

Package ‘rbefdata’

February 20, 2015

Title BEFdata R package

Version 0.3.5

Description Basic R package to access data structures offered by any BEFdata portal instance.

Suggests ggplot2, plyr

Imports XML, RCurl, rjson, wordcloud, rtematres, RColorBrewer

Date 2013-04-17

License MIT + file LICENSE

URL <https://github.com/befdata/rbefdata>

Collate 'bef.options.R' 'zzz.R' 'rbefdata-package.R'
'bef.portal.get.dataset.R' 'bef.portal.get.proposal.R'
'bef.portal.get.metadata.R' 'bef.portal.get.keywords.R'
'bef.portal.get.categories.R' 'bef.portal.upload.R'
'bef.portal.api.R' 'bef.mediawiki.api.R' 'bef.tematres.api.R'
'helper.R' 'bef.portal.get.attachments.R'
'rbefdata-deprecated.R' 'rbefdata-defunct.R'
'bef.portal.vizalize.R'

Author Claas-Thido Pfaff [cre, aut],
Karin Nadrowski [aut],
Xingxing Man [aut]

Maintainer Claas-Thido Pfaff <claas-thido.pfaff@idiv-biodiversity.de>

NeedsCompilation no

Repository CRAN

Date/Publication 2013-11-18 17:50:08

R topics documented:

rbefdata-package	2
bef.mediawiki.api.define	2
bef.options	3
bef.portal.api.dataset_info	4

bef.portal.api.proposal_info	4
bef.portal.attach.to_dataset	5
bef.portal.attach.to_proposal	5
bef.portal.get.attachments	6
bef.portal.get.categories_for	7
bef.portal.get.dataset	8
bef.portal.get.datasets.for_keyword	9
bef.portal.get.datasets.for_proposal	9
bef.portal.get.keywords	10
bef.portal.get.metadata	11
bef.portal.update.dataset	11
bef.portal.upload.categories	12
bef.portal.upload.dataset	13
bef.portal.vizualize.keywords	14
bef.tematres.api	14

Index	15
--------------	-----------

rbefdata-package	<i>A package to fetch datasets and metadata information from a BEFdata portal</i>
------------------	---

Description

The package is by default associated to the websites of the BEF-China experiment: <http://china.befdata.biow.uni-leipzig.de>.

Details

If you use the package with your own instance of a BEFdata portal you need to change the default URLs to your needs via `bef.options()`.

bef.mediawiki.api.define	<i>Basic Mediawik API with one task to retrieve descriptions</i>
--------------------------	--

Description

This function queries Wikipedia in the language of your choice. You can select the language per two character country code Wikipedia uses in they sub domains.

Usage

```
bef.mediawiki.api.define(term, language = "en")
```

Arguments

term	Is the term or string you are searching for on Wikipedia.
language	Is the language of the Wikipedia (default: en). You can use two character country codes that Wikipedia uses in their sub domains.

Value

The function returns a vector with description strings.

Examples

```
## Not run:
descriptions = bef.mediawiki.api.define(term = "Tree")
descriptions = bef.mediawiki.api.define(term = "Baum", language = "de")

## End(Not run)
```

bef.options *Set or query options related to the BEFdata R package.*

Description

This function is used to query and set R BEFdata package specific options like the URL to the BEFdata portal and the URL to the tematres thesaurus.

Usage

```
bef.options(...)
```

Arguments

... similar to [options](#). see examples below.

Examples

```
# BEFdata URL
bef.options('url')
bef.options(url='http://www.example.com')
#Tematres URL
bef.options('tematres_url')
bef.options(tematres_url="http://www.example.com")
```

bef.portal.api.dataset_info

Basic access to the BEFdata portal XML API.

Description

This function gives access to the meta data available for a dataset

Usage

```
bef.portal.api.dataset_info(id)
```

Arguments

id Is the ID of the dataset you like to get the information for. You find the ID in the URL of the dataset on the BEFdata portal.

