

Package ‘mlstrOpalr’

October 11, 2023

Type Package

Title Support Compatibility Between 'Maelstrom' R Packages and 'Opal' Environment

Version 1.0.2

Description Functions to support compatibility between 'Maelstrom' R packages and 'Opal' environment. 'Opal' is the 'OBiBa' core database application for biobanks. It is used to build data repositories that integrates data collected from multiple sources. 'Opal Maelstrom' is a specific implementation of this software. This 'Opal' client is specifically designed to interact with 'Opal Maelstrom' distributions to perform operations on the R server side. The user must have adequate credentials. Please see <<https://opaldoc.obiba.org/>> for complete documentation.

License GPL-3

LazyData true

Depends R (>= 3.4)

Imports dplyr, rlang, tidyr, stringr, tools, opalr, fabR (>= 2.0.0), madshapR (>= 1.0.2)

URL <https://github.com/maelstrom-research/mlstrOpalr>

BugReports <https://github.com/maelstrom-research/mlstrOpalr/issues>

RoxygenNote 7.2.3

Suggests knitr

VignetteBuilder knitr

Encoding UTF-8

Language en-US

NeedsCompilation no

Author Guillaume Fabre [aut, cre, cph]
(<<https://orcid.org/0000-0002-0124-9970>>),
Maelstrom-research group [cph, fnd],
OBiBa group [ctb]

Maintainer Guillaume Fabre <gui.joseph.fabre@gmail.com>

Repository CRAN

Date/Publication 2023-10-11 09:20:02 UTC

R topics documented:

data_dict_opalr_fix	2
mlstrOpalr_help	3
opal_files_pull	4
opal_files_push	5
opal_project_create	6
opal_tables_pull	7
opal_tables_push	9
taxonomy_opal_get	10
taxonomy_opal_mlstr_get	11
valueType_list	12

Index **14**

data_dict_opalr_fix *Transform a data dictionary object from 'opalr' format to Opal format*

Description

Transforms a data dictionary from 'opalr' format to Opal format. A data dictionary imported using the 'opalr' package has a different structure and will be transformed to the structure used in Maelstrom packages.

Usage

```
data_dict_opalr_fix(data_dict)
```

Arguments

data_dict	A list of tibble(s) representing meta data to be transformed. Automatically generated if not provided.
-----------	--

Details

Opal is the OBiBa application for data management, and Opal environments can be integrated with RStudio environments using the package opalr. The user must have adequate credentials to interact with a specific Opal environment. Some function errors produced may be associated with the handler or Opal read/write permissions.

Value

A list of tibble(s) identifying a data dictionary.

See Also

Please see [Opal documentation](#) for complete documentation. `madshapR::as_data_dict_mlstr()`

Examples

```
## Not run:

library(opalr)
opal <-
  opal.login('administrator','password',
    url = 'https://opal-demo.obiba.org/')

# use DEMO_files provided by the package
library(madshapR)
library(stringr)

dossier <-
  DEMO_files[str_detect(names(DEMO_files),"dataset_MELBOURNE")]

tempdir <- basename(tempdir())
try(opal_project_create(opal, tempdir))

try(
  opal_tables_push(
    opal, dossier, project_name = tempdir, .force = TRUE, .overwrite = TRUE))

# get the data dictionary and reshape it.
data_dict <-
  try(
    opal.table_dictionary_get(
      opal, project = tempdir, table = "dataset_MELBOURNE_1"))

data_dict <- try(data_dict_opalr_fix(data_dict))

## End(Not run)
```

mlstrOpalr_help

Call the help center for full documentation

Description

This function is a direct call to the documentation in the repository hosting the package. The user can access the description of the latest version of the package, the vignettes, and the list of functions.

Usage

```
mlstrOpalr_help()
```

Value

Nothing to be returned. The function opens a web page.

Examples

```
{  
  mlstrOpalr_help()  
}
```

opal_files_pull	<i>Download files from an Opal environment</i>
-----------------	--

Description

Downloads files from an Opal environment to the local environment. This is a wrapper function for `opalr::opal.file_download()`.

