Package ‘odyssey’

January 8, 2026

Title Interface to the HAL Open Archive API

Version 1.0.0

Description An interface to the search API of 'HAL' <https://hal.science/>, the French open archive for scholarly documents from all academic fields. This package provides programmatic access to the API <https://api.archives-ouvertes.fr/docs> and allows to search for records and download documents.

License GPL (>= 3)

URL https://codeberg.org/nfrerebeau/odyssey,
https://nfrerebeau.r-universe.dev/odyssey,
https://nfrerebeau.codeberg.page/odyssey/

BugReports https://codeberg.org/nfrerebeau/odyssey/issues

Depends R (>= 4.1)

Imports httr2, rlang, utils

Suggests tinytest

Encoding UTF-8

RoxygenNote 7.3.3

Collate 'AllGenerics.R' 'hal_count.R' 'hal_download.R' 'hal_facet.R' 'hal_filter.R' 'hal_group.R' 'hal_parse.R' 'hal_query.R' 'hal_request.R' 'hal_search.R' 'hal_select.R' 'hal_sort.R' 'odyssey-package.R' 'operators.R' 'zzz.R'

NeedsCompilation no

Author Nicolas Frerebeau [aut, cre] (ORCID: <https://orcid.org/0000-0001-5759-4944>),
Université Bordeaux Montaigne [fnd] (ROR: <https://ror.org/03pbgwk21>)

Maintainer Nicolas Frerebeau <nicolas.frerebeau@u-bordeaux-montaigne.fr>

Repository CRAN

Date/Publication 2026-01-08 19:30:02 UTC

Contents

`as.data.frame` *Coerce to a Data Frame*

Description

Coerce to a `data.frame`, if possible.

Usage

```
## S3 method for class 'HALSearch'
as.data.frame(x, ...)
```

Arguments

`x`	An object of class `HALSearch` (typically returned by `hal_search()`).
`...`	Currently not used.

Value

A `data.frame`.

Author(s)

N. Frerebeau

Examples

```
## Not run:
## Simple search
topic <- list("archéologie", "Celtes", "France") # Combined with AND
## Get the first ten results
hal_query(topic) |>
  hal_search(limit = 10) |>
```

```
    as.data.frame()
  ## Get all results
  hal_query(topic) |>
    hal_search(limit = 30, cursor = TRUE) |>
    as.data.frame()

  ## Get a list of archaeological journals
  topic <- c("archéologie", "archaeology", "archäologie") # Combined with OR
  hal_query(topic) |>
    hal_select("title_s", "issn_s") |>
    hal_filter("" %TO% "*" %IN% "issn_s") |>
    hal_sort("title_s") |>
    hal_search(path = "ref", instance = "journal") |>
    as.data.frame()

  ## Get a list of archaeological laboratories
  ## (only joint laboratories of the CNRS and a French university)
  topic <- list("archéologie" %IN% "text", "UMR" %IN% "code_t")
  hal_query(topic) |>
    hal_select("name_s", "acronym_s", "code_s") |>
    hal_filter("VALID" %IN% "valid_s") |>
    hal_sort("acronym_s", decreasing = TRUE) |>
    hal_search(path = "ref", instance = "structure", limit = 15) |>
    as.data.frame()

  ## End(Not run)
```

`hal_count` *Count*

Description

Count

Usage

```
hal_count(query, ...)

## S3 method for class 'HALQuery'
hal_count(query, path = "search", instance = NULL, ...)
```

Arguments

`query`	An object of class `HALQuery` (typically returned by `hal_query()`).
`...`	Currently not used.
`path`	A `character` string giving the API name (defaults to "`search`").
`instance`	A `character` string giving the HAL portal name.

Value

An integer (the number of results).

