides access to these resources:

- `concepts`: Concept lookup and batch operations
- `search`: Basic and advanced concept search
- `vocabularies`: Vocabulary listing and details
- `domains`: Domain listing and concept filtering
- `hierarchy`: Ancestor and descendant navigation
- `relationships`: Concept relationships
- `mappings`: Concept mappings between vocabularies

Value

A new `OMOPHubClient` object.

Active bindings

`concepts` Access to concept operations.

`search` Access to search operations.

`vocabularies` Access to vocabulary operations.

`domains` Access to domain operations.

`hierarchy` Access to hierarchy operations.

`relationships` Access to relationship operations.

`mappings` Access to mapping operations.

`fhir` Access to FHIR-to-OMOP Concept Resolver operations.

Methods

Public methods:

- `OMOPHubClient$new()`
- `OMOPHubClient$print()`
- `OMOPHubClient$clone()`

Method `new()`**:** Create a new OMOPHub client.

Usage:

```
OMOPHubClient$new(
  api_key = NULL,
  base_url = NULL,
  timeout = 30,
  max_retries = 3,
  vocab_version = NULL
)
```

Arguments:

`api_key` API key for authentication. If not provided, reads from `OMOPHUB_API_KEY` environment variable or system keyring.

`base_url` API base URL. Defaults to `https://api.omophub.com/v1`.

`timeout` Request timeout in seconds. Defaults to 30.

`max_retries` Maximum retry attempts for failed requests. Defaults to 3.

`vocab_version` Optional vocabulary version (e.g., "2025.1"). If not specified, uses the latest version.

Method `print()`**:** Print client information.

Usage:

```
OMOPHubClient$print()
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
OMOPHubClient$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
## Not run:
# Create client (uses OMOPHUB_API_KEY env var)
client <- OMOPHubClient$new()

# Or with explicit API key
client <- OMOPHubClient$new(api_key = "your_api_key")

# Search for concepts
results <- client$search$basic("diabetes")

# Get a specific concept
concept <- client$concepts$get(201826)

# Use specific vocabulary version
client <- OMOPHubClient$new(vocab_version = "2025.1")

## End(Not run)
```

`set_api_key` *Set OMOPHub API Key*

Description

Stores the OMOPHub API key in the specified location.

Usage

```
set_api_key(key, store = c("env", "keyring"))
```

Arguments

`key`	The API key to store.
`store`	Where to store the key. One of: • `"env"`: Set as environment variable for current session (default) • `"keyring"`: Store securely in system keyring (requires `keyring` package)

Value

Invisibly returns `TRUE` on success.

Examples

```
## Not run:
# Store in environment (current session only)
set_api_key("your_api_key")

# Store securely in keyring (persistent)
set_api_key("your_api_key", store = "keyring")

## End(Not run)
```

Index

fhir_resolve, 2
fhir_resolve(), *3, 4*
fhir_resolve_batch, 3
fhir_resolve_batch(), *2*, *4*
fhir_resolve_codeable_concept, 4
fhir_resolve_codeable_concept(), *2, 3*
FhirResource, *2–4*

get_api_key, 5

has_api_key, 5

omophub_fhir_url, 6
OMOPHubClient, *2–4*, 7

set_api_key, 9