

Package ‘data.world’

April 5, 2018

Title Functions and Add-Ins for Working with 'data.world' Data Sets
and Projects

Version 1.2.2

Description High-level tools for working with 'data.world' data sets. 'data.world' is a platform where you can find interesting data, store and showcase your own data and data projects, and find and collaborate with other members. In addition to exploring, querying and charting data on the data.world site, you can access data via 'API' endpoints and integrations. Use this package to access, query and explore data sets, and to publish your insights. Visit <https://data.world>, for additional information.

Depends R (>= 3.3.0), dwapi (>= 0.1.3)

Imports httr, ini, miniUI, shiny, stringi

Suggests covr, knitr, lintr, readr, rmarkdown, testthat (>= 2.0.0)

License Apache License 2.0

Encoding UTF-8

LazyData true

URL <https://github.com/datadotworld/data.world-r>

BugReports <https://github.com/datadotworld/data.world-r/issues>

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Rafael Pereira [aut, cre],
Triet Le [aut],
Bryon Jacob [aut],
Scott Came [aut]

Maintainer Rafael Pereira <rafael.pereira@data.world>

Repository CRAN

Date/Publication 2018-04-04 22:29:14 UTC

R topics documented:

cfg	2
cfg_env	3
cfg_saved	3
data.world	4
data.world-defunct	4
qry_sparql	5
qry_sql	6
query	7
save_config	8
set_config	8
Index	10

cfg	<i>Construct configuration object using runtime values.</i>
-----	---

Description

Construct configuration object using runtime values.

Usage

```
cfg(auth_token)
```

Arguments

auth_token API authorization token.

Value

Object of type cfg

Examples

```
runtime_cfg <- data.world::cfg(auth_token = "YOUR_TOKEN")
data.world::set_config(runtime_cfg)
```

cfg_env	<i>Construct configuration object using environment variables.</i>
---------	--

Description

Construct configuration object using environment variables.

Usage

```
cfg_env(auth_token_var = "DW_AUTH_TOKEN")
```

Arguments

auth_token_var Name of environment variable for API authorization token (default: DW_AUTH_TOKEN).

Value

Object of type cfg_env

Examples

```
envvars_cfg <- data.world::cfg_env()  
data.world::set_config(envvars_cfg)
```

cfg_saved	<i>Construct configuration object using file-based configuration.</i>
-----------	---

Description

Construct configuration object using file-based configuration.

Usage

```
cfg_saved(profile = "DEFAULT")
```

Arguments

profile Name of configuration profile (default: DEFAULT).

Value

Object of type cfg_saved

Examples

```
saved_cfg <- data.world::cfg_saved()  
data.world::set_config(saved_cfg)
```

data.world	<i>data.world: The main R package for working with data.world datasets.</i>
------------	---

Description

The data.world package makes it easy to access and work with data.world's datasets.

Configuration

The package can be configured with the [set_config](#) function.

Query features

Use [query](#) to run SPARQL and SQL queries. Use [qry_sql](#) and [qry_sparql](#) to construct query objects and to pass parameters to queries.

Add-ins

Use the included "New insight" add-in to publish plots as project insights from R Studio.

REST API

data.world's REST APIs can be accessed via the [dwapi](#) package. [dwapi](#) is bundled and automatically loaded with data.world. See [dwapi](#) for additional information.

data.world-defunct	<i>Defunct function(s) in the data.world package</i>
--------------------	--

Description

These functions have been removed from the data.world package.

A note on data.world vs. dwapi

Most of these defunct functions are low-level REST API endpoint wrappers and have been renamed for consistency and moved to the [dwapi](#) package in order to keep the data.world namespace as clean as possible.

Details

data.world	is replaced by set_config
query.data.world	is replaced by query
addFile	is now add_file
addFileBySource	is now add_file_by_source
addFilesBySource	is now add_files_by_source
createDataset	is now create_dataset
deleteFileAndSyncSources	is now delete_file
deleteFilesAndSyncSources	is now delete_files
downloadFile	is now download_file
downloadFileAsDataFrame	is now download_file_as_data_frame
getDataset	is now get_dataset
patchDataset	is now update_dataset
replaceDataset	is now replace_dataset
syncDataset	is now sync
uploadDataFrame	is now upload_data_frame
uploadFile	is now upload_file
uploadFiles	is now upload_files
DatasetCreateRequest	is now dataset_create_request
DatasetPatchRequest	is now dataset_update_request
DatasetPutRequest	is now dataset_replace_request
FileBatchUpdateRequest	is now file_batch_update_request
FileCreateOrUpdateRequest	is now file_create_or_update_request

See Also[dwapi](#)

`qry_sparql`*Constructor function for SPARQL queries.*

Description

Constructor function for SPARQL queries.