Author(s)

N. Frerebeau

See Also

Other search tools: hal_download(), hal_search()

Examples

```
## Not run:
## Simple search
topic <- list("archéologie", "Celtes", "France") # Combined with AND
## Get the first ten results
hal_query(topic) |>
  hal_search(limit = 10) |>
  as.data.frame()
## Get all results
hal_query(topic) |>
  hal_search(limit = 30, cursor = TRUE) |>
  as.data.frame()

## Get a list of archaeological journals
topic <- c("archéologie", "archaeology", "archäologie") # Combined with OR
hal_query(topic) |>
  hal_select("title_s", "issn_s") |>
  hal_filter("" %TO% "*" %IN% "issn_s") |>
  hal_sort("title_s") |>
  hal_search(path = "ref", instance = "journal") |>
  as.data.frame()

## Get a list of archaeological laboratories
## (only joint laboratories of the CNRS and a French university)
topic <- list("archéologie" %IN% "text", "UMR" %IN% "code_t")
hal_query(topic) |>
  hal_select("name_s", "acronym_s", "code_s") |>
  hal_filter("VALID" %IN% "valid_s") |>
  hal_sort("acronym_s", decreasing = TRUE) |>
  hal_search(path = "ref", instance = "structure", limit = 15) |>
  as.data.frame()

## End(Not run)
```

hal_download *Download Documents*

Description

Download Documents

Usage

```
hal_download(query, ...)

## S3 method for class 'HALQuery'
hal_download(
  query,
  destination,
  limit = 30,
  start = 0,
  progress = getOption("odyssey.progress"),
  verbose = getOption("odyssey.verbose"),
  ...
)
```

Arguments

`query`	An object of class `HALQuery` (typically returned by `hal_query()`).
`...`	Currently not used.
`destination`	A `character` string giving the directory path where the downloaded files are to be saved.
`limit`	An `integer` giving the maximum number of results. According to HAL policy, it cannot exceed 10000.
`start`	An `integer` specifying an absolute offset in the complete sorted list of matches to be used as the beginning of the current page. Only used if `cursor` is `FALSE`.
`progress`	A `logical` scalar: should a progress bar for for the request be printed?
`verbose`	A `logical` scalar: should extra information be reported?

Value

Invisibly returns `destination`.

Author(s)

N. Frerebeau

See Also

Other search tools: `hal_count()`, `hal_search()`

Examples

```
## Not run:
## Download the 10 most recent archaeological publication
## (if any files)
hal_query("archéologie") |>
  hal_filter("ART" %IN% "docType_s") |>
  hal_sort("producedDate_tdate", decreasing = TRUE) |>
  hal_download(destination = tempdir(), limit = 10)

## End(Not run)
```

---

hal_facet *Facet Search*

Description

Facet Search

Usage

```
hal_facet(query, ...)

## S3 method for class 'HALQuery'
hal_facet(
  query,
  field = NULL,
  limit = 5,
  sort = c("count", "index"),
  prefix = NULL,
  contains = NULL,
  pivot = NULL,
  range = NULL,
  ignore_case = FALSE,
  ...
)
```

Arguments

`query`	An object of class `HALQuery` (typically returned by `hal_query()`).
`...`	Currently not used.
`field`	A `character` string specifying the field to group by.
`limit`	An `integer` giving the maximum number of results per group.
`sort`	A `character` string specifying the field to be used to sort the results.
`prefix`	A `character` string.
`contains`	A `character` string.

pivot	A character string.
range	A list containing the following components: "range", "start", "end", "gap".
ignore_case	A logical scalar: should character case be ignored?

Value

An object of class HALQuery.

Author(s)

N. Frerebeau

See Also

Other query tools: hal_filter(), hal_group(), hal_query(), hal_select(), hal_sort(), operators

Examples

```
## Not run:
## Get the number of archaeological document by journal
hal_query("archéologie") |>
  hal_facet(field = "journalTitle_s", sort = "count", decreasing = TRUE) |>
  hal_search() |>
  as.data.frame()

## Get the number of documents per year
hal_query("archéologie") |>
  hal_facet(
    sort = "count",
    range = list(
      field = "producedDateY_i",
      start = 2000,
      end = 2020,
      gap = 1
    )
  ) |>
  hal_search() |>
  as.data.frame()

## End(Not run)
```

hal_filter *Filter Results*

Description

Filter Results

Usage

```
hal_filter(query, ...)

## S3 method for class 'HALQuery'
hal_filter(query, value, field = NULL, ...)
```

Arguments

`query` An object of class `HALQuery` (typically returned by `hal_query()`).

`...` Currently not used.

