

Package ‘metajam’

October 13, 2022

Type Package

Title Easily Download Data and Metadata from 'DataONE'

Version 0.2.3

Date 2020-10-14

Maintainer Julien Brun <brun@nceas.ucsb.edu>

Description A set of tools to foster the development of reproducible analytical workflow by simplifying the download of data and metadata from 'DataONE' (<<https://www.dataone.org>>) and easily importing this information into R.

License Apache License (== 2.0)

URL <https://github.com/nceas/metajam>

BugReports <https://github.com/nceas/metajam/issues>

Imports dataone, dplyr, EML, emld, lubridate, purrr, readr, stats, stringr, tibble, tidyr, rlang

Suggests testthat, knitr, rmarkdown, udunits2

Encoding UTF-8

VignetteBuilder knitr

RoxygenNote 7.1.1

NeedsCompilation no

Author Julien Brun [cre, aut] (<<https://orcid.org/0000-0002-7751-6238>>),
Irene Steves [aut] (<<https://orcid.org/0000-0002-5511-9717>>,
<https://github.com/isteves>),
Mitchell Maier [aut] (<<https://orcid.org/0000-0001-6955-0535>>),
Nathan Hwangbo [ctb] (<<https://orcid.org/0000-0002-3777-0821>>),
Derek Strong [ctb] (<<https://orcid.org/0000-0002-0347-8574>>),
Colin Smith [ctb] (<<https://orcid.org/0000-0003-2261-9931>>),
Kristen Peach [ctb] (<<https://orcid.org/0000-0002-6388-0901>>)

Repository CRAN

Date/Publication 2020-11-03 11:40:02 UTC

R topics documented:

check_version	2
download_d1_data	3
download_d1_data_pkg	4
metajam	4
read_d1_files	5
tabularize_eml	5

Index	7
--------------	----------

check_version	<i>Check PID version</i>
---------------	--------------------------

Description

This function takes an identifier and checks to see if it has been obsoleted.

Usage

```
check_version(pid, formatType = NULL)
```

Arguments

pid	(character) The persistent identifier of a data, metadata, or resource map object on a DataONE member node.
formatType	(character) Optional. The format type to return (one of data, metadata, or resource).

Value

(data.frame) A data frame of object version PIDs and related information.

Examples

```
## Not run:
# Most data URLs and identifiers work
check_version("https://cn.dataone.org/cn/v2/resolve/urn:uuid:a2834e3e-f453-4c2b-8343-99477662b570")
check_version("doi:10.18739/A2ZF6M")

# Specify a formatType (data, metadata, or resource)
check_version("doi:10.18739/A2ZF6M", formatType = "metadata")

# Returns a warning if the identifier has been obsoleted
check_version("doi:10.18739/A2HF7Z", formatType = "metadata")

# Returns an error if no matching identifiers are found
check_version("a_test_pid")

# Returns a warning if several identifiers are returned
```

```
check_version("10.18739/A2057CR99")  
## End(Not run)
```

download_d1_data	<i>Download data and metadata from DataONE</i>
------------------	--

Description

Downloads a data object from DataONE along with metadata.

Usage

```
download_d1_data(data_url, path, dir_name = NULL)
```

Arguments

data_url	(character) An identifier or URL for a DataONE object to download.
path	(character) Path to a directory to download data to.
dir_name	(character) (Optional) Desired name for the folder containing the downloaded data. Defaults to the data file name.

Value

(character) Path where data is downloaded to.

See Also

[read_d1_files\(\)](#) [download_d1_data_pkg\(\)](#)

Examples

```
## Not run:  
soi_moist_path <- download_d1_data(  
  data_url = "urn:uuid:a2834e3e-f453-4c2b-8343-99477662b570",  
  path = tempdir())  
  
download_d1_data(  
  data_url = "https://cn.dataone.org/cn/v2/resolve/urn:uuid:a2834e3e-f453-4c2b-8343-99477662b570",  
  path = tempdir(),  
  dir_name = "test"  
)  
  
## End(Not run)
```

`download_d1_data_pkg` *Download all data and metadata of a data package from DataONE*

Description

Downloads all the data objects of a data package from DataONE along with metadata.

Usage

```
download_d1_data_pkg(meta_obj, path)
```

Arguments

`meta_obj` (character) A DOI or metadata object PID for a DataONE package to download.
`path` (character) Path to a directory to download data to.

Value

(list) Paths where data are downloaded to.

See Also

[read_d1_files\(\)](#) [download_d1_data\(\)](#)

Examples

```
## Not run:  
download_d1_data_pkg("doi:10.18739/A2028W", ".")  
download_d1_data_pkg("https://doi.org/10.18739/A2028W", ".")  
  
## End(Not run)
```

metajam

metajam *package*

Description

Metadata jam - bringing data and metadata together

Details

See [GitHub](#)

read_d1_files	<i>Read data and metadata based on <code>download_d1_data()</code> file structure</i>
---------------	---

Description

Reads data along with metadata into your R environment based on `download_d1_data()` file structure.

Usage

```
read_d1_files(folder_path, fnc = "read_csv", ...)
```

Arguments

`folder_path` (character) Path to a directory where data and metadata are located.
`fnc` (character) Function to be used to read the data (default is `readr::read_csv()`).
... Parameters to pass into the function specified in `fnc`.

Value

(list) Named list containing data and metadata as data frames.

See Also

[download_d1_data\(\)](#) [download_d1_data_pkg\(\)](#)

Examples

```
data_folder <- system.file("extdata", "test_data", package = "metajam")
soil_moist_data <- read_d1_files(data_folder)

# You can specify the function you would like to use to read the file and pass parameters
soil_moist_data_skipped <- read_d1_files(data_folder, "read.csv",
                                       skip = 8, stringsAsFactors = FALSE)
```

tabularize_eml	<i>Get tabular metadata</i>
----------------	-----------------------------

Description

This function takes a path to an EML (.xml) metadata file and returns a data frame.

Usage

```
tabularize_eml(eml, full = FALSE)
```

Arguments

<code>eml</code>	An <code>emld</code> class object, the path to an EML (.xml) metadata file, or a raw EML object.
<code>full</code>	(logical) Returns the most commonly used metadata fields by default. If <code>full = TRUE</code> is specified, the full set of metadata fields are returned.

Value

(data.frame) A data frame of selected EML values.

Examples

```
eml <- system.file("extdata", "test_data", "SoilMois2012_2017__full_metadata.xml",  
                  package = "metajam")  
tabularize_eml(eml)
```

Index

`check_version`, [2](#)

`download_d1_data`, [3](#)

`download_d1_data()`, [4](#), [5](#)

`download_d1_data_pkg`, [4](#)

`download_d1_data_pkg()`, [3](#), [5](#)

`metajam`, [4](#)

`read_d1_files`, [5](#)

`read_d1_files()`, [3](#), [4](#)

`readr::read_csv()`, [5](#)

`tabularize_eml`, [5](#)