Usage

```
opal_files_pull(opal, from, to)
```

Arguments

opal	Opal login attributes.
from	A character string of a path where the files will be taken from in R.
to	A character string of a path where the files will be placed to in Opal.

Details

Opal is the OBiBa application for data management, and Opal environments can be integrated with RStudio environments using the package `opalr`. The user must have adequate credentials to interact with a specific Opal environment. Some function errors produced may be associated with the handler or Opal read/write permissions.

Value

Folder(s) containing files coming from Opal in user R environment.

See Also

Please see [Opal documentation](#) for complete documentation.

Examples

```
## Not run:

library(opalr)
opal <-
  opal.login('administrator', 'password',
            url = 'https://opal-demo.obiba.org/')

tempdir <- tempdir()
invisible(dir.create(paste0(tempdir, "/a_file")))
invisible(file.create(paste0(tempdir, "/a_file", "/file.txt")))
try(opal_files_push(opal, from = paste0(tempdir, "/a_file"), to = tempdir))
try(opal_files_pull(opal, from = paste0(tempdir, "/a_file"), to = tempdir))

## End(Not run)
```

opal_files_push

Upload files to an Opal environment

Description

Uploads files from the local environment to an Opal environment. This is a wrapper function for `opalr::opal.file_upload()`.

Usage

```
opal_files_push(opal, from, to)
```

Arguments

opal	Opal login attributes.
from	A character string of a path where the files will be taken from in R.
to	A character string of a path where the files will be placed to in Opal.

Details

Opal is the OBiBa application for data management, and Opal environments can be integrated with RStudio environments using the package `opalr`. The user must have adequate credentials to interact with a specific Opal environment. Some function errors produced may be associated with the handler or Opal read/write permissions.

Value

Folder(s) containing files coming from the user R environment in Opal. The path to Opal needs to be pasted with Opal absolute path.

See Also

Please see [Opal documentation](#) for complete documentation.

Examples

```
## Not run:

library(opalr)
opal <-
  opal.login('administrator', 'password',
            url = 'https://opal-demo.obiba.org/')

tempdir <- tempdir()
invisible(dir.create(paste0(tempdir, "/a_file")))
invisible(file.create(paste0(tempdir, "/a_file", "/file.txt")))
try(opal_files_push(opal, from = paste0(tempdir, "/a_file"), to = tempdir))

## End(Not run)
```

opal_project_create *Create a project in an Opal environment*

Description

Creates an empty project in an Opal environment. This is a wrapper function for `opalr::opal.project_create()`.

Usage

```
opal_project_create(opal, project, tag = NULL)
```

Arguments

opal	Opal login attributes.
project	A character string to name the project in Opal.
tag	A character string to provide a tag for the Opal project.

Details

Opal is the OBiBa application for data management, and Opal environments can be integrated with RStudio environments using the package `opalr`. The user must have adequate credentials to interact with a specific Opal environment. Some function errors produced may be associated with the handler or Opal read/write permissions.

Value

A project in an Opal environment. If the project already exists, it will remain as it is, and no new project is created. The user must have adequate credentials to interact with their Opal environment.

See Also

Please see [Opal documentation](#) for complete documentation.

Examples

```
## Not run:

library(opalr)
opal <-
  opal.login('administrator', 'password',
            url = 'https://opal-demo.obiba.org/')

tempdir <- basename(tempdir())
try(opal_project_create(opal, tempdir))

## End(Not run)
```

opal_tables_pull	<i>Download tables from an Opal project as a dossier</i>
------------------	--

Description

Downloads a dossier or dataset from an Opal environment to the local environment. This is a wrapper function for `opalr::opal.table_get()` and `opalr::opal.table_dictionary_get()`.

