

Package: amcat4r (via r-universe)

June 3, 2026

Type Package

Title R client bindings for the amcat4 API

Description Functions to work with AmCAT4 from R - create projects, run queries, etc.

Version 4.2.11

Depends R (>= 4.1.0)

Imports cli, curl (>= 7.0.0), dplyr, glue, httr2, jsonlite, methods, pillar, purrr, rappdirs, remotes, rlang, tibble, utils, lubridate

Suggests covr, dockr, rstudioapi, testthat, withr

Remotes JBGruber/dockr

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.0.0

Config/pak/sysreqs git libssl-dev

Repository <https://ccs-amsterdam.r-universe.dev>

Date/Publication 2026-06-03 07:50:33 UTC

RemoteUrl <https://github.com/ccs-amsterdam/amcat4r>

RemoteRef HEAD

RemoteSha da95df4f0bbcd15bc832bc017b9a1deac32f8f2c

Contents

add_index_user	2
amcat_login	3
create_index	4
create_user	5
delete_by_query	6
delete_documents	6

delete_index	7
delete_index_user	7
delete_user	8
get_document	8
get_documents	9
get_fields	9
get_index	10
get_user	10
index_exists	11
list_index_users	11
list_indexes	12
list_users	12
modify_index_user	13
modify_user	13
pillar_shaft.id_col	14
ping	14
query_aggregate	15
query_documents	15
refresh_index	17
reindex	18
run_amcat_docker	19
set_fields	19
set_metareader_access	20
stop_amcat_docker	21
update_by_query	21
update_documents	22
update_tags	23
upload_documents	24

Index **26**

add_index_user	<i>Add index user</i>
----------------	-----------------------

Description

Add index user

Usage

```
add_index_user(index, email, role, credentials = NULL)
```

Arguments

index	The index to list
email	The email of an (existing) user
role	The role of the user (METAREADER, READER, WRITER, ADMIN)
credentials	The credentials to use. If not given, uses last login information

amcat_login

Authenticate to an AmCAT instance

Description

Authenticate to an AmCAT instance

Usage

```
amcat_login(
  server,
  api_key = NULL,
  token_refresh = FALSE,
  force_refresh = FALSE,
  cache = NULL,
  test_login = TRUE
)
```

Arguments

server	URL of the AmCAT instance
api_key	The API Key to use for authentication (API version 4.1+)
token_refresh	Whether to enable refresh token rotation (see details; for API version 4.0).
force_refresh	Overwrite existing cached authentication
cache	select where tokens should be cached to suppress the user menu. 1 means to store on disk, 2 means to store only in memory.
test_login	If TRUE (default), fetch /users/me to test succesful login.

Details

Enabling refresh token rotation ensures added security as leaked refresh tokens also become invalidated after a short while. It is currently disabled by default as it is not fully supported by the underlying httr2 package.

If you select to store your tokens on disk in the interactive menu, they are stored in the location indicated by `rappdirs::user_cache_dir("httr2")`.

The function needs to open a browser, which will usually only work in an interactive session. However, you can save the returned object in an rds file (with `saveRDS()`) and tell `amcat4r` where to look for it: `options(amcat4r_token_cache = "path/to/location/tokens.rds")`. If you still have issues in an interactive session, check [browseURL](#) to see if you can set a browser manually.

It returns an `amcat4_token` object, which contains a number of standard fields (`host`, `api_version`, `authorization`) and fields depending on the authentication method, currently `api_token` (for 4.1) and the `httr2_token` fields for 4.0

Value

an amcat4_token object, which besides the token itself will contain:

- **host:** The base URL of the AmCAT server.
- **api_version:** Character string of the API version.
- **authorization:** The authorization configuration

Examples

```
## Not run:
amcat_login("https://middlecat.up.railway.app/api/demo_resource")

## End(Not run)
```

create_index	<i>Create an index</i>
--------------	------------------------

Description

Create or modify an index.

