

Package: bdDwC (via r-universe)

September 1, 2024

Version 0.1.15

Date 2018-08-28

Title Darwinizer: Darwin Core (DwC) Field Names Standardization

Description The 'shiny' application 'bdDwC' makes biodiversity data field names Darwin Core compatible.

Maintainer Povilas Gibas <povilasgibas@gmail.com>

Depends R (>= 2.10)

Imports shiny, shinyBS, shinydashboard, shinyjs, shinyFiles

LazyData true

RoxygenNote 6.1.0

License GPL-3

Encoding UTF-8

Suggests testthat, covr

NeedsCompilation no

Author Povilas Gibas [aut, cre]

(<<https://orcid.org/0000-0001-5311-6021>>), Tomer Gueta [aut]

(<<https://orcid.org/0000-0003-1557-8596>>), Vijay Barve [aut]

(<<https://orcid.org/0000-0002-4852-2567>>), Thiloshon Nagarajah

[aut], Yohay Carmel [aut]

Date/Publication 2018-08-31 19:40:03 UTC

Repository <https://pogibas.r-universe.dev>

RemoteUrl <https://github.com/cran/bdDwC>

RemoteRef HEAD

RemoteSha c8aaedaeb982797cc0f8b47dd714e775df3c9a6

Contents

combineOldNew	2
darwinizeNames	2
dataDarwinCloud	3

dataReptiles	3
downloadCloudData	4
getDarwinCoreInfo	4
renameUserData	5
runDwC	6

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

combineOldNew	<i>Combine old/new name for checkboxes</i>
---------------	--

Description

‘combineOldNew()’ is a function that combines (‘paste’) fieldname and standard names with uni-code characters to present in checkboxes.

Usage

```
combineOldNew(data, symbolArrow = "->", symbolSpace = " ")
```

Arguments

data	data.frame of matched names (‘darwinizeNames()’ output)
symbolArrow	character string for symbol that is used in ‘paste0’ to connect old and new name
symbolSpace	character string for symbol used instead of white space

Value

data.frame of darwinized user names.

darwinizeNames	<i>Darwinize names</i>
----------------	------------------------

Description

‘darwinizeNames()’ is a function to darwinize given names using reference (ie, Darwin Cloud) dataset. It returns \$Old\$ and \$New\$ (\$fieldname\$ and \$standard\$) name (ie, name from given user dataset that had a match in reference dataset).

Usage

```
darwinizeNames(dataUser, dataDWC)
```

Arguments

dataUser	data.frame with user data
dataDWC	data.frame with Darwin Cloud data

Value

data.frame of darwinized user names.

Examples

```
darwinizeNames(dataUser = bdDwC:::dataReptiles,  
               dataDWC  = bdDwC:::dataDarwinCloud$data)
```

dataDarwinCloud	<i>Darwin Cloud Data</i>
-----------------	--------------------------

Description

Darwin Cloud data used to Darwinize. Darwin Cloud data was retrieved using ‘bdDwC::downloadCloudData()’ function. This dataset contains Darwin Cloud data and download date.

Usage

```
dataDarwinCloud
```

Format

A list of length two. First entry contains a data frame with 422 rows and 2 columns. Second entry contains a ‘Date’ class object.

dataReptiles	<i>Indian Reptiles</i>
--------------	------------------------

Description

Dataset that contains subsample of Indian reptile observations from the iNaturalist platform.

Usage

```
dataReptiles
```

Format

A data frame with 50 rows and 41 columns.

downloadCloudData *Download Darwin Cloud Data*

Description

'downloadCloudData()' is a function used to download Darwin Cloud data. First it downloads data, then it subsets informative columns and renames them so they could be used in 'shiny' app.

Usage

```
downloadCloudData(pathRemote = "https://raw.githubusercontent.com/kurator-org",
  pathGithub = "/kurator-validation/master/packages/kurator_dwca/data/vocabularies/",
  pathFile = "darwin_cloud.txt", columnField = "fieldname",
  columnStand = "standard")
```

Arguments

pathRemote	Path to remote repository (Kurator in Github).
pathGithub	Path within given repository.
pathFile	Name of a file.
columnField	Name of the column that contains field information.
columnStand	Name of the column that contains standard information.

Value

data.frame of Darwin Cloud data.

Examples

```
downloadCloudData()
```

getDarwinCoreInfo *Retrieve Information about Darwin Core Terms*

Description

'getDarwinCoreInfo()' is a function (not exported) to download Darwin Core Terms. This information is displayed when using manually renaming.

Usage

```
getDarwinCoreInfo(pathDarwinCloud = "http://tdwg.github.io/dwc/terms/")
```

Arguments

pathDarwinCloud
Path to Darwin Cloud data.

Value

data.frame that contains name and definition for each available Darwin Cloud term.

renameUserData	<i>Rename Dataset According Darwinized Names</i>
----------------	--

Description

'renameUserData()' is a function used to rename given dataset (usually given user data) according previously Darwinized Names.

Usage

```
renameUserData(dataUser, dataRenamed)
```

Arguments

dataUser data.frame of be renamed.
dataRenamed data.frame that was created using 'bdDwC:::darwinizeNames()' function using same 'dataUser'.

Value

data.frame of submitted user data, but renamed according given renames data.frame.

Examples

```
result <- darwinizeNames(dataUser = bdDwC:::dataReptiles,  
                        dataDWC = bdDwC:::dataDarwinCloud$data)  
renameUserData(bdDwC:::dataReptiles, result)
```

`runDwC`*Launch bdDwC Shiny Application*

Description

'bdDwC' is a function that starts bdverse Darwin Cloud cleaning 'shiny' app.

Usage

```
runDwC()
```

Value

'shiny::runApp()' result within browser.

Index

* datasets

- dataDarwinCloud, 3
- dataReptiles, 3

combineOldNew, 2

darwinizeNames, 2
dataDarwinCloud, 3
dataReptiles, 3
downloadCloudData, 4

getDarwinCoreInfo, 4

renameUserData, 5
runDwC, 6