

Package ‘rdatamarket’

August 29, 2016

Imports RCurl, RJSONIO

Maintainer Gunnlaugur Thor Briem <gthb@datamarket.com>

License MIT + file LICENSE

Title Data access API for DataMarket.com

Author Gunnlaugur Thor Briem <gthb@datamarket.com>

Description Fetches data from DataMarket.com, either as timeseries in zoo form (dmseries) or as long-form data frames (dmlist). Metadata including dimension structure is fetched with dminfo, or just the dimensions with dmdims.

Version 0.6.5

URL <http://datamarket.com/api/lib/rdatamarket>

Depends R (>= 2.10.0), zoo

NeedsCompilation no

Repository CRAN

Date/Publication 2014-11-24 18:40:24

R topics documented:

rdatamarket-package	2
dmCurlOptions	3
dmdims	3
dminfo	4
dminit	5
dmlist	6
dmseries	7
Index	8

rdatamarket-package *Data access API for DataMarket.com*

Description

Interacts with DataMarket.com, fetching data in timeseries (zoo) and long form (suitable for reshape), and metadata on datasets.

Quick start

Just find the data you want on <http://datamarket.com/> and then paste the URL from your browser into `dmlist` or `dmseries`:

```
plot(dmseries("http://datamarket.com/data/set/zy0/#ds=zy0!8o7=2"))
```

```
plot(dmseries("http://data.is/nyFeP9"))
```

```
l <- dmlist("http://data.is/nyFeP9")
```

or into `dminfo` to get the dataset metadata:

```
ds <- dminfo("http://data.is/nyFeP9")
```

DS parameters

In functions that take a parameter named `ds`, that parameter can be:

- a dataset ID ("17tm")
- a DS string ("17tm!kqc=a") consisting of a dataset ID along with some dimension filtering specifications; for more about DS strings, see Datamarket API documentation at <http://datamarket.com/api/v1/#about-ds>
- a query string ("foo=bar&ds=17tm!kqc=a&baz=xyzy") containing such a string in a `ds` parameter
- a URL ("http://datamarket.com/data/set/17tm/#ds=17tm!kqc=a") containing such a query string after `?` or `#`
- such a URL represented by a redirect ("http://data.is/r6JEsC") in a short-URL service (data.is, bit.ly, is.gd, t.co, url.is)
- a dataset object, obtained by calling `dminfo`.

Thus when viewing any dataset on datamarket.com, you can copy the URL from your browser bar and paste into a call to `dmseries` or `dmlist` to load the same data into R, or a call to `dminfo` to get a dataset object with which to make data queries.

Dimension filtering

The functions `dmseries` and `dmlist` can accept parameters named for the dimensions of the dataset, in order to filter the data: `dmlist("17tm", Country="Algeria")`

Examples

```
oil <- dminfo("17tm")
print(oil)
series <- dmseries(oil, Country="Yemen")
lis <- dmlist(oil, Country=c("Algeria", "Angola"))
```

dmCurlOptions *Set extra RCurl options for full HTTP control.*

Description

Use this to control the options with which RCurl connections are created, see [RCurl::curlOptions](#). A common use is setting a proxy to work with your company's firewall setup, see example below.

Usage

```
dmCurlOptions(..., .opts = list())
```

Arguments

...	name-value pairs specifying curl options. See full list of options with <code>'names(getCurlOptionsConstants())'</code> , their types with <code>'getCurlOptionTypes()'</code> , and full documentation in the curl man-page .
.opts	a named list of options, or a previously created <code>'CURLOptions'</code> object. These are merged with the options specified in <code>'...'</code> .

Examples

```
## Not run: dmCurlOptions(proxy='http://outproxy.mycompany.com')
```

dmdims *Fetch dimensions of a DataMarket dataset.*

Description

Get a list of dataset dimension objects for the given dataset.

Usage

```
dmdims(ds, .params = list())
```

Arguments

<code>ds</code>	a dataset ID, DS string, URL query-string, or whole URL. The DS string to send is extracted from the URL as needed, and short URLs at data.is, bit.ly, is.gd, t.co and url.is are expanded. If the DS string contains dimension filter specifications (the stuff after the ! character, so it's not just a dataset ID), these are preserved in the request to the API, but for normal DataMarket datasets they do not affect the response.
<code>.params</code>	extra GET parameters to pass along in the API request.

Value

named list of dataset dimension information. Each name is a dataset ID and each element is a named list of dimensions of that dataset. Each dimension is named for its dimension ID in that list, and is itself a named list of the four properties `id`, `title`, `type`, and `values`. The first three of these properties are character strings, while `values` is a named list of dimension values. Each of these is a list of two properties `id`, `title`, and the `id` is also the name of the dimension value

Examples

```
dmdims("17tm")
dmdims("17tm!kqc=a")
dmdims("ds=17tm")
dmdims("ds=17tm!kqc=a")
dmdims("foo=bar&ds=17tm&baz=xyzy")
dmdims("http://datamarket.com/api/v1/series.json?foo=bar&ds=17tm&baz=xyzy")
dmdims("http://datamarket.com/data/set/17tm/#ds=17tm")
```

dminfo

Fetch information about a DataMarket dataset.