Details

This function takes a dataset id and returns the metadata of the dataset as list. This is not the eml metadata. It is the metadata read from the XML API of the data portal.

Value

Returns a list of metadata information for a dataset.

bef.portal.api.proposal_info

This function returns the information available for a paper proposal.

Description

This function returns the information available for a paper proposal.

Usage

```
bef.portal.api.proposal_info(id)
```

Arguments

id Is the ID of the paperproposal you like to get information for. You can find the ID in the URL on the paper proposal.

Value

Returns a list of metadata information for the proposal.

`bef.portal.attach.to_dataset`*Attach a freeformat file to a dataset*

Description

This function attaches a freeformat file to a dataset.

Usage

```
bef.portal.attach.to_dataset(id, attachment,  
  description = "", open_browser = FALSE,  
  curl = getCurlHandle())
```

Arguments

<code>id</code>	The ID of the dataset you like to attach to. You can get the ID from the URL of the dataset.
<code>attachment</code>	Can be either a path to a file or an R data frame.
<code>description</code>	An optional description to your attached freeformat file.
<code>curl</code>	You can pass in a curl handle to reduce memory footprint and to add own options
<code>open_browser</code>	If this is set to true the page of the dataset is opened in the browser after successful upload. This defaults to FALSE.

Value

Returns a status message whether the update was successful or not.

Examples

```
## Not run:  
  bef.portal.attach.to_dataset(id = 72, attachment = dataset, description = "This  
    is a pure test description for the new attachment file", open_browser = T)  
## End(Not run)
```

`bef.portal.attach.to_proposal`*Attach a freeformat file to a paper proposal*

Description

This function attaches a freeformat file to a proposal

Usage

```
bef.portal.attach.to_proposal(id, attachment,
  description = "", is_paper = FALSE, doi = "",
  open_browser = FALSE, curl = getCurlHandle())
```

Arguments

id	The ID of the proposal you like to attach to. You can get the ID from the URL of the proposal.
attachment	Can be either a path to a file or an R data frame.
description	An optional description to your attached freeformat file.
curl	You can pass in a curl handle to reduce memory footprint and to add own options
is_paper	Just to tell the portal if it is a published paper you are attaching as pdf for example. This parameter can be TRUE or FALSE and defaults to FALSE.
doi	If the attachment has a persistent digital identifier you can add it here.
open_browser	If this is set to true the page of the proposal is opened in the browser after successful upload. This defaults to FALSE.

Value

Returns a status message whether the update was successful or not.

Examples

```
## Not run:
  bef.portal.attach.to_proposal(id = 72, attachment = dataset, description = "This
    is a pure test description for the new attachment file", open_browser = T)

## End(Not run)
```

```
bef.portal.get.attachments
```

Download attachment files of a dataset in BEFdata portal

Description

This function will download attachment files associated with a dataset into specified directory.

Usage

```
bef.portal.get.attachments(dataset,
  directory = bef.options("download_dir"),
  curl = getCurlHandle(), ...)
```

Arguments

dataset	The ID of dataset you like to fetch the attachments from. You find the ID on the dataset page on the BEFdata portal.
directory	The directory to store attachment files to. By default it creates a folder called "downloads" under the current working directory. The default can be changed by bef.options.
curl	If using in a loop, call getCurlHandle() first and pass the returned value in here (avoids unnecessary footprint).
...	arguments passed to getURLContent .

Value

A data frame of file information is returned invisibly. NULL is returned when the dataset has no attachment files.

```
bef.portal.get.categories_for
```

Get a data groups categories as data frame (data admin interface)

Description

This function gets you a data groups categories as data frame. You can use the data frame to harmonize your data using the last column and the IDs in the first row. After you have relinked the data you can upload the data frame using the categories upload command.