Usage

```
create_index(
  index,
  name = index,
  description = NULL,
  create_fields = NULL,
  guest_role = NULL,
  credentials = NULL
)

modify_index(
  index,
  name = index,
  description = NULL,
  guest_role = NULL,
  credentials = NULL
)
```

Arguments

index	short name of the index to create (follows naming conventions of Elasticsearch, see details).
name	optional more descriptive name of the index to create (all characters are allowed here)

description	optional description of the index to create
create_fields	create fields in the new index.
guest_role	Role for unauthorized users. Options are "none", "observer", "metareader", "reader", and "writer".
credentials	The credentials to use. If not given, uses last login information.

Details

The short name for the new index (index argument) must meet these criteria:

- Lowercase only
- Cannot include \, /, *, ?, ", <, >, |, :, (space), , (comma), #
- Cannot start with -, _, +
- Cannot be . or ..
- Cannot be longer than 255 character (note that some symbols like emojis take up two characters)
- If names start with ., the index will be hidden and non accessible

Functions

- `modify_index()`: Modify an index

Examples

```
## Not run:
create_index("test_index")

## End(Not run)
```

create_user	<i>Create a new user</i>
-------------	--------------------------

Description

Create a new user

Usage

```
create_user(email, role = "writer", index_access = NULL, credentials = NULL)
```

Arguments

email	email of the user to add.
role	global role of the user ("metareader", "reader", "writer" or "admin").
index_access	index to grant access to for the new user.
credentials	The credentials to use. If not given, uses cached login information.

delete_by_query *Delete documents by query*

Description

Delete documents by query

Usage

```
delete_by_query(
  index,
  ids = NULL,
  queries = NULL,
  filters = NULL,
  credentials = NULL
)
```

Arguments

index	The index to query
ids	A optional vector of ids to add/remove tags from
queries	An optional vector of queries to run (implicit OR)
filters	An optional list of filters, e.g. list(publisher='A', date=list(gte='2022-01-01'))
credentials	The credentials to use. If not given, uses last login information

Examples

```
## Not run:
delete_by_query("my_index", filters=list(publisher='NY Times'))
delete_by_query("my_index", ids=c(42, 69))
delete_by_query("my_index", queries="advertisement")

## End(Not run)
```

delete_documents *Delete documents from index*

Description

Delete documents from index

Usage

```
delete_documents(index, docid, credentials = NULL)
```

Arguments

index	The index name in which documents should be deleted.
docid	the .ids of the documents that should be deleted.
credentials	The credentials to use. If not given, uses last login information.

delete_index	<i>Delete an index</i>
--------------	------------------------

Description

Delete an index

Usage

```
delete_index(index, credentials = NULL)
```

Arguments

index	name of the index on this server
credentials	The credentials to use. If not given, uses last login information

Examples

```
## Not run:
delete_index("test_index")

## End(Not run)
```

delete_index_user	<i>Delete index user</i>
-------------------	--------------------------

Description

Delete index user

Usage

```
delete_index_user(index, email, credentials = NULL)
```

Arguments

index	The index to list
email	The email of an (existing) user
credentials	The credentials to use. If not given, uses last login information

delete_user	<i>Delete new user</i>
-------------	------------------------

Description

Delete new user

Usage

```
delete_user(email, credentials = NULL)
```

Arguments

email	email of the user to remove.
credentials	The credentials to use. If not given, uses cached login information.

get_document	<i>Retrieve a single document</i>
--------------	-----------------------------------

Description

Retrieve a single document

Usage

```
get_document(index, doc_id, fields, credentials = NULL)
```

Arguments

index	The index to get fields for
doc_id	A single document_id
fields	Optional character vector listing the fields to retrieve
credentials	The credentials to use. If not given, uses last login information

Value

A tibble with one row containing the requested fields

get_documents	<i>Retrieve multiple documents using a purrr map over get_document</i>
---------------	--

Description

Retrieve multiple documents using a purrr map over get_document

Usage

```
get_documents(index, doc_ids, fields, credentials = NULL, ...)
```

Arguments

index	The index to get fields for
doc_ids	A vector of document_ids
fields	Optional character vector listing the fields to retrieve
credentials	The credentials to use. If not given, uses last login information
...	Other options to pass to map, e.g. .progress

Value

A tibble with one row containing the requested fields

get_fields	<i>Get fields</i>
------------	-------------------

Description

Get fields

Usage

```
get_fields(index, credentials = NULL)
```

Arguments

index	The index to get fields for
credentials	The credentials to use. If not given, uses last login information

get_index	<i>Get a single index</i>
-----------	---------------------------

Description

Get a single index

Usage

```
get_index(index, credentials = NULL)
```

Arguments

index	name of the index
credentials	The credentials to use. If not given, uses last login information.