Description

This function makes an `info` API call to fetch metadata (including dimension structure) of a DataMarket dataset.

Usage

```
dminfo(ds, .params = list())
```

Arguments

<code>ds</code>	a dataset ID, DS string, URL query-string, or whole URL. The DS string to send is extracted from the URL as needed, and short URLs at data.is, bit.ly, is.gd, t.co and url.is are expanded.
<code>.params</code>	extra GET parameters to pass along in the API request.

Value

a structure of named lists representing the dataset metadata. If the resolved DS string is a multiple-dataset DS string (that is, contains / forward-slash characters), then the returned value is a named list of such structures, with 'strsplit(ds, '/')' as names.

Examples

```
dminfo("17tm")
dminfo("17tm!kqc=a")
dminfo("ds=17tm")
dminfo("ds=17tm!kqc=a")
dminfo("foo=bar&ds=17tm&baz=xyzyzy")
dminfo("http://datamarket.com/api/v1/series.json?foo=bar&ds=17tm&baz=xyzyzy")
dminfo("http://datamarket.com/data/set/17tm/#ds=17tm")
dminfo("http://datamarket.com/data/set/12r9/male-population-thousands")
```

dminit

Initialize DataMarket client with an API key

Description

An API key is required to access locked and non-public data. Access to public datasets does not require an API key, for now at least.

Usage

```
dminit(api.key)
```

Arguments

api.key a DataMarket API key, found in **your account profile**, or NULL to forget the previously entered API key.

Examples

```
dminit("1234567890abcdef1234567890abcdef")
dminit(NULL)
```

dmlist

Fetch data in long form as a data.frame from a DataMarket dataset.

Description

This performs a list API request at DataMarket.com, fetching the requested data and wrapping it in a `data.frame` object.

Usage

```
dmlist(ds, .params = list(), ...)
```

Arguments

<code>ds</code>	a dataset ID, DS string, URL query-string, or whole URL. The DS string to send is extracted from the URL as needed, and short URLs at <code>data.is</code> , <code>bit.ly</code> , <code>is.gd</code> , <code>t.co</code> and <code>url.is</code> are expanded.
<code>.params</code>	extra GET parameters to pass along in the API request.
<code>...</code>	named parameters whose names are dimension titles or IDs, and whose values are titles or IDs of values of those dimensions. E.g. if dataset <code>17tm</code> has a dimension named <code>Country</code> , then <code>dmlist("17tm", Country="Algeria")</code> filters on that dimension. If the dimension name includes spaces, it needs to be quoted: <code>dmlist("12rb", "Country or Area"="Afghanistan")</code>

Value

a data frame representing the fetched data. The data frame has one column for each of the dataset's dimensions, containing a factor whose levels are values of that dimension, and a 'Value' column holding a numerical value for each combination of dimension values.

Examples

```
dmlist("17tm")
dmlist("17tm!kqc=a")
dmlist("ds=17tm")
dmlist("ds=17tm!kqc=a")
dmlist("foo=bar&ds=17tm&baz=xyzzy")
dmlist("http://datamarket.com/api/v1/series.json?foo=bar&ds=17tm&baz=xyzzy")
dmlist("http://datamarket.com/data/set/17tm/#ds=17tm")
dmlist("17tm", Country="Algeria")
dmlist("17tm", Country=c("Algeria", "Angola"))
dmlist("12rb", "Country or Area"="Afghanistan")
```

`dmseries`*Fetch timeseries from a DataMarket dataset.*

Description

This performs a series API request at DataMarket.com, fetching the requested data and wrapping it in a zoo object.

Usage

```
dmseries(ds, .params = list(), ...)
```

Arguments

<code>ds</code>	a dataset ID, DS string, URL query-string, or whole URL. The DS string to send is extracted from the URL as needed, and short URLs at data.is, bit.ly, is.gd, t.co and url.is are expanded.
<code>.params</code>	extra GET parameters to pass along in the API request.
<code>...</code>	named parameters whose names are dimension titles or IDs, and whose values are titles or IDs of values of those dimensions. E.g. if dataset 17tm has a dimension named Country, then <code>dmseries("17tm", Country="Algeria")</code> filters on that dimension. If the dimension name includes spaces, it needs to be quoted: <code>dmseries("12rb", "Country or Area"="Afghanistan")</code>

Value

a zoo object representing the fetched timeseries.

Examples

```
dmseries("17tm")
dmseries("17tm!kqc=a")
dmseries("ds=17tm")
dmseries("ds=17tm!kqc=a")
dmseries("foo=bar&ds=17tm&baz=xyzy")
dmseries("http://datamarket.com/api/v1/series.json?foo=bar&ds=17tm&baz=xyzy")
dmseries("http://datamarket.com/data/set/17tm/#ds=17tm")
dmseries("17tm", Country="Algeria")
dmseries("17tm", Country=c("Algeria", "Angola"))
```

Index

`dmCurlOptions`, [3](#)

`dmdims`, [3](#)

`dminfo`, [2, 4](#)

`dminit`, [5](#)

`dmList`, [2, 6](#)

`dmseries`, [2, 7](#)

`rdatamarket-package`, [2](#)