Usage

```
bef.portal.get.categories_for(datagroup_id,
  curl = getCurlHandle(), ...)
```

Arguments

datagroup_id	It the ID of the data group you like to get the categories from.
curl	Pass in a curl handle here if you like with own options or to reduce the memory footprint.
...	Additional parameters passed to curl.

Value

Returns a data frame with the categories ID, short and long description and

Examples

```
## Not run:
  bef.portal.get.categories_for(datagroup_id = 22)

## End(Not run)
```

`bef.portal.get.dataset`*Fetch primary data from a BEFdata portal dataset.*

Description

This function fetches data associated with a BEFdata portal dataset. By default it will fetch the CSV file of a dataset. You need to provide the function with a dataset id which you can find in the URL of the dataset on the BEFdata portal. As this usually requires authentication you need to set your credentials in the options (`bef.options("user_credentials" = "asdfpoj")`). You can find the credentials inside of your profile page on the BEFdata portal. The credentials ensure you have the rights to download the data.

Usage

```
bef.portal.get.dataset(id, curl = getCurlHandle(), ...)
```

Arguments

<code>id</code>	This is the ID of a dataset on a BEFdata portal.
<code>curl</code>	If the function is used inside a loop, call <code>getCurlHandle()</code> first and pass in the returned value here. This avoids an unnecessary footprint.
<code>...</code>	Arguments passed to getURLContent

Details

The function returns a dataset object which you can store in a variable as shown in the examples below. The object also offers additional information by attributes. You can query the information via the `attributes()` function which is also shown in the examples. If you like to fetch multiple datasets you can use the apply functions provided by R see example below.

Value

The function returns a data frame of the dataset. An error is thrown when the dataset is not found or if you don't have the rights to access it.

Examples

```
## Not run:
  dataset1 = bef.portal.get.dataset(id=8)
  metadata1 = attributes(dataset1)

  curl = getCurlHandle()
  ids = c(8,70)
  dataset_list = lapply(ids, function(x) bef.portal.get.dataset_by(id = x, curl = curl))
  metadata = attributes(dataset_list[[1]])

## End(Not run)
```

```
bef.portal.get.datasets.for_keyword
```

Fetch a list of datasets for a keyword

Description

This function fetches a list of datasets associated with a BEFdata portal keyword.

Usage

```
bef.portal.get.datasets.for_keyword(keyword)
```

Arguments

keyword The keyword you like to fetch the associated datasets for

Examples

```
## Not run:
  list = bef.portal.get.datasets.for_keyword(keyword = "carbon")

## End(Not run)
```

```
bef.portal.get.datasets.for_proposal
```

Fetch primary data in CSV format from a BEFdata portal paperproposal

Description

This function fetches data associated to a paperproposal on a BEFdata portal. By default it will fetch all associated files in CSV format next to the informations available like the title, the download URL and the id of the datasets.

Usage

```
bef.portal.get.datasets.for_proposal(id,
  curl = getCurlHandle(), ...)
```

Arguments

id This is the ID of a paper proposal. You can download all datasets in one turn given the ID.

curl If using in a loop, call `getCurlHandle()` first and pass the returned value in here (avoids unnecessary footprint)

... This are other arguments passed to [getURLContent](#)

Details

You need to provide the function with a proposal id which you can find in the URL of the proposal and your user credentials. You can find the credentials inside of your profile page on the BEFdata portal. The credentials ensure you have the rights to download the data.

The function returns a list object which you can store to a variable as shown in the examples below.

Value

The function returns a list of raw data attached to a proposal. An error is thrown when the proposal is not found or you don't have the access rights for it.

Examples

```
## Not run:
prop1 = bef.portal.get.datasets.for_proposal(proposal = 8)

## End(Not run)
```

```
bef.portal.get.keywords
```

Fetch keywords from a BEFdata portal.

Description

This function fetches keywords from a BEFdata portal.