Usage

```
opal_tables_pull(
  opal,
  project,
  table_list = NULL,
  content = c("dataset", "data_dict"),
  keep_as_dossier = TRUE,
  .remove_id = FALSE
)
```

Arguments

opal	Opal login attributes.
project	A character string specifying the Opal project name.
table_list	A vector character string specifying Opal tables name.
content	A vector of character string which indicates if the function returns a dataset, or data dictionary. Default is 'dataset'.
keep_as_dossier	whether to return a dossier or a dataset if there is only one table. TRUE by default, if FALSE returns dataset.
.remove_id	whether to return the id column created in Opal or not. TRUE by default.

Details

Opal is the OBiBa application for data management, and Opal environments can be integrated with RStudio environments using the package `opalr`. The user must have adequate credentials to interact with a specific Opal environment. Some function errors produced may be associated with the handler or Opal read/write permissions.

Value

Objects (tibbles and list of tibbles) representing tables and their respective data dictionary.

See Also

Please see [Opal documentation](#) for complete documentation. `madshapR::as_dossier()` `madshapR::as_dataset()` `madshapR::as_data_dict_mlstr()`

Examples

```
## Not run:

library(opalr)
opal <-
  opal.login('administrator','password',
            url = 'https://opal-demo.obiba.org/')

# use DEMO_files provided by the package
library(madshapR)
library(stringr)

dossier <-
  DEMO_files[str_detect(names(DEMO_files), "dataset_MELBOURNE")]

tempdir <- basename(tempdir())
try(opal_project_create(opal, tempdir))

try(
  opal_tables_push(
    opal, dossier, project_name = tempdir, .force = TRUE, .overwrite = TRUE))

##### Example pull a table from a project.
try(
  opal_tables_pull(
    opal, project = tempdir, table_list = 'dataset_MELBOURNE_1'))

## End(Not run)
```

opal_tables_push	<i>Upload datasets into an Opal environment as tables in an Opal project</i>
------------------	--

Description

Uploads a dossier or dataset from the local environment to an Opal environment. This is a wrapper function for `opalr::opal.table_create()`, `opalr::opal.table_save()` and `opalr::opal.table_dictionary_update`

Usage

```
opal_tables_push(
  opal,
  dossier = NULL,
  data_dict = NULL,
  dataset = NULL,
  table_name = NULL,
  project_name,
  .force = FALSE,
  .overwrite = FALSE
)
```

Arguments

opal	Opal login attributes.
dossier	List of tibble, each of them being datasets.
data_dict	A list of tibble(s) representing meta data of an associated dataset. Automatically generated if not provided.
dataset	A tibble identifying the input dataset observations associated to its data dictionary.
table_name	A character string specifying an Opal table name.
project_name	A character string specifying the Opal project name.
.force	If the destination already exists, stop with an informative message if this flag is FALSE (default).
.overwrite	If the destination table already exists, it will be replaced (deleted, re-created with associated permissions reinstated and then imported). Otherwise the table will be updated (data dictionaries merge may conflict). Default is FALSE.*

Details

Opal is the OBiBa application for data management, and Opal environments can be integrated with RStudio environments using the package `opalr`. The user must have adequate credentials to interact with a specific Opal environment. Some function errors produced may be associated with the handler or Opal read/write permissions.

Value

A table or table(s) in Opal.

See Also

Please see [Opal documentation](#) for complete documentation. `madshapR::as_dossier()` `madshapR::as_dataset()` `madshapR::as_data_dict_mlstr()`

Examples

```
## Not run:

library(opalr)
opal <-
  opal.login('administrator', 'password',
            url = 'https://opal-demo.obiba.org/')

# use DEMO_files provided by the package
library(madshapR)
library(stringr)
dossier <- DEMO_files[str_detect(names(DEMO_files), "dataset_MELBOURNE")]

tempdir <- basename(tempdir())
try(opal_project_create(opal, tempdir))

# push a table in a project.
try(
  opal_tables_push(
    opal,
    dataset = dossier$dataset_MELBOURNE_1,
    table_name = 'dataset_MELBOURNE_1',
    project_name = tempdir,
    .force = TRUE,
    .overwrite = TRUE))

## End(Not run)
```

taxonomy_opal_get

Get available taxonomies in format used by Opal

Description

Downloads all taxonomies from an Opal server in a specific tibble format used by Opal.

