Package 'MolgenisArmadillo'

October 14, 2024

Type Package

Version 2.7.0

Title Armadillo Client for the Armadillo Service

Description A set of functions to manage data shared on a 'MOLGENIS Armadillo' server.

Maintainer Mariska Slofstra <m.k.slofstra@umcg.nl>

Depends R (>= 3.6)

Imports base64enc, httr, urltools, dplyr, purrr, stringr, tidyr, tibble, MolgenisAuth (>= 0.0.25), arrow, rlist, httr2, readr, cli

Suggests stringi, withr, knitr, testthat, webmockr, mockery, datasets, rmarkdown

License LGPL (>= 2.1)

URL https://github.com/molgenis/molgenis-r-armadillo/, https://molgenis.github.io/molgenis-r-armadillo/

BugReports https://github.com/molgenis/molgenis-r-armadillo/issues/

RoxygenNote 7.3.1

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

Author Mariska Slofstra [aut, cre] (<https://orcid.org/0000-0002-0400-0468>), Sido Haakma [aut] (<https://orcid.org/0000-0003-1278-9144>), Tommy de Boer [aut] (<https://orcid.org/0000-0002-4492-7225>), Fleur Kelpin [aut] (<https://orcid.org/0000-0003-1527-5130>), Morris Swertz [aut] (<https://orcid.org/0000-0002-0979-3401>), Tim Cadman [aut] (<https://orcid.org/0000-0002-7682-5645>)

Repository CRAN

Date/Publication 2024-10-14 13:00:03 UTC

Contents

`.add_slash_if_missing` *Add a Slash to a URL if Missing*

Description

This function ensures that a given URL string ends with a slash ('/'). If the URL does not end with a slash, it appends one.

Usage

```
.add_slash_if_missing(url)
```

Arguments

`url`	A character string representing the URL.

Value

A character string with a trailing slash if it was missing.

`.compress_resource` *Helper function for compressing to an RDS file*

Description

Helper function for compressing to an RDS file

Usage

```
.compress_resource(resource, file)
```

Arguments

`resource`	the resource to write to file
`file`	the name of the file (without extension)

Value

the extension of the file

`.compress_table` *Helper function for compressing to a parquet file*

Description

Helper function for compressing to a parquet file

Usage

```
.compress_table(table, file)
```

Arguments

`table`	the table to write to file
`file`	the name of the file (without extension)

Value

the extension of the file

.format_api_posts *Formats API posts based on subset definition*

Description

Formats API posts based on subset definition

Usage

```
.format_api_posts(posts, subset_def)
```

Arguments

posts	A list of API posts
subset_def	A tibble containing subset definition

Value

A tibble consisting of original subset_def with columns 'posts' and 'status' appended.

.get_linkfile_content *Helper function to get the contents of a linkfile*

Description

Helper function to get the contents of a linkfile

Usage

```
.get_linkfile_content(project, object_name)
```

Arguments

project	projectname where the linkfile is stored
object_name	folder/name of linkfile

Value

the contents of the linkfile

.load_linked_table *Helper function to extract the source parquet file in a linkfile*

Description

Helper function to extract the source parquet file in a linkfile

Usage

```
.load_linked_table(file, columns)
```

Arguments

file	source table parquet file
columns	character list of columns to select from source file

Value

the contents of the file, as data frame

.load_resource *Helper function to extract an RDS file*

Description

Helper function to extract an RDS file

Usage

```
.load_resource(file)
```

Arguments

file	file to extract

Value

the contents of the file

.load_table *Helper function to extract a parquet file*

Description

Helper function to extract a parquet file

Usage

```
.load_table(file)
```

Arguments

file	file to extract

Value

the contents of the file, as data frame

armadillo.copy_resource *Copy resource*

Description

Copy resource

Usage

```
armadillo.copy_resource(
  project,
  folder,
  name,
  new_folder = folder,
  new_name = name
)
```

Arguments

project	study or other variable collection
folder	the folder containing the resource
name	specific resource for copy action
new_folder	name of the folder in which to place the copy, defaults to folder
new_name	name of the copy, defaults to name

Value

the response from the server

Examples

```
## Not run:
armadillo.copy_resource(
  project = "gecko",
  folder = "core_all",
  name = "table1",
  new_folder = "core_all_v2",
)

## End(Not run)
```

`armadillo.copy_table` *Copy table*

Description

Copy table

Usage

```
armadillo.copy_table(
  project,
  folder,
  name,
  new_folder = folder,
  new_name = name
)
```

Arguments

`project`	study or other variable collection
`folder`	the folder containing the table
`name`	specific table for copy action
`new_folder`	name of the folder in which to place the copy, defaults to folder
`new_name`	name of the copy, defaults to name

Value

the response from the server

Examples

```
## Not run:
armadillo.copy_table(
  project = "gecko",
  folder = "core_all",
  name = "table1",
  new_folder = "core_all_v2",
)

## End(Not run)
```

`armadillo.create_project`

Create a project for a variable collection

Description

Create a project for a variable collection

Usage

```
armadillo.create_project(
  project_name = NULL,
  users = NULL,
  overwrite_existing = "choose"
)
```

Arguments

`project_name`	The name of the project to create. The project name • cannot be empty. • must be no more than 56 characters. • cannot end with a -. • must consist of lowercase letters and numbers.
`users`	A list collection of the users that should have access to the project
`overwrite_existing`	Character, specifying action to take if project still exists: 'choose' (default) displays a menu giving the option to overwrite or not, 'yes' overwrites the existing project and 'no' exists the function with a message.