`value` A `character` string specifying the value to be used to filter the results.

`field` A `character` string specifying the field to filter along.

Value

An object of class `HALQuery`.

Author(s)

N. Frerebeau

See Also

Other query tools: `hal_facet()`, `hal_group()`, `hal_query()`, `hal_select()`, `hal_sort()`, `operators`

Examples

```
## Simple filer
hal_query() |> hal_filter("file", "submitType_s")

## Advanced filter
hal_query() |> hal_filter("THESE" %OR% "HDR", "docType_s")

## Multiple filters
hal_query() |>
  hal_filter("NOW-1MONTHS/DAY" %TO% "NOW/HOUR", "submittedDate_tdate") |>
  hal_filter("-notice", "submitType_s")

## Range filters
hal_query() |> hal_filter(2000 %TO% 2013, "submittedDateY_i")

hal_query() |> hal_filter("Aa" %TO% "Ab", "city_s")
```

hal_group *Group Results*

Description

Group Results

Usage

```
hal_group(query, ...)

## S3 method for class 'HALQuery'
hal_group(query, field, limit = 20, sort = "score", decreasing = FALSE, ...)
```

Arguments

`query`	An object of class `HALQuery` (typically returned by `hal_query()`).
`...`	Currently not used.
`field`	A `character` string specifying the field to group by.
`limit`	An `integer` giving the maximum number of results per group.
`sort`	A `character` string specifying the field to be used to sort the results.
`decreasing`	A `logical` scalar: should the sort be increasing or decreasing?

Value

An object of class `HALQuery`.

Author(s)

N. Frerebeau

See Also

Other query tools: `hal_facet()`, `hal_filter()`, `hal_query()`, `hal_select()`, `hal_sort()`, `operators`

Examples

```
## Not run:
## Get the most recent archaeological publication (in French) by journal
hal_query("archéologie") |>
  hal_filter("ART" %IN% "docType_s") |>
  hal_select("producedDate_tdate", "docid") |>
  hal_sort("producedDate_tdate", decreasing = TRUE) |>
  hal_group(
    field = "journalTitle_s",
    sort = "producedDate_tdate",
    decreasing = TRUE
  ) |>
  hal_search() |>
  as.data.frame()

## End(Not run)
```

hal_query *Build Query*

Description

Initialize a query to the HAL API.

Usage

```
hal_query(value = "*", field = "text", ...)
```

Arguments

value	A vector giving the topics to be searched for (see details).
field	A character string specifying the field to search within.
...	Currently not used.

Value

A list of class HALQuery containing at least the following components:

q (default to "text:*").

fl default to "docid,label_s").

wt (default to "json").

Author(s)

N. Frerebeau

See Also

Other query tools: hal_facet(), hal_filter(), hal_group(), hal_select(), hal_sort(), operators

Examples

```
## Search for 'asie' in the 'text' field
hal_query("asie")

## Search for 'japon' and 'france' in the 'title_t' field
term <- list("japon", "france")
hal_query(term, field = "title_t")

## Search for 'Journal', and 'Histoire' or 'History' in the 'title_t' field
term <- list("Journal", c("Histoire", "History"))
hal_query(term, field = "title_t")
```

hal_search *Search*

Description

Search

Usage

```
hal_search(query, ...)

## S3 method for class 'HALQuery'
hal_search(
  query,
  path = "search",
  instance = NULL,
  limit = 30,
  start = 0,
  cursor = FALSE,
  max_requests = Inf,
  on_error = "stop",
  progress = getOption("odyssey.progress"),
  verbose = getOption("odyssey.verbose"),
  ...
)
```

Arguments

`query` An object of class `HALQuery` (typically returned by `hal_query()`).

`...` Currently not used.

`path` A `character` string giving the API name (defaults to "`search`").

`instance` A `character` string giving the HAL portal name.

`limit` An `integer` giving the maximum number of results. According to HAL policy, it cannot exceed 10000.

start An integer specifying an absolute offset in the complete sorted list of matches to be used as the beginning of the current page. Only used if cursor is FALSE.

cursor A logical scalar: should a cursor be used for the pagination of results? If TRUE, the sort parameter of the query will set to "docid asc".

max_requests An integer specifying the maximum number of requests to perform. Use Inf to perform all requests. Only used if cursor is TRUE.

on_error A character string specifying what should happen if a request fails (see httr2::req_perform_iterati
Only used if cursor is TRUE.

progress A logical scalar: should a progress bar for for the request be printed?

verbose A logical scalar: should extra information be reported?