Value

a list with details about this index, or NULL if it does not exist

Examples

```
## Not run:  
get_index("my_index")  
  
## End(Not run)
```

get_user	<i>Get information about a user</i>
----------	-------------------------------------

Description

Get information about a user

Usage

```
get_user(user = "me", credentials = NULL)
```

Arguments

user	The user to get information on, or 'me' to get information on the current user
credentials	The credentials to use. If not given, uses cached login information.

index_exists	<i>Check if an index exists</i>
--------------	---------------------------------

Description

Convenience function that calls !is.null(get_index(...))

Usage

```
index_exists(index, credentials = NULL)
```

Arguments

index	name of the index
credentials	The credentials to use. If not given, uses last login information.

Value

a list with details about this index, or NULL if it does not exist

Examples

```
## Not run:  
index_exists("my_index")  
  
## End(Not run)
```

list_index_users	<i>List index users</i>
------------------	-------------------------

Description

List index users

Usage

```
list_index_users(index, credentials = NULL)
```

Arguments

index	The index to list
credentials	The credentials to use. If not given, uses last login information

list_indexes	<i>List the indexes on this server</i>
--------------	--

Description

List the indexes on this server

Usage

```
list_indexes(credentials = NULL)
```

Arguments

`credentials` The credentials to use. If not given, uses last login information.

Value

a tibble with index information including `id`, `name`, `user_role`, `archived`, `description`, `folder`, `image_url`

Examples

```
## Not run:  
list_indexes()  
  
## End(Not run)
```

list_users	<i>List users</i>
------------	-------------------

Description

List users

Usage

```
list_users(credentials = NULL)
```

Arguments

`credentials` The credentials to use. If not given, uses cached login information.

modify_index_user	<i>Modify index user</i>
-------------------	--------------------------

Description

Modify index user

Usage

```
modify_index_user(index, email, role, credentials = NULL)
```

Arguments

index	The index to list
email	The email of an (existing) user
role	The role of the user (METAREADER, READER, WRITER, ADMIN)
credentials	The credentials to use. If not given, uses last login information

modify_user	<i>Modify an existing user</i>
-------------	--------------------------------

Description

Modify an existing user

Usage

```
modify_user(email, role = "writer", credentials = NULL)
```

Arguments

email	email of the user to modify.
role	global role of the user ("metareader", "reader", "writer" or "admin").
credentials	The credentials to use. If not given, uses cached login information.

pillar_shaft.id_col *Truncate id columns when printing*

Description

Truncate id columns when printing

Usage

```
## S3 method for class 'id_col'  
pillar_shaft(x, ...)
```

Arguments

x	id column in a data.frame with amcat4 data.
...	Arguments passed to methods.

ping *Check if an amcat instance is reachable*

Description

Check if a server is reachable by sending a request to its config endpoint.

Usage

```
ping(server)
```

Arguments

server	A character string of the server URL. If missing the server for the logged in session is tried.
--------	---

Value

A logical value indicating if the server is reachable.

Examples

```
## Not run:  
ping("http://localhost/amcat")  
  
## End(Not run)
```

query_aggregate	<i>Conduct a query and return the resulting documents</i>
-----------------	---

Description

Conduct a query and return the resulting documents

Usage

```
query_aggregate(
  index,
  axes = NULL,
  queries = NULL,
  filters = NULL,
  credentials = NULL
)
```

Arguments

index	The index to query
axes	The aggregation axes, e.g. <code>list(list(field="publisher", list(field="date", interval="year")))</code>
queries	An optional vector of queries to run (implicit OR)
filters	An optional list of filters, e.g. <code>list(publisher='A', date=list(gte='2022-01-01'))</code>
credentials	The credentials to use. If not given, uses last login information

Examples

```
## Not run:
query_aggregate("state_of_the_union",
  axes = list(list(field="party", list(field="date", interval="year"))),
  queries = c("war", "peace"),
  filters = list(party = c("Democratic", "Republican"),
    date = list(gte = "1900-01-01"))

## End(Not run)
```

query_documents	<i>Conduct a query and return the resulting documents</i>
-----------------	---

Description

This function queries the database and retrieves documents that fit the query.