Usage

```
taxonomy_opal_get(opal)
```

Arguments

opal Opal login attributes.

Details

Opal is the OBiBa application for data management, and Opal environments can be integrated with RStudio environments using the package `opalr`. The user must have adequate credentials to interact with a specific Opal environment. Some function errors produced may be associated with the handler or Opal read/write permissions.

Value

A tibble identifying a taxonomy (generally generated from Opal taxonomy).

See Also

Please see [Opal documentation](#) for complete documentation. `madshapR::as_taxonomy()`

Examples

```
## Not run:

library(opalr)
opal <-
  opal.login('administrator', 'password',
            url = 'https://opal-demo.obiba.org/')

try(taxonomy_opal_get(opal))

## End(Not run)
```

taxonomy_opal_mlstr_get

Get available taxonomies in format used by Maelstrom Research

Description

Downloads all taxonomies from an Opal server in a specific tibble format used in the Maelstrom Research ecosystem.

Usage

```
taxonomy_opal_mlstr_get(opal)
```

Arguments

opal Opal login attributes.

Details

Opal is the OBiBa application for data management, and Opal environments can be integrated with RStudio environments using the package `opalr`. The user must have adequate credentials to interact with a specific Opal environment. Some function errors produced may be associated with the handler or Opal read/write permissions.

Value

A tibble identifying a taxonomy (generally generated from Opal taxonomy).

See Also

Please see [Opal documentation](#) for complete documentation. `madshapR::as_taxonomy()`

Examples

```
## Not run:

library(opalr)
opal <-
  opal.login('administrator', 'password',
            url = 'https://opal-demo.obiba.org/')

try(taxonomy_opal_mlstr_get(opal))

## End(Not run)
```

valueType_list

Built-in tibble of allowed valueType values

Description

Provides a built-in tibble showing the list of allowed Opal valueType values and their corresponding R data types. This tibble is mainly used for internal processes and programming.

Usage

```
valueType_list
```

Format

tibble:

A data frame with 12 rows and 6 columns:

valueType data type as described in Opal

typeof data type provided by `base::typeof`

class data class provided by base::class
call function to transpose object according base::do.call function
toValueType ensemble data type as described in Opal
toTypeof ensemble data type provided by base::typeof
genericType ensemble data type which valueType belongs ...

Details

The valueType is a property of a variable and is required in certain functions to determine the handling of the variables. The valueType refers to the OBiBa-internal type of a variable. It is specified in a data dictionary in a column valueType and can be associated with variables as attributes. Acceptable valueTypes include 'text', 'integer', 'decimal', 'boolean', 'datetime', 'date'). The full list of OBiBa valueType possibilities and their correspondence with R data types are available using [valueType_list](#).

See Also

[Opal documentation](#)

Examples

```
{  
  valueType_list  
}
```

Index

* datasets

valueType_list, [12](#)

data_dict_opalr_fix, [2](#)

madshapR::as_data_dict_mlstr(), [3](#), [8](#), [10](#)

madshapR::as_dataset(), [8](#), [10](#)

madshapR::as_dossier(), [8](#), [10](#)

madshapR::as_taxonomy(), [11](#), [12](#)

mlstrOpalr_help, [3](#)

opal_files_pull, [4](#)

opal_files_push, [5](#)

opal_project_create, [6](#)

opal_tables_pull, [7](#)

opal_tables_push, [9](#)

opalr::opal.file_download(), [4](#)

opalr::opal.file_upload(), [5](#)

opalr::opal.project_create(), [6](#)

opalr::opal.table_create(), [9](#)

opalr::opal.table_dictionary_get(), [7](#)

opalr::opal.table_dictionary_update(),
[9](#)

opalr::opal.table_get(), [7](#)

opalr::opal.table_save(), [9](#)

taxonomy_opal_get, [10](#)

taxonomy_opal_mlstr_get, [11](#)

valueType_list, [12](#), [13](#)