

Package ‘BerlinData’

July 2, 2014

Type Package

Title Easy access to Berlin related data

Version 1.0.0

Date 2014-03-14

Author Kate McCurdy, Dirk Schumacher

Maintainer Dirk Schumacher <mail@dirk-schumacher.net>

Description This package provides easy access to <http://daten.berlin.de>
It allows you to search through the data catalogue and to download
the data directly from within R.

License MIT + file LICENSE

URL <http://dirkschumacher.github.io/RBerlinData>

Depends R (>= 2.13.0)

Imports XML (>= 3.95), stringr

Suggests testthat, rjson

NeedsCompilation no

Repository CRAN

Date/Publication 2014-05-11 16:02:39

R topics documented:

cached_datasets_feed	2
datasets_feed_tree	2
download	3
download.berlin_data_resource	4
download.CSV	4
download.default	5

download.JSON	5
download.TXT	6
download.XML	6
getDatasetMetaData	7
parseMetaData	8
resources	8
resources.berlin_data_dataset	8
searchBerlinDatasets	9
updateCachedFeed	9

Index 10

cached_datasets_feed	<i>Cached version of daten.berlin.de dataset feed. Stored as XML tree in text form</i>
----------------------	--

Description

Cached version of daten.berlin.de dataset feed. Stored as XML tree in text form

Format

XML tree with timestamp attribute 'last_updated'

Source

<http://daten.berlin.de/datensaetze/rss.xml>

datasets_feed_tree	<i>Cached version of daten.berlin.de dataset feed. Stored as XML tree in text form</i>
--------------------	--

Description

Cached version of daten.berlin.de dataset feed. Stored as XML tree in text form

Format

XML tree with timestamp attribute 'last_updated'

Source

<http://daten.berlin.de/datensaetze/rss.xml>

download	<i>Downloads a resource</i>
----------	-----------------------------

Description

Downloads a resource

Usage

```
download(x, ...)
```

Arguments

x	a resource, or object containing resources, which can be downloaded
...	optional additional arguments to methods, e.g.: <ul style="list-style-type: none">• <code>message.on.faillogical</code>: show message on download failure?• <code>message.on.succeedlogical</code>: show message on download success?

Examples

```
# query, select a dataset, get its metadata
## Not run:
result <- searchBerlinDatasets(query = "vornamen")
summary(result)
dataset <- getDatasetMetaData(result[[2]])
summary(dataset)

# pick a resource to download
resource <- resources(dataset)[[23]]
data <- download(resource)
# returns a data.frame
class(data)
# or download multiple resources simultaneously
data <- download(resources(dataset)[1:6])
# returns a list of data.frames
class(data)
class(data[[1]])

# download all resources in dataset
data <- download(dataset)

# turn off individual notifications for failed downloads
# due to unsupported file formats, URL schemes, etc.
data <- download(dataset, message.on.fail=FALSE)

# turn off all individual notifications for downloads
data <- download(dataset, message.on.fail=FALSE, message.on.succeed=FALSE)

# pass in other arguments to download function
```

```

result <- searchBerlinDatasets(query = "stolpersteine")
dataset <- getDatasetMetaData(result[[2]])
resource_list <- resources(dataset)
data <- download(resource_list[[1]])
# gives wrong output, so we try a different argument for 'sep'
data <- download(resource_list[[1]], sep=',')

## End(Not run)

```

```

download.berlin_data_resource
      Downloads a resource

```

Description

Downloads a resource

Usage

```

## S3 method for class 'berlin_data_resource'
download(x, message.on.fail = TRUE,
        message.on.succeed = TRUE, ...)

```

Arguments

x	a resource of type berlin_data_resource
message.on.fail	logical: show message on download failure?
message.on.succeed	logical: show message on successful download?
...	optional additional arguments to download function

```

download.CSV      Downloads CSV

```

Description

Downloads CSV

Usage

```

## S3 method for class 'CSV'
download(x, message.on.fail = TRUE, ..., sep = ";")

```

Arguments

x url of csv file location
 message.on.fail logical: show message on download failure?
 sep field separator
 ... optional additional arguments to download function

download.default *Default fallback method for download Indicates unsupported file format*

Description

Default fallback method for download Indicates unsupported file format

Usage

```
## Default S3 method:
download(x, message.on.fail = TRUE, ...)
```

Arguments

x a resource which can be downloaded
 ... optional additional arguments
 message.on.fail logical: show message indicating failure to download?