Usage

```

query_documents(
  index,
  queries = NULL,
  fields = c("date", "title"),
  filters = NULL,
  per_page = 200,
  max_pages = 1,
  page = NULL,
  merge_tags = ";",
  scroll = "5m",
  verbose = TRUE,
  credentials = NULL
)

```

Arguments

index	The index to query.
queries	An optional vector of queries to run (implicit OR).
fields	An optional vector of fields to return (returns all fields if NULL).
filters	An optional list of filters, e.g. <code>list(publisher='A', date=list(gte='2022-01-01'))</code> .
per_page	Number of results per page.
max_pages	Stop after getting this many pages. Set to Inf to retrieve all.
page	Request a specific page (is ignored when scroll is set).
merge_tags	Character to merge tag fields with, default ';'. Set to NULL to prevent merging.
scroll	Instead of scrolling indefinitely until max_pages is reached, you can set a time here that amcat4r keeps retrieving new pages before it stops (see examples).
verbose	Should a progress bar be printed during upload.
credentials	The credentials to use. If not given, uses last login information

Details

This function queries the database and retrieves documents that fit the query. The results can be further narrowed down using filters. If there are many results, they are divided into pages to keep the data that is sent from the amcat instance small. You can use the function to iterate over these pages to retrieve many or all or just a specific one (if you want to batch process an index and only work on, e.g., 100 documents at a time).

AmCAT uses the Elasticsearch query language. Find the documentation here: <https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-query-string-query.html#query-string-query-notes>.

Examples

```

## Not run:
# retrieve all fields from all documents

```

```

query_documents("state_of_the_union", queries = NULL, fields = NULL)

# query "migration" and select text field
query_documents("state_of_the_union", queries = "migration", fields = "text")

# note that by default, the query searches all text fields (see ?get_fields for field types)
query_documents("state_of_the_union", queries = "1908", fields = "text")

# to narrow a search to the title field use
query_documents("state_of_the_union", queries = "title:1908", fields = "text")

# searches support wild cards
query_documents("state_of_the_union", queries = "migra*", fields = NULL)

# if you query more than one term, you can use OR or leave it out since it is
# used implicitly anyway. So these two do the same
query_documents("state_of_the_union", queries = "migra* OR refug*")
query_documents("state_of_the_union", queries = "migra* refug*")

# you can search for literal matches using double quotes
query_documents("state_of_the_union", queries = '"migration laws"')

# and you can chain several boolean operators together
query_documents("state_of_the_union", queries = "(migra* OR refug*) AND illegal NOT legal")

# get only the first result
query_documents("state_of_the_union", queries = "migra*", per_page = 1, page = 1, fields = NULL)

# get the 81st result
query_documents("state_of_the_union", queries = "migra*", per_page = 80, page = 2, fields = NULL)

# If you want to retrieve many pages/documents at once, you should use the
# scroll API by setting a scroll value. E.g., to scroll for 5 seconds before
# collecting results use:
query_documents("state_of_the_union", scroll = "5s", per_page = 1, max_pages = Inf)
# or scroll for 5 minutes
query_documents("state_of_the_union", scroll = "5m", per_page = 1, max_pages = Inf)

## End(Not run)

```

refresh_index

Refresh an index

Description

Refresh an index

Usage

```
refresh_index(index, credentials = NULL)
```

Arguments

index	The index to refresh
credentials	The credentials to use. If not given, uses last login information

reindex	<i>Reindex documents to a destination index</i>
---------	---

Description

Reindexes documents from index to destination. If the destination does not exist it is created. Field changes are specified via fields: any field not mentioned is carried over from the source unchanged.

