Package 'hdar'

September 22, 2024

Type Package

Title 'REST' API Client for Accessing Data on 'WEkEO HDA V2'

Version 1.0.1

URL https://www.wekeo.eu/

BugReports https://github.com/eea/hdar/issues

Maintainer Matteo Mattiuzzi <matteo.mattiuzzi@eea.europa.eu>

Description Provides seamless access to the WEkEO Harmonised Data Access (HDA) API, enabling users to query, download, and process data efficiently from the HDA platform. With 'hdar', researchers and data scientists can integrate the extensive HDA datasets into their R workflows, enhancing their data analysis capabilities. Comprehensive information on the API functionality and usage is available at <https://gateway.prod.wekeo2.eu/hda-broker/docs>.

License EUPL (>= 1.2)

Encoding UTF-8

RoxygenNote 7.3.2

Imports R6, httr2, jsonlite, magrittr, htmltools, stringr, humanize

Depends R (>= 2.10),

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Matteo Mattiuzzi [aut, cre]

Repository CRAN

Date/Publication 2024-09-21 22:40:06 UTC

Contents

Auth *Auth Class*

Description

Authorization

Methods

Public methods:

- `Auth$new()`
- `Auth$token()`
- `Auth$get_token()`
- `Auth$clone()`

Method `new()`**:** This function initializes a new instance of the 'Auth' class with the specified parameters.

Usage:

```
Auth$new(user = NULL, password = NULL)
```

Arguments:

`user` A character string representing the username for authentication.

`password` A character string representing the password for authentication.

Returns: An instance of the 'Auth' class.

Method `token()`**:** This function retrieves a previously generated token.

Usage:

```
Auth$token()
```

Returns: A character string representing the retrieved token.

Method `get_token()`**:** This function generates a unique token for authentication or other purposes.

Usage:

```
Auth$get_token()
```

Returns: A character string representing the generated token.

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

Usage:

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

Arguments:

`deep` Whether to make a deep clone.

Client *Client Class*

Description

The Client is the central gateway for interfacing with the HDA Service. It provides a comprehensive suite of methods to perform operations and retrieve data from the service efficiently.

Public fields

`apiUrl` API endpoint

Methods

Public methods:

- `Client$new()`
- `Client$token()`
- `Client$get_token()`
- `Client$send_request()`
- `Client$show_terms()`
- `Client$terms_and_conditions()`
- `Client$datasets()`
- `Client$search()`
- `Client$get_querytemplate()`
- `Client$generate_query_template()`
- `Client$clone()`

Method `new()`**:** Constructor for the 'Client' class. Initializes a new instance with authentication credentials.

Usage:

```
Client$new(user, password, save_credentials = FALSE)
```

Arguments:

`user` Character string representing the username for authentication.

`password` Character string representing the password for authentication.

`save_credentials` A logical value indicating whether to save the credentials to a configuration file. Default is FALSE.

Returns: An instance of the 'Client' class.

Method `token()`**:** Retrieves the current authentication token.

Usage:

```
Client$token()
```

Returns: Character string representing the authentication token.

Method `get_token()`**:** Generates a new authentication token.

Usage:

```
Client$get_token()
```

Returns: Character string representing the newly generated token.

Method `send_request()`**:** Sends a specified request to the server and returns the response.

Usage:

```
Client$send_request(req, raw_response = FALSE)
```

Arguments:

`req` A request object or list representing the HTTP request.

`raw_response` Optional logical value indicating whether the raw response should be returned instead of the parsed body.

Returns: A response object containing the server's response.

Method `show_terms()`**:** This function displays the terms and conditions for the services.

Usage:

```
Client$show_terms()
```

Returns: An HTML document containing the terms and conditions in a collapsible format.

Method `terms_and_conditions()`**:** Function to retrieve and accept terms and conditions. Accepting T&C is permanent, it is enough to run this function one. To read T&C see `show_terms`.

Usage:

```
Client$terms_and_conditions(term_id, reject = FALSE)
```

Arguments:

`term_id` A character vector of term_ids that you wish to accept. If missing current status is returned. Use "**all**" if you want to accept all terms at once.

`reject` Logical, default 'FALSE'. If TRUE it inverts the operation and the provided term_id's are rejected/revoked.

Returns: A data frame reflecting the actual acceptance status for each term.

Method `datasets()`**:** Lists datasets available on WEkEO, optionally filtered by a text pattern.

Usage:

```
Client$datasets(pattern = NULL)
```

Arguments:

`pattern` Optional character string to filter dataset names by matching text.

Returns: List containing datasets and associated information.

Method `search()`**:** This function performs a search based on a specified query and returns an instance of `SearchResults`.

Usage:

```
Client$search(json_query, limit = NULL)
```

Arguments:

`json_query` Character string representing the search query.

`limit` Optional; a number specifying the maximum number of results to return.

Returns: An instance of the `SearchResults` class containing the search results.

Method `get_querytemplate()`**:** Retrieves the raw query metadata for a specified datasetId.

Usage:

```
Client$get_querytemplate(datasetId, to_json = FALSE)
```

Arguments:

`datasetId` Character, representing the dataset's identifier.

`to_json` Logical; if 'TRUE', returns the data in JSON format.

Returns: List or JSON file containing the raw query options.

Method `generate_query_template()`**:** This function generates a query template based on a specified datasetId.

Usage:

```
Client$generate_query_template(datasetId)
```

Arguments:

`datasetId` A numeric or character ID representing the dataset.

Returns: A JSON representing the generated query template.

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

Usage:

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

Arguments:

`deep` Whether to make a deep clone.

Note

There are some inconsistencies between the return of 'GET querymetadata' and what must be submitted to the HDA. Use `generate_query_template` to resolve these inconsistencies.

See Also

`show_terms` to read the Terms and conditions.

`SearchResults` for details on the returned object.

SearchResults *SearchResults Class*

Description

This class handles the storage and manipulation of search results including downloading resources based on a search query.

SearchResults

Public fields

`results` Stores the search results data.

`total_count` Stores the total count of results' element.

`total_size` Stores the total size of results

Methods

Public methods:

- `SearchResults$new()`
- `SearchResults$download()`
- `SearchResults$clone()`

Method `new()`**:** Initializes a new SearchResults object with the specified client, results, and dataset identifier.

Usage:

```
SearchResults$new(client, results, dataset_id)
```

Arguments:

`client` An object containing the API client used to interact with the dataset.

`results` List containing search results.

`dataset_id` The identifier for the dataset being queried.

Returns: SearchResult instance

Method `download()`**:** Downloads resources based on stored results or selected indices of results.

Usage:

```
SearchResults$download(
  output_dir,
  selected_indexes,
  stop_at_failure = TRUE,
  force = FALSE
)
```

Arguments:

`output_dir` A string specifying the directory where downloaded files will be saved.

`selected_indexes` Optional; indices of the specific results to download.

`stop_at_failure` Optional; controls whether the download process of multiple files should immediately stop upon encountering the first failure.

`force` Optional; forces the download even if the file already exists in the specified output directory.

Returns: Nothing returned but downloaded files are saved at the specified location.

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

Usage:

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

Arguments:

`deep` Whether to make a deep clone.

Index