Examples

```
## Not run:
armadillo.create_project("gecko")

## End(Not run)
```

armadillo.delete_project *Delete project*

Description

A project represents usually a study or collection of variables

Usage

```
armadillo.delete_project(project_name)
```

Arguments

project_name	the name of the study or collection of variables name

Examples

```
## Not run:
armadillo.delete_project(project_name = "gecko")

## End(Not run)
```

armadillo.delete_project_folder *Delete project folder*

Description

Delete project folder

Usage

```
armadillo.delete_project_folder(project, folder)
```

Arguments

project	project to delete the object from
folder	folder to delete the object from

armadillo.delete_resource *Delete resource*

Description

Delete resource

Usage

```
armadillo.delete_resource(project, folder, name)
```

Arguments

project	project to delete the resource from
folder	folder to delete the resource from
name	resource name

Examples

```
## Not run:
armadillo.delete_resource(
  project = "gecko",
  folder = "core_all",
  name = "table1"
)

## End(Not run)
```

armadillo.delete_table *Delete table*

Description

Delete table

Usage

```
armadillo.delete_table(project, folder, name)
```

Arguments

project	project to delete the table from
folder	folder to delete the table from
name	table name

Examples

```
## Not run:
armadillo.delete_table(
  project = "gecko",
  folder = "core_all",
  name = "table1"
)

## End(Not run)
```

`armadillo.get_projects_info`
Gets the Projects information

Description

Gets the Projects information

Usage

```
armadillo.get_projects_info()
```

Value

the projects and their information

Examples

```
## Not run:
armadillo.get_projects_info()

## End(Not run)
```

`armadillo.get_project_users`
Gets the users of an given project name

Description

Gets the users of an given project name

Usage

```
armadillo.get_project_users(project_name)
```

Arguments

project_name	the name of the project to extract the users from

Value

List of all users within "project_name"

Examples

```
## Not run:
armadillo.get_project_users("some-project")

## End(Not run)
```

---

armadillo.install_packages

Install package

Description

Installs a user defined package into the provided profile. The package is automatically whitelisted after installation. Only available during development.

Usage

```
armadillo.install_packages(paths, profile = "default")
```

Arguments

paths	the path(s) to the package(s), can be a vector or a string
profile	the selected profile

---

armadillo.list_projects

List the projects

Description

List the projects

Usage

```
armadillo.list_projects()
```

Value

the projects

Examples

```
## Not run:
armadillo.list_projects()

## End(Not run)
```

`armadillo.list_resources` *List the resources in a project*

Description

List the resources in a project

Usage

```
armadillo.list_resources(project)
```

Arguments

`project`	the shared project in which the resources are located

Value

the resources in the project

Examples

```
## Not run:
armadillo.list_resources("gecko")

## End(Not run)
```

armadillo.list_tables *List the tables in a project*

Description

List the tables in a project

Usage

```
armadillo.list_tables(project)
```

Arguments

project	the shared project in which the tables are located

Value

the table names, without the extension

Examples

```
## Not run:
armadillo.list_tables("gecko")

## End(Not run)
```

armadillo.load_resource *Load a resource from a project*

Description

Load a resource from a project

Usage

```
armadillo.load_resource(project, folder, name)
```

Arguments

project	study or collection variables
folder	the folder containing the resource
name	name of the resource

Value

the loaded resource

Examples

```
## Not run:
armadillo.load_resource(
  project = "gecko",
  folder = "core_all",
  name = "lc_core_1"
)

## End(Not run)
```

`armadillo.load_table` *Load a table from a project*

Description

Load a table from a project

Usage

```
armadillo.load_table(project, folder, name)
```

Arguments

`project`	study or collection variables
`folder`	the folder containing the table
`name`	name of the table

Value

the contents of the table file, as data frame

Examples

```
## Not run:
armadillo.load_table(
  project = "gecko",
  folder = "core_all",
  name = "lc_core_1"
)

## End(Not run)
```

armadillo.login *Login*

Description

Interactively obtains an id token and uses it to create a session token for an Armadillo Service

Usage

```
armadillo.login(armadillo)
```

Arguments

armadillo	URL of the Armadillo server,

Value

the id token

Examples

```
## Not run:
armadillo.login(
  "https://armadillo.dev.molgenis.org"
)
armadillo.login("http://localhost:8080")

## End(Not run)
```

armadillo.login_basic *Login with username / password (meant for dev and test environments)*

Description

Login with username / password (meant for dev and test environments)

Usage

```
armadillo.login_basic(armadillo, username, password)
```

Arguments

armadillo	URL of the Armadillo server
username	the username
password	the password