Usage

```
reindex(
  index,
  destination,
  fields = NULL,
  name = destination,
  description = NULL,
  guest_role = NULL,
  queries = NULL,
  filters = NULL,
  credentials = NULL
)
```

Arguments

index	The source index name
destination	The destination index name
fields	Optional named list of per-field changes. Each element is a named list with any of: rename (new field name), exclude (logical, drop field), type (new AM-CAT field type string). E.g. <code>list(old_field=list(rename="new_field"), bad=list(exclude=TRUE), text_col=list(type="keyword"))</code> .
name	Display name for the destination index (defaults to destination); only used when creating a new index
description	Optional description; only used when creating a new index
guest_role	Optional guest role; only used when creating a new index
queries	Optional list of query strings to filter documents during reindex
filters	Optional list of filters to apply during reindex
credentials	The credentials to use. If not given, uses last login information

run_amcat_docker	<i>Run docker containers with AmCAT modules</i>
------------------	---

Description

Automatically sets up services defined in a Docker Compose file. Only options relevant to AmCAT are implemented.

Usage

```
run_amcat_docker(compose = NULL, force_install = FALSE)
```

Arguments

compose	Path to a Docker Compose file. Uses https://github.com/JBGruber/amcat4docker/blob/main/docker-compose.yml by default.
force_install	If TRUE, removes all containers and re-creates them from the compose file. If 2, the images are also re-downloaded. Danger: this will destroy the indexes in your containers!

set_fields	<i>Set fields</i>
------------	-------------------

Description

Set fields

Usage

```
set_fields(index, fields, credentials = NULL)
```

Arguments

index	The index to set fields for
fields	A list with fields and data types, e.g. list(author="keyword")
credentials	The credentials to use. If not given, uses last login information

Details

AmCAT currently supports the following field types:

- **text**: For general text columns
- **keyword**: Keywords - this is like text, but is not parsed as words. Most suitable for 'factor' / 'group' level data
- **tag**: Tags - like keywords, but the assumption is that every document can have multiple values.

- **date**: For date or date+time columns
- **boolean**: Boolean (true/false) columns
- **number**: General numeric columns
- **integer**: Whole numbers
- **object**: Nested dictionaries. These are not really analysed by AmCAT, but can store any data you need
- **json**: Generic json data. This is analysed (and searchable) as text, use object if AmCAT does not need to search it
- **vector**: Dense vectors, useful for e.g. embedding vectors
- **geo_point**: Geometrical locations (long+lat)
- **url**: A generic URL
- **image, video**: A URL pointing to an image or video file

set_metareader_access *Set field-level access for metareaders*

Description

Controls which fields metareaders (unauthenticated/guest users) can access, and how much of those fields they can see.

Usage

```
set_metareader_access(index, fields, credentials = NULL)
```

Arguments

index	The index to modify fields for.
fields	A named list of field access settings. Each element is named after a field and contains a list with: <ul style="list-style-type: none"> • access: "none" (hidden), "read" (full), or "snippet" (truncated). • max_snippet: (optional, only used when access = "snippet") a list with nomatch_chars (chars to show without a query match, default 100), max_matches (max highlighted matches, default 0), match_chars (chars per match, default 50).
credentials	The credentials to use. If not given, uses last login information.

Examples

```
## Not run:
# Make 'url' fully readable and 'text' snippet-only for metareaders:
set_metareader_access("de-news", list(
  url = list(access = "read"),
  text = list(access = "snippet", max_snippet = list(nomatch_chars = 150))
))

## End(Not run)
```

stop_amcat_docker	<i>Stop docker containers with AmCAT modules</i>
-------------------	--

Description

Stop docker containers with AmCAT modules

Usage

```
stop_amcat_docker(compose = NULL, filters = NULL)
```

Arguments

compose	Path to a Docker Compose file. Uses https://github.com/JBGruber/amcat4docker/blob/main/docker-compose.yml by default.
filters	Names of containers or named values for other filters.

Details

Stops either the containers defined in a compose file or the filters. If filters is set, compose is ignored.