Usage`qry_sparql(query_string, params = NULL)`**Arguments**

`query_string` SPARQL query string.
`params` Sequence of named query parameters.

Value

Object of type qry_sparql.

Examples

```
## Not run:
sparql_stmt <- data.world::qry_sparql("SELECT ?s ?p ?o
                                     WHERE {
                                       ?s ?p ?o.
                                     }")

## End(Not run)
```

qry_sql

Constructor function for SQL queries.

Description

Constructor function for SQL queries.

Usage

```
qry_sql(query_string, params = NULL)
```

Arguments

query_string SQL query string.
params Sequence of positional query parameters.

Value

Object of type qry_sql.

Examples

```
## Not run:
sql_stmt <- data.world::qry_sql("SELECT * FROM Tables")

## End(Not run)
```

query	<i>Execute a query on data.world.</i>
-------	---------------------------------------

Description

Execute a query on data.world.

Usage

```
query(qry, ...)  
  
## S3 method for class 'qry_sql'  
query(qry, ...)  
  
## S3 method for class 'qry_sparql'  
query(qry, ...)
```

Arguments

qry	Query object of type qry_sql or qry_sparql.
...	S3 method specific params.

Value

Query results as a data frame.

Methods (by class)

- qry_sql: Execute a SQL query on data.world.
- qry_sparql: Execute a SPARQL query on data.world.

See Also

[sql sparql](#)

Examples

```
## Not run:  
sql_stmt <- data.world::qry_sql("SELECT * FROM Tables")  
query_results_df <- data.world::query(  
  sql_stmt, "jonloyens/an-intro-to-dataworld-dataset")  
  
## End(Not run)  
## Not run:  
sparql_stmt <- data.world::qry_sparql("SELECT ?s ?p ?o  
  WHERE {  
    ?s ?p ?o.  
  }")
```

```

query_results_df <- data.world::query(
  sparql_stmt, "jonloyens/an-intro-to-dataworld-dataset")

## End(Not run)

```

save_config	<i>Save configuration to file in the user's home directory.</i>
-------------	---

Description

Save configuration to file in the user's home directory.

Usage

```
save_config(auth_token, ..., profile = "DEFAULT")
```

Arguments

auth_token	API authorization token.
...	Reserved for future configuration parameters.
profile	Configuration profile.

Value

Object of type `cfg_saved`

Examples

```

## Not run:
saved_cfg <- data.world::save_config(auth_token = "YOUR_TOKEN")
data.world::set_config(saved_cfg)

## End(Not run)

```

set_config	<i>Apply configuration to current R session.</i>
------------	--

Description

Apply configuration to current R session.

Usage

```
set_config(cfg)

## Default S3 method:
set_config(cfg)

## S3 method for class 'cfg_env'
set_config(cfg)

## S3 method for class 'cfg_saved'
set_config(cfg)
```

Arguments

cfg Configuration object.

Methods (by class)

- default: Apply configuration using runtime values.
- `cfg_env`: Apply configuration using environment variables.
- `cfg_saved`: Apply configuration using file-based configuration.

See Also

[cfg](#)
[cfg_env](#)
[cfg_saved](#)

Examples

```
data.world::set_config(data.world::cfg_saved())
data.world::set_config(data.world::cfg_env())
data.world::set_config(data.world::cfg("YOUR_TOKEN"))
```

Index

`add_file`, 5
`add_file_by_source`, 5
`add_files_by_source`, 5

`cfg`, 2, 9
`cfg_env`, 3, 9
`cfg_saved`, 3, 8, 9
`create_dataset`, 5

`data.world`, 4
`data.world-defunct`, 4
`data.world-defunct-package`
 (`data.world-defunct`), 4
`data.world-package` (`data.world`), 4
`dataset_create_request`, 5
`dataset_replace_request`, 5
`dataset_update_request`, 5
`delete_file`, 5
`delete_files`, 5
`download_file`, 5
`download_file_as_data_frame`, 5
`dwapi`, 4, 5

`file_batch_update_request`, 5
`file_create_or_update_request`, 5

`get_dataset`, 5

`qry_sparql`, 4, 5
`qry_sql`, 4, 6
`query`, 4, 5, 7
`query.data.world` (`data.world-defunct`), 4

`replace_dataset`, 5

`save_config`, 8
`set_config`, 4, 5, 8
`sparql`, 7
`sql`, 7
`sync`, 5

`update_dataset`, 5
`upload_data_frame`, 5
`upload_file`, 5
`upload_files`, 5