Value

A list of class HALSearch.

Author(s)

N. Frerebeau

References

Apache Solr documentation.

HAL search documentation.

HAL reference frame documentation.

See Also

Other search tools: hal_count(), hal_download()

Examples

```
## Not run:
## Simple search
topic <- list("archéologie", "Celtes", "France") # Combined with AND
## Get the first ten results
hal_query(topic) |>
  hal_search(limit = 10) |>
  as.data.frame()
## Get all results
hal_query(topic) |>
  hal_search(limit = 30, cursor = TRUE) |>
  as.data.frame()

## Get a list of archaeological journals
topic <- c("archéologie", "archaeology", "archäologie") # Combined with OR
hal_query(topic) |>
  hal_select("title_s", "issn_s") |>
  hal_filter("" %TO% "*" %IN% "issn_s") |>
  hal_sort("title_s") |>
```

```
    hal_search(path = "ref", instance = "journal") |>
    as.data.frame()

  ## Get a list of archaeological laboratories
  ## (only joint laboratories of the CNRS and a French university)
  topic <- list("archéologie" %IN% "text", "UMR" %IN% "code_t")
  hal_query(topic) |>
    hal_select("name_s", "acronym_s", "code_s") |>
    hal_filter("VALID" %IN% "valid_s") |>
    hal_sort("acronym_s", decreasing = TRUE) |>
    hal_search(path = "ref", instance = "structure", limit = 15) |>
    as.data.frame()

  ## End(Not run)
```

hal_select *Select Fields*

Description

Select Fields

Usage

```
hal_select(query, ...)

## S3 method for class 'HALQuery'
hal_select(query, ...)
```

Arguments

query	An object of class HALQuery (typically returned by hal_query()).
...	One or more character string separated by commas giving the fields to be returned.

Value

An object of class HALQuery.

Author(s)

N. Frerebeau

See Also

Other query tools: hal_facet(), hal_filter(), hal_group(), hal_query(), hal_sort(), operators

Examples

```
## Select fields
hal_query() |> hal_select("label_s")

(q <- hal_query() |> hal_select("halId_s", "uri_s"))

## Update query
hal_select(q, "docType_s")
```

hal_sort *Sort Results*

Description

Sort Results

Usage

```
hal_sort(query, field, decreasing = FALSE, ...)

hal_sort(query, field, decreasing = FALSE, ...)
```

Arguments

`query`	An object of class `HALQuery` (typically returned by `hal_query()`).
`field`	A `character` string specifying the field to be used to sort the results.
`decreasing`	A `logical` scalar: should the sort be increasing or decreasing?
`...`	Currently not used.

Value

An object of class `HALQuery`.

Author(s)

N. Frerebeau

See Also

Other query tools: `hal_facet()`, `hal_filter()`, `hal_group()`, `hal_query()`, `hal_select()`, `operators`

Examples

```
## Select fields
(q <- hal_query() |> hal_sort("docid"))

## Update query
hal_sort(q, "producedDate_tdate", decreasing = TRUE)
```

operators *Operators*

Description

Operators

Usage

```
x %OR% y

x %AND% y

x %NOT% y

x %BY% y

x %IN% y

x %TO% y
```

Arguments

x, y	A (list of) character vector.

Value

A character string.

Author(s)

N. Frerebeau

See Also

Other query tools: hal_facet(), hal_filter(), hal_group(), hal_query(), hal_select(), hal_sort()

Examples

```
## Infix operators are composed left to right
c("aluminium", "fer") %BY% 3 %IN% "title_t"

## We need parenthesis
("ecology" %IN% "title_t") %AND% ("cell" %IN% "text")

## A vector allows to combine all the terms with OR
c("Histoire", "History") %IN% "title_t"
```

```
## A list allows to combine terms with AND
list("Paris", "France", "history") %NOT% list("Texas", "history")
```

Index