Examples

```
## Not run:  
# stop AmCAT modules  
stop_amcat_docker()  
  
# stop container by id  
stop_amcat_docker(filters = c(id = "a6cbe4787227"))  
  
# stop all containers  
stop_amcat_docker(filters = "")  
  
## End(Not run)
```

update_by_query	<i>Update documents by query</i>
-----------------	----------------------------------

Description

Update documents by query

Usage

```
update_by_query(
  index,
  field,
  value,
  ids = NULL,
  queries = NULL,
  filters = NULL,
  credentials = NULL
)
```

Arguments

index	The index to query
field	The field name to update
value	The new value for the field
ids	A optional vector of ids to add/remove tags from
queries	An optional vector of queries to run (implicit OR)
filters	An optional list of filters, e.g. list(publisher='A', date=list(gte='2022-01-01'))
credentials	The credentials to use. If not given, uses last login information

Examples

```
## Not run:
update_by_query("my_index", "publisher", "NYT", filters=list(publisher='New York Times'))
update_by_query("my_index", "sentiment", -1, ids=c(3, 7, 9, 11))

## End(Not run)
```

update_documents	<i>Update documents</i>
------------------	-------------------------

Description

Update documents

Usage

```
update_documents(index, ids = NULL, documents, credentials = NULL)
```

Arguments

index	The index name to create.
ids	The IDs (.id) of the document to update description (if NULL, the .id column from documents will be used).
documents	A data frame with columns to update.
credentials	The credentials to use. If not given, uses last login information.

update_tags	<i>Add or remove tags to/from documents by query or ID</i>
-------------	--

Description

Add or remove tags to/from documents by query or ID

Usage

```
update_tags(
  index,
  action,
  field,
  tag,
  ids = NULL,
  queries = NULL,
  filters = NULL,
  credentials = NULL
)
```

Arguments

index	The index to query
action	'add' or 'remove' the tags
field	The tag field name
tag	The tag to add or remove
ids	A vector of ids to add/remove tags from
queries	An optional vector of queries to run (implicit OR)
filters	An optional list of filters, e.g. <code>list(publisher='A', date=list(gte='2022-01-01'))</code>
credentials	The credentials to use. If not given, uses last login information

Examples

```
## Not run:
set_fields("state_of_the_union", list(test = "tag"))
update_tags(
  index = "state_of_the_union",
  action = "add",
  field = "test",
  tag = "test",
  filters = list(party = "Republican",
                date = list(gte = "2000-01-01"))
)

## End(Not run)
```

upload_documents	<i>Upload documents</i>
------------------	-------------------------

Description

Upload documents

Usage

```
upload_documents(  
  index,  
  documents,  
  columns = NULL,  
  chunk_size = 100L,  
  max_tries = 5L,  
  verbose = TRUE,  
  credentials = NULL  
)
```

Arguments

index	The name of the index documents should be added to.
documents	A data frame with columns title, text, date, and optional other columns. An .id column is interpreted as elastic document IDs
columns	An optional list with data types, e.g. list(author = "keyword").
chunk_size	Uploads are broken into chunks to prevent errors. Smaller chunks are less error-prone, but this also makes the upload slower.
max_tries	In case something goes wrong, how often should the function retry to send the documents?
verbose	Should a progress bar be printed during upload.
credentials	The credentials to use. If not given, uses last login information.

Value

Nothing.

Examples

```
## Not run:  
amcat_login("http://localhost/amcat")  
docs <- data.frame(  
  date = "2024-01-01",  
  title = "This is a title",  
  text = "This is some text"  
)  
create_index(index = "new_index")
```

upload_documents

25

```
upload_documents(index = "new_index",  
                 documents = docs)
```

```
## End(Not run)
```

Index

[add_index_user](#), 2
[amcat_login](#), 3

[browseURL](#), 3

[create_index](#), 4
[create_user](#), 5

[delete_by_query](#), 6
[delete_documents](#), 6
[delete_index](#), 7
[delete_index_user](#), 7
[delete_user](#), 8

[get_document](#), 8
[get_documents](#), 9
[get_fields](#), 9
[get_index](#), 10
[get_user](#), 10

[index_exists](#), 11

[list_index_users](#), 11
[list_indexes](#), 12
[list_users](#), 12

[modify_index \(create_index\)](#), 4
[modify_index_user](#), 13
[modify_user](#), 13

[pillar_shaft.id_col](#), 14
[ping](#), 14

[query_aggregate](#), 15
[query_documents](#), 15

[refresh_index](#), 17
[reindex](#), 18
[run_amcat_docker](#), 19

[set_fields](#), 19
[set_metareader_access](#), 20

[stop_amcat_docker](#), 21

[update_by_query](#), 21
[update_documents](#), 22
[update_tags](#), 23
[upload_documents](#), 24