Examples

```
## Not run:
armadillo.login(
  "https://armadillo.dev.molgenis.org", "admin", "admin"
)
armadillo.login("http://localhost:8080", "admin", "admin")

## End(Not run)
```

`armadillo.move_resource`

Move the resource

Description

Move the resource

Usage

```
armadillo.move_resource(
  project,
  folder,
  name,
  new_folder = folder,
  new_name = name
)
```

Arguments

`project`	a study or collection of variables
`folder`	the folder containing the resource to move
`name`	a resource to move
`new_folder`	the folder to move the resource to, defaults to folder
`new_name`	use to rename the file, defaults to name

Value

NULL, invisibly

Examples

```
## Not run:
armadillo.move_resource(
  project = "gecko",
  folder = "core_all",
  name = "table1",
```

```
  new_folder = "core_all_v2",
)

## End(Not run)
```

`armadillo.move_table` *Move the table*

Description

Move the table

Usage

```
armadillo.move_table(
  project,
  folder,
  name,
  new_folder = folder,
  new_name = name
)
```

Arguments

`project`	a study or collection of variables
`folder`	the folder containing the table to move
`name`	a table to move
`new_folder`	the folder to move the table to, defaults to folder
`new_name`	use to rename the file, defaults to name

Value

NULL, invisibly

Examples

```
## Not run:
armadillo.move_table(
  project = "gecko",
  folder = "core_all",
  name = "table1",
  new_folder = "core_all_v2",
)

## End(Not run)
```

`armadillo.subset` *Describes data available to subset and makes subset*

Description

This automates the process of:

1. Checking what data is available to create subsets
2. Make the subset

Usage

```
armadillo.subset(
  input_source = NULL,
  subset_def = NULL,
  source_project = NULL,
  source_folder = NULL,
  source_table = NULL,
  target_project = NULL,
  target_folder = NULL,
  target_table = NULL,
  target_vars = NULL,
  new_project = NULL,
  dry_run = NULL
)
```

Arguments

`input_source`	Character specifying how information about the target view is provided: choose 'subset_def' if providing a subset definition object, or 'arguments' if providing information directly.
`subset_def`	R object containing subset definition created by `armadillo.subset_definition()`. Compulsory if input_source = 'subset_def'
`source_project`	project from which to subset data
`source_folder`	folder from which to subset data. Compulsory if input_source = 'arguments'.
`source_table`	table from which to subset data. Compulsory if input_source = 'arguments'.
`target_project`	project to upload subset to. Will be created if it doesn't exist.
`target_folder`	folder to upload subset to. Will be created if it doesn't exist. Compulsory if input_source = 'arguments'.
`target_table`	table to upload subset to. Compulsory if input_source = 'arguments'.
`target_vars`	variables from 'source_table' to include in the view. Compulsory if input_source = 'arguments'.
`new_project`	Deprecated: use `target_project` instead
`dry_run`	Defunct: previously enabgled dry-run to check which variables are missing

Value

missing variables provided in the subset definition

Examples

```
## Not run:
armadillo.subset(
  source_project = "gecko",
  target_project = "study1",
  subset_def = local_subset
)

## End(Not run)
```

`armadillo.subset_definition`
Builds an R object containing info required to make subsets

Description

Builds an R object containing info required to make subsets

Usage

```
armadillo.subset_definition(reference_csv = NULL, vars = NULL)
```

Arguments

`reference_csv`	`.csv` file containing details of the variable to subset. Must contain 5 columns: 'source_folder' specifying the folder from which to subset, 'souce_table' specifying the table from which to subset, 'target_folder' specifying the folder in which to create the subset 'target_table' specifying the name of the subset and 'variable' specifying the variable(s) to include in the subset. Note that 'source_project' and 'target_project' are specified as arguments to 'armadillo.subset'.
`vars`	Deprecated: use `reference_csv` instead

Value

A dataframe containing variables that is used for input in the `armadillo.subset()` method

Examples

```
## Not run:
armadillo.subset_definition(
  reference_csv = "C:/tmp/vars.csv"
)
```

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

`armadillo.upload_resource`

Uploads a resource to a folder in a project

Description

Uploads a resource to a folder in a project

Usage

```
armadillo.upload_resource(project, folder, resource, name = NULL)
```

Arguments

`project`	the project to upload to
`folder`	the folder to upload to
`resource`	the resource to upload
`name`	name of the resource (optional)

Examples

```
## Not run:
armadillo.upload_table(
  project = "gecko",
  folder = "core_all",
  table1
)

## End(Not run)
```

`armadillo.upload_table`

Uploads a table to a folder in a project

Description

Uploads a table to a folder in a project

Usage

```
armadillo.upload_table(project, folder, table, name = NULL)
```

Arguments

project	the project to upload to
folder	the folder to upload to
table	the table to upload
name	name of the table (optional)

Examples

```
## Not run:
armadillo.upload_table(
  project = "gecko",
  folder = "core_all",
  table1
)

## End(Not run)
```

Index