download.JSON *Downloads json file*

Description

Downloads json file

Usage

```
## S3 method for class 'JSON'
download(x, message.on.fail = TRUE, ..., parse.to.df = TRUE)
```

Arguments

x url of json file location
 message.on.fail logical: show message on download failure?
 parse.to.df logical: should the function try to parse the JSON output into a data.frame?
 ... optional additional arguments to download function

download.TXT	<i>Downloads TXT</i>
--------------	----------------------

Description

Downloads TXT

Usage

```
## S3 method for class 'TXT'
download(x, message.on.fail = TRUE, ..., sep = ",")
```

Arguments

x	url of txt file location
sep	field separator
message.on.fail	logical: show message on download failure?
...	optional additional arguments to download function

download.XML	<i>Downloads xml file</i>
--------------	---------------------------

Description

Downloads xml file

Usage

```
## S3 method for class 'XML'
download(x, message.on.fail = TRUE, ..., parse.to.df = TRUE)
```

Arguments

x	url of xml file location
message.on.fail	logical: show message on download failure?
parse.to.df	logical: should the function try to parse the XML output into a data.frame?
...	optional additional arguments to download function

getDatasetMetaData *Gets metadata for a dataset or list of datasets*

Description

Gets metadata for a dataset or list of datasets

Usage

```
getDatasetMetaData(where, ...)
```

Arguments

where	where to look for metadata
...	optional additional arguments to methods

Examples

```
# query
## Not run:
result <- searchBerlinDatasets(query = "vornamen")
summary(result)

# options to get metadata:
# pass in URL
metadata <- getDatasetMetaData(result[[2]]$link)
# returns 'berlin_data_dataset' object with list of available resources
class(metadata)

# pass in object
metadata <- getDatasetMetaData(result[[2]])
# same result as passing in link

# pass in berlin_data_list with query results to get metadata
# for all objects in list
metadata <- getDatasetMetaData(result)
summary(metadata)
summary(metadata[[1]])

## End(Not run)
```

parseMetaData	<i>Parses and downloads the meta data for a dataset</i>
---------------	---

Description

Parses and downloads the meta data for a dataset

Usage

```
parseMetaData(dataset_url)
```

Arguments

dataset_url	the url of the dataset
-------------	------------------------

resources	<i>Gets the resources from an object</i>
-----------	--

Description

Gets the resources from an object

Usage

```
resources(object, ...)
```

Arguments

object	an object with resources
...	optional additional arguments to methods

resources.berlin_data_dataset	<i>Gets the resources from a dataset</i>
-------------------------------	--

Description

Gets the resources from a dataset

Usage

```
## S3 method for class 'berlin_data_dataset'
resources(object, ...)
```


Arguments

object a dataset with a list of resources
 ... optional additional arguments

searchBerlinDatasets *Queries daten.berlin.de*

Description

Queries daten.berlin.de

Usage

```
searchBerlinDatasets(query, ...)
```

Arguments

query a query string to search daten.berlin.de
 ... optional additional arguments

Examples

```
## Not run:
result <- searchBerlinDatasets(query = "stolpersteine")
summary(result)
dataset <- getDatasetMetaData(result[[2]])
summary(dataset)
resource_list <- resources(dataset)
summary(resource_list)
data <- download(resource_list[[1]], sep=',')

## End(Not run)
```

updateCachedFeed *Updates cached_datasets_feed*

Description

Updates cached_datasets_feed

Usage

```
updateCachedFeed(xml.url = "http://daten.berlin.de/datensaetze/rss.xml")
```

Arguments

xml.url the url to the rss feed

Index

*Topic **datasets**

- cached_datasets_feed, [2](#)
- datasets_feed_tree, [2](#)

cached_datasets_feed, [2](#)

datasets_feed_tree, [2](#)

download, [3](#)

download.berlin_data_resource, [4](#)

download.CSV, [4](#)

download.default, [5](#)

download.JSON, [5](#)

download.TXT, [6](#)

download.XML, [6](#)

getDatasetMetaData, [7](#)

parseMetaData, [8](#)

resources, [8](#)

resources.berlin_data_dataset, [8](#)

searchBerlinDatasets, [9](#)

updateCachedFeed, [9](#)