Usage

```
bef.portal.get.keywords(curl = getCurlHandle(), ...)
```

Arguments

<code>curl</code>	You can pass in a curl handle with additional options. By default a curl handle is used to improve the memory footprint.
<code>...</code>	Arguments passed to getURLContent

Value

The function returns a data frame of keyword (id, name, count).

Examples

```
## Not run:
keywords=bef.portal.get.keywords()

## End(Not run)
```

```
bef.portal.get.metadata
```

Get the metadata of a dataset from the BEFdata portal

Description

This function fetches the metadata associated with a dataset on a BEFdata portal. You need to provide the function with an id of the dataset or a path to an eml file you downloaded from the portal manually. You can find the ID in the URL of the dataset page in the portal (e.g ID 6: ../datasets/6).

Usage

```
bef.portal.get.metadata(dataset)
```

Arguments

`dataset` This is either a dataset ID or a file path to a eml metadata file.

Value

A list of metadata. Metadata That doesn't exist is represented as NA

```
bef.portal.update.dataset
```

Update a dataset already on BEFdata portal

Description

This function updates a dataset from within R. Before you can do so you need to have an account on the portal and set your credentials via the `bef.options()` function so the authentication works. The function requires an ID of the dataset and either a path to a CSV file or a data frame used to update the old dataset.

Usage

```
bef.portal.update.dataset(dataset, dataset_id,  
  warn = TRUE, open_browser = F, curl = getCurHandle())
```

Arguments

`dataset` The data you like to use for upload and to update the old dataset. This can be a either a path to a CSV file or a data frame.

`dataset_id` The ID of the dataset you like to update. You can get the ID from the URL of the dataset you like to update.

warn	Is a warning that should prevent overriding dataset accidentally. Set this to FALSE if you know what you do.
curl	You can pass in a curl handle to reduce memory footprint and to add own options
open_browser	If this is set to true the page of the dataset is opened in the browser after successful upload. This defaults to FALSE.

Details

Please be careful as this function overrides the old dataset (raw data) on the portal completely.

Value

Returns a status message whether the update was successful or not.

Examples

```
## Not run:
dataset = data.frame(a = c(1,2,3,4), b = c("a","b","c","d"))
bef.portal.update.dataset(dataset = dataset, dataset_id = 72)

## End(Not run)
```

```
bef.portal.upload.categories
      Upload a category set
```

Description

This function is thought for data admin of a BEFdata portal. It uses the data harmonization features provided by the BEFdata portal. In order to do harmonize a datagroup you download the category with the command below: `bef.portal.get.categories_for(datagroups_id = 22)`

Usage

```
bef.portal.upload.categories(datagroup_id, categories,
                             curl = getCurlHandle())
```

Arguments

datagroup_id	Categories ID you like to harmonize. You can find this in the URL of the category.
categories	Either a data frame with the categories or a csv containing the categories and merge ID and stuff downloaded from the portal.
curl	Pass in a curl handle with own options or to reduce memory footprint.

Value

Returns a status message whether the upload was successful or not.

Examples

```
## Not run:
  dataframe = my_categories_df
  bef.portal.upload.categories(categories_id = 22, categories = dataframe)

## End(Not run)
```

```
bef.portal.upload.dataset
```

Upload a dataset to the BEFdata portal

Description

This function uploads a dataset from within R to a BEFdata portal set in your options. Before you can do so you need to have an account on the portal and set your credentials via the `bef.options()` function so the authentication works. The function requires a title for the dataset and either a path to a CSV file for for upload or a dataset object. The dataset must have unique column names.

Usage

```
bef.portal.upload.dataset(dataset, dataset_title,
  curl = getCurlHandle(), open_browser = F)
```

Arguments

<code>dataset</code>	The data you like to upload as dataset. This can be a path to a CSV file or a data frame. Please not that you have to ensure that the column names are unique.
<code>dataset_title</code>	The title of the dataset.
<code>curl</code>	You can pass in a curl handle with additional options. By default a curl handle is used to improve the memory footprint.
<code>open_browser</code>	If this is set to true the page of the dataset is opened in the browser after successful upload. This defaults to FALSE.

