Package 'datamgmt'

February 16, 2018

Type Package

Title Data Management Utilities for Curating, Documenting, and Publishing Data

Version 0.1.0

Author Who wrote it

Maintainer The package maintainer <yourself@somewhere.net>

Description More about what it does (maybe more than one line)
Use four spaces when indenting paragraphs within the Description.

License Apache License (== 2.0)

Encoding UTF-8

LazyData true

Suggests testthat

RoxygenNote 6.0.1

R topics documented:

add_creator_id *add_creator_id*

Description

This function allows you to add an ORCID or reference ID to a creator in EML.

Usage

```
add_creator_id(eml, orcid = NULL, id = NULL, surname = NULL)
```

Arguments

eml	EML script to modify
orcid	ORCID in the format 'https://orcid.org/WWWW-XXXX-YYYY-ZZZZ'
id	reference ID. Character string to set reference ID for creators with additional roles (i.e. metadataProvider, etc.)
surname	creator surname (last name), defaults to first creator if not specified. Not case-sensitive.

Details

The function invisibly returns the full EML, which can be saved to a variable. It also prints the changed creator entry so that it's easy to check that the appropriate change was made. All prameters other than the EML are optional, but since the point of the function is to modify either the orcid, ref id, or both, you need to specify at least one. Requires the crayon package.

Examples

```
library(dataone)
library(arcticdatautils)
library(EML)

cnTest <- dataone::CNode('STAGING')
mnTest <- dataone::getMNode(cnTest,'urn:node:mnTestARCTIC')
eml_pid <- arcticdatautils::create_dummy_metadata(mnTest)
eml1 <- EML::read_eml(rawToChar(getObject(mnTest, eml_pid)))
add_creator_id(eml1, orcid = "https://orcid.org/WWWW-XXXX-YYYY-ZZZZ")
```

build_custom_units *Build shiny custom units table*

Description

Build shiny custom units table

Usage

```
build_custom_units(inputdf, standardUnits, inputdf2, unq_unitType, unq_parentSI)
```

Arguments

inputdf	attributes table
standardUnits	standardUnits table
inputdf2	current units table
unq_unitType	unique standardUnits unitTypes
unq_parentSI	unique standardUnits parentSI

build_factors *Build shiny factors table*

Description

Build shiny factors table

Usage

```
build_factors(inputdf, inputdf2, data)
```

Arguments

inputdf	attributes table
inputdf2	current factors table
data	initial data inputed by user

clone_one_package *Clone a Data Package without its child packages.*

Description

The wrapper function 'clone_package' should be used instead. This function copies a data package from one DataOne member node to another, excluding any child data packages.

Usage

```
clone_one_package(mn_pull, mn_push, resource_map_pid)
```

Arguments

mn_pull	(MNode) The Member Node to download from.
mn_push	(MNode) The Member Node to upload to.
resource_map_pid	(chraracter) The identifier of the Resource Map for the package to download.

Value

(list) List of all the identifiers in the new Data Package. TODO switch remaining for loops to applys TODO add more messages TODO better way to set physical in Update Metadata section? TODO pull/push terminology could potentially be confusing. perhaps consider download/upload, from/to, source/new could be better? I do like that they match well (both 4-letter p-words) TODO since messages print in red (scary!), you might want to consider the crayon workaround you found. maybe it's worth having a discussion on our package 'style'?

Author(s)

Dominic Mullen, <dmullen17@gmail.com>

clone_package *Clone a Data Package*

Description

This function copies a Data Package from one DataOne member node to another. It can also be used to copy an older version of a Data Package to the same member node in order to restore it, provided that the old Package is then obsoleted by the copied version.

Usage

```
clone_package(mn_pull, mn_push, resource_map_pid)
```

Arguments

mn_pull	(MNode) The Member Node to download from.
mn_push	(MNode) The Member Node to upload to.
resource_map_pid	(chraracter) The identifier of the Resource Map for the package to download.

Author(s)

Dominic Mullen, <dmullen17@gmail.com>

Examples

```
## Not run:
cn_pull <- CNode("PROD")
mn_pull <- getMNode(cn_pull, "urn:node:ARCTIC")
cn_push <- CNode('STAGING')
mn_push <- getMNode(cn_push,'urn:node:mnTestARCTIC')
clone_package(mn_pull, mn_push, "resource_map_doi:10.18739/A2RZ6X")

## End(Not run)
```

---

compare_eml_to_package_pids

Return data object identifiers from 'dataTable' and 'otherEntity' objects in an EML file

---

Description

This function is a helper function for 'clone_package'. It checks that all of the data objects pids present in the metadata match those that the resource map points to. It also returns the data pids in the order in which they appear in the EML: dataTable pids first and then otherEntity pids. We cannot update the new .xml file without this information.

