Package 'dpmr'

October 13, 2022

Title Data Package Manager for R

Description Create, install, and summarise data packages that follow the Open Knowledge Foundation's Data Package Protocol.

Version 0.1.9

Date 2016-03-06

URL http://cran.r-project.org/package=dpmr

BugReports https://github.com/christophergandrud/dpmr/issues

License GPL-3

Imports digest, httr, jsonlite, magrittr, rio

Suggests devtools, testthat

RoxygenNote 5.0.1

NeedsCompilation no

Author Christopher Gandrud [aut, cre], Yann-Ael Le Borgne [ctb]

Maintainer Christopher Gandrud <christopher.gandrud@gmail.com>

Repository CRAN

Date/Publication 2016-03-07 08:08:34

R topics documented:

datapackage_info *Return key meta information about the data package*

Description

Return key meta information about the data package

Usage

```
datapackage_info(path, as_list = FALSE)
```

Arguments

path	character string file path to the data package. If empty, then the datapackage.json meta data file is searched for in the working directory. Can also accept a datapackage.json file parsed in R as a list.
as_list	logical indicating whether or not to return the datapackage.json file as a list.

Examples

```
## Not run:
# Print information when working directory is a data package
datapackage_info()

## End(Not run)
```

datapackage_init *Initialise a data package from a data frame, metadata list, and source code file used to create the data set.*

Description

Initialise a data package from a data frame, metadata list, and source code file used to create the data set.

Usage

```
datapackage_init(df, package_name = NULL, output_dir = getwd(),
  meta = NULL, source_cleaner = NULL, source_cleaner_rename = TRUE, ...)
```

Arguments

df	The object name of the data frame you would like to convert into a data package.
package_name	character string name for the data package. Unnecessary if the name field is specified in meta.
output_dir	character string naming the output directory to save the data package into. By default the current working directory is used.
meta	The list object with the data frame's meta data. The list item names must conform to the Open Knowledge Foundation's Data Package Protocol (see http://dataprotocols.org/data-packages/). Must include the name, license, and version fields. If resources is not specified then this will be automatically generated. dpmr uses jsonlite to convert the list into a JSON file. See the toJSON documentation for details. If meta = NULL then a barebones datapackage.json file will be created.
source_cleaner	a character string or vector of file paths relative to the current working directory pointing to the source code file used to gather and clean the df data frame. Can be in R or any other language, e.g. Python. Following Data Package convention the scripts are renamed process*.*, unless specified otherwise with source_cleaner_rename. source_cleaner is not required, but HIGHLY RECOMMENDED.
source_cleaner_rename	logical. Whether or not to rename the source_cleaner files.
...	arguments to pass to export.

Examples

```
## Not run:
# Create fake data
A <- B <- C <- sample(1:20, size = 20, replace = TRUE)
ID <- sort(rep('a', 20))
Data <- data.frame(ID, A, B, C)

# Initialise data package with barebones, automatically generated metadata
datapackage_init(df = Data, package_name = 'my-data-package')

# Initialise with user specified metadata
meta_list <- list(name = 'my-data-package',
                  title = 'A fake data package',
                  last_updated = Sys.Date(),
                  version = '0.1',
                  license = data.frame(type = 'PDDL-1.0',
                           url = 'http://opendatacommons.org/licenses/pddl/'),
                  sources = data.frame(name = 'Fake',
                           web = 'No URL, its fake.'))

 datapackage_init(df = Data, meta = meta_list)

## End(Not run)
```

`datapackage_install` *Install a data package*

Description

Install a data package

Usage

```
datapackage_install(path, load_file, full_meta = FALSE, ...)
```

Arguments

`path`	character string path to the data package directory. Can be a local directory or a URL. If a URL is given the package will be installed in the current working directory. If the file is compressed then it currently must be `.zip`-ped.
`load_file`	character string specifying the path of the data file to load into R. The correct file paths will be printed when the function runs. By default the first file in the datapackage.json path list is loaded. Note: only one file can be loaded at a time.
`full_meta`	logical. Wheter or not to return the full datapackage.json metadata. Note: when `TRUE` only the meta data is returned not the data.
`...`	arguments to pass to `import`.

Examples

```
## Not run:
# Load a data package called gdp stored in the current working directory:
gdp_data = datapackage_install(path = 'gdp')

# Install the gdp data package from GitHub using its .zip URL
URL <- 'https://github.com/datasets/gdp/archive/master.zip'
gdp_data <- datapackage_install(path = URL)

# Install co2 data
library(dplyr)
co2_data <- "https://github.com/datasets/co2-ppm/archive/master.zip" %>%
         datapackage_install()

## End(Not run)
```

`meta_template` *Template for datapackage.json*

Description

Template for datapackage.json

Usage

```
meta_template(df, name, data_paths)
```

Arguments

`df`	The data frame object name of the data frame you would like to convert into a data package.
`name`	character string name of the datapackage.
`data_paths`	character vector of df paths.

Index