Value

Returns a status message with the ID of the dataset.

Examples

```
## Not run:
  dataset = data.frame(a = c(1,2,3,4), b = c("a","b","c","d"))
  bef.portal.upload.dataset(dataset = dataset, dataset_title = "Testtitle", open_browser = T)

## End(Not run)
```

```
bef.portal.vizualize.keywords
```

Vizualization functions to your BEFdata portal

Description

Vizalize the keywords contained in your BEFdata portal vai a nice wordcloud.

Usage

```
bef.portal.vizualize.keywords(color_pal = brewer.pal(8, "Dark2"),
  min_freq = 1)
```

Arguments

color_pal	Is a color palette you can create for example with the color brewer package. See in the examples below. This defaults to brewer.pal(8,"Dark2").
min_freq	I passed to the wordcloud function and delimits the minimum frequency the key-word needs to appear in the cloud. This defaults to 1.

Value

The function returns a wordcloud based on the keywords of the BEFdata portal.

Examples

```
## Not run:
  require(RColorBrewer)
  my_palette = brewer.pal(8,"Dark2")
  bef.portal.vizualize.keywords()

## End(Not run)
```

```
bef.tematres.api
```

Basic tematres API via rtematres

Description

This function queries a tematres thesaurus via the rtematres api package.

Usage

```
bef.tematres.api(...)
```

Arguments

... Any task or term to be passed to the rtematres package

Index

bef.attach.to_dataset
 (bef.portal.attach.to_dataset),
 5

bef.attach.to_proposal
 (bef.portal.attach.to_proposal),
 5

bef.get.attachments
 (bef.portal.get.attachments), 6

bef.get.attachments_for
 (bef.portal.get.attachments), 6

bef.get.categories_for
 (bef.portal.get.categories_for),
 7

bef.get.dataset
 (bef.portal.get.dataset), 8

bef.get.dataset_by
 (bef.portal.get.dataset), 8

bef.get.datasets.for_keyword
 (bef.portal.get.datasets.for_keyword),
 9

bef.get.datasets.for_proposal
 (bef.portal.get.datasets.for_proposal),
 9

bef.get.keywords
 (bef.portal.get.keywords), 10

bef.get.metadata
 (bef.portal.get.metadata), 11

bef.get.metadata_for
 (bef.portal.get.metadata), 11

bef.mediawiki.api.define, 2

bef.options, 3

bef.portal.api.dataset_info, 4

bef.portal.api.proposal_info, 4

bef.portal.attach.to_dataset, 5

bef.portal.attach.to_proposal, 5

bef.portal.get.attachments, 6

bef.portal.get.attachments_for
 (bef.portal.get.attachments), 6

bef.portal.get.categories_for, 7

bef.portal.get.dataset, 8

bef.portal.get.dataset_by
 (bef.portal.get.dataset), 8

bef.portal.get.datasets.for_keyword, 9

bef.portal.get.datasets.for_proposal,
 9

bef.portal.get.keywords, 10

bef.portal.get.metadata, 11

bef.portal.get.metadata_for
 (bef.portal.get.metadata), 11

bef.portal.update.dataset, 11

bef.portal.upload.categories, 12

bef.portal.upload.dataset, 13

bef.portal.vizualize.keywords, 14

bef.tematres.api, 14

bef.update.dataset
 (bef.portal.update.dataset), 11

bef.upload.categories
 (bef.portal.upload.categories),
 12

bef.upload.dataset
 (bef.portal.upload.dataset), 13

bef.vizualize.keywords
 (bef.portal.vizualize.keywords),
 14

getURLContent, 7–10

options, 3

rbefdata (rbefdata-package), 2

rbefdata-package, 2