Usage

```
compare_eml_to_package_pids(eml, package_data_pids)
```

Arguments

eml	(EML) EML object
package_data_pids	(character) All of the Data identifiers in a Data Package. These can be selected with get_package(mn, resource_map)$data

Value

(list) List of all the identifiers in the EML. Sorted into dataTable and otherEntity identifiers TODO better name for this function?

Author(s)

Dominic Mullen, <dmullen17@gmail.com>

`create_attributes_table` *Allows editing of an attribute table and custom units table in a shiny environment*

Description

Allows editing of an attribute table and custom units table in a shiny environment

Usage

```
create_attributes_table(data = NULL, attributes_table = NULL)
```

Arguments

`data`	The data.frame of data that needs an attribute table
`attributes_table`	A existing attributes table for `data` that needs to be updated. If specified, all non empty fields will be used (i.e. if numberType is specified in `attributes_table`, then this function will use those values instead of automatically generating values from `data`).

Examples

```
create_attributes_table(NULL, NULL)

data <- read.csv("Test.csv")
create_attributes_table(data, NULL)

attributes_table <- EML::get_attributes(eml@dataset@dataTable[[i]]@attributeList)$attributes
create_attributes_table(NULL, attributes_table)

create_attributes_table(data, attributes_table)
```

`data_objects_exist` *Check if data objects exist in a list of Data Packages.*

Description

This function is primarily intended to assist Mark Schildhauer in preparation for the Carbon synthesis working group.

Usage

```
data_objects_exist(mn, pids, write_to_csv = FALSE, folder_path = NULL,
  file_name = NULL)
```

Arguments

`mn`	(MNode/CNode) The Node to query for Object sizes
`pids`	(character) The identifier of the Data Packages' Metadata
`write_to_csv`	(logical) Optional. Write the query results to a csv
`folder_path`	(character) Optional. Folder to write results to
`file_name`	(character) Optional. Name of results file in csv format

Value

(data.frame) Data frame containing query results.

Author(s)

Dominic Mullen, <dmullen17@gmail.com>

Examples

```
## Not run:
cn <- CNode("PROD")
mn <- getMNode(cn, "urn:node:ARCTIC")
data_objects_exist(mn,
c("doi:10.5065/D60P0X4S", "urn:uuid:3ea5629f-a10e-47eb-b5ce-de10f8ef325b"))

## End(Not run)
```

excel_to_csv *Convert excel workbook to multiple csv files*

Description

Converts an excel workbook into multiple csv files (one per tab). Names the files in the following format: sheetName_excelName.csv.

Usage

```
excel_to_csv(path, directory = NULL, ...)
```

Arguments

path	(character) File location of the excel workbook.
directory	(character) Optional. Directory to download csv files to. Defaults to the base directory that path is located in.

Value

(invisible())

Author(s)

Dominic Mullen <dmullen17@gmail.com>

factor_att_table *Creates factors for attributes table*

Description

Creates factors for attributes table

Usage

```
factor_att_table(inputdf)
```

Arguments

inputdf	attributes table

get_eml_pids *Return data identifiers from within an EML*

Description

This function returns data object identifers present within an EML (electronic metadata language) document.

Usage

```
get_eml_pids(eml)
```

Arguments

eml	() an EML object

Value

Returns a list of data object pids in the eml

get_numberType *Returns the numberType (either "real", "integer", "whole", or "natural") of input values*

Description

Returns the numberType (either "real", "integer", "whole", or "natural") of input values

Usage

```
get_numberType(values)
```

Arguments

values	A vector of values. If vector is non-numeric will return NA

Value

The numberType of `values` (either "real", "integer", "whole", or "natural").

Examples

```
# To get numberType for each column in a data.frame \code{df}:
unlist(lapply(df, function(x) get_numberType(x)))
```

get_object_metadata

Return the object metadata of each data object in a Data Package

Description

This function returns the formatId, fileName and identifier of each data object in a Data Package. This is a helper function for the function 'clone_package' - which copies a dataOne Data Package from one member node to another.

Usage

```
get_object_metadata(mn, resource_map_pid, formatType = "DATA")
```

Arguments

mn	(MNode/CNode) The Node to query for Object sizes
resource_map_pid	(character) The identifier of the Data Package's Resource Map
formatType	(character) Optional. Filter to just Objects of the given formatType. One of METADATA, RESOURCE, or DATA or * for all types

Value

(character) The formatId, fileName, and identifier of each data object in a package.

Author(s)

Dominic Mullen, <dmullen17@gmail.com>

hello

Hello, World!

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

`output_text_func` *Outputs data.frame to text for shiny app*

Description

Outputs data.frame to text for shiny app

Usage

```
output_text_func(df)
```

Arguments

`df`	data.frame

`shiny_attributes_table` *Build shiny UI for editing attributes table within function create_attributes_table()*

Description

Build shiny UI for editing attributes table within function create_attributes_table()

Usage

```
shiny_attributes_table(att_table, data)
```

Arguments

`att_table`	an attribute table built from create_attributes_table()