Package 'RblDataLicense'

March 15, 2019

Type Package

Title Connect R to 'Bloomberg Data License'

Version 0.1.2

Date 2019-03-01

Author Emanuele Guidotti

Maintainer Emanuele Guidotti <emanuele.guidotti@hotmail.it>

Description Download prices, market data, and master data with
the 'Bloomberg Data License' service from
<https://www.bloomberg.com/professional/product/data-license/>.
As a prerequisite, you need a valid license from 'Bloomberg'.
This software and its author are in no way affiliated,
endorsed, or approved by 'Bloomberg' or any of its affiliates.
'Bloomberg' is a registered trademark.

License GPL-3

URL https://github.com/emanuele-guidotti/RblDataLicense

BugReports https://github.com/emanuele-guidotti/RblDataLicense/issues

Depends curl, RCurl, xts, R (>= 3.0)

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-03-15 16:53:48 UTC

R topics documented:

`RblConnect` *Create an SFTP connection to Bloomberg Datalicense*

Description

An sftp connection to Bloomberg Datalicense is established. On some Linux systems, this may not work out of the box, as libcurl does not natively support sftp. In that case, you need to compile curl with SFTP support. See here for details: http://askubuntu.com/questions/195545/how-to-enable-sftp-support-in-curl

Usage

```
RblConnect(user, pw, verbose = TRUE)
```

Arguments

`user`	The account number assigned by Bloomberg
`pw`	The password assigned by Bloomberg
`verbose`	logical. Should R report extra information on progress?

Value

logical. Is the connection succesful?

Examples

```
## Not run:
# These are dummy credentials. Replace with the credentials received from Bloomberg
RblConnect(user = 'dl000000', pw = '0000000000000000')

## End(Not run)
```

RblDownload *Download a file from Bloomberg*

Description

Download a generic file from Bloomberg. Use `RblFiles` to list the files available at Bloomberg. If the file is not available yet (i.e. a request file has just been uploaded), the function waits until the response file is there or the timeout is reached

Usage

```
RblDownload(file, pollFrequency = 60, timeout = 3600, verbose = TRUE)
```

Arguments

`file`	character string representing the file to download
`pollFrequency`	the polling frequency to check if file is available at Bloomberg
`timeout`	the timeout in seconds
`verbose`	logical. Should R report extra information on progress?

Value

character string. Path to the downloaded file. NULL on failure

Examples

```
## Not run:
# Run RblConnect first
RblRequest <- RblRequestBuilder(header = list(FIRMNAME = RblUser(), PROGRAMNAME = 'getdata'),
                                fields = c('PX_LAST'), identifiers = c('SXXE Index'))
req <- RblUpload(RblRequest)
out <- RblDownload(req$out)
out

## End(Not run)
```

RblFiles *List files available at Bloomberg*

Description

Retrieve files available at Bloomberg.

Usage

```
RblFiles()
```

Value

Vector of character strings representing the filenames available at Bloomberg

Examples

```
## Not run:
# Run RblConnect first
RblFiles()

## End(Not run)
```

`RblParse` *Parse Bloomberg response file and fetch data*

Description

Parse Bloomberg response file. The PROGRAMNAME in use is auto detected: 'getdata' and 'gethistory' are supported.

Usage

```
RblParse(file, auto.assign = FALSE, env = parent.frame(),
  verbose = TRUE)
```

Arguments

`file`	character string representing the local file to parse (see `RblDownload`)
`auto.assign`	logical. Should results be loaded to env when using PROGRAMNAME=gethistory?
`env`	where to create objects if auto.assign = TRUE
`verbose`	logical. Should R report extra information on progress?

Value

PROGRAMNAME=getdata data.frame containig identifiers (rows) and fields (columns). NULL on failure.

PROGRAMNAME=gethistory list of xts objects. If *auto.assign*=TRUE the xts objects are loaded in *env* and the object names are returned. NULL on failure.

Examples

```
## Not run:
# Run RblConnect first
RblRequest <- RblRequestBuilder(header = list(FIRMNAME = RblUser(), PROGRAMNAME = 'getdata'),
                                fields = c('PX_LAST'), identifiers = c('SXXE Index'))
req <- RblUpload(RblRequest)
out <- RblDownload(req$out)
data <- RblParse(d)
data

## End(Not run)
```

RblQuery *Query Bloomberg and Fetch Data*

Description

The function provides a high level interface to Bloomberg Datalicense 'getdata' and 'gethistory' programs.

Usage

```
RblQuery(identifiers, fields, from = NULL, to = Sys.Date(),
  overrides = NULL, auto.assign = FALSE, env = parent.frame(),
  split = 100, pollFrequency = 60, timeout = 3600, verbose = TRUE)
```

Arguments

identifiers	vector of Bloomberg identifiers. Ex c('SXXE Index', 'SX5E Index')
fields	vector of Bloomberg fields. Ex. c('PX_LAST', 'PX_CLOSE', 'PX_OPEN', 'PX_HIGH', 'PX_LOW')
from	date or string (format YYYY-MM-DD). The start time of the period of interest
to	date or string (format YYYY-MM-DD). The end time of the period of interest. Ignored if *from* is not provided
overrides	named list of Bloomberg overrides. Ex list('END_DT' = '20100101')
auto.assign	logical. Should results be loaded to env? Ignored if *from* is not provided
env	where to create objects if auto.assign = TRUE
split	maximum number of identifiers to process at once. Split requests to avoid memory leaks
pollFrequency	the polling frequency to check if the response file is available at Bloomberg
timeout	the timeout in seconds
verbose	logical. Should R report extra information on progress?

Details

The following routine to query Bloomberg is implemented:

Build the request file see RblRequestBuilder

Upload the request file see RblUpload

Download the response file see RblDownload

Parse the response file see RblParse

Value

A list with components

req List of characters representing each of the request files uploaded to Bloomberg

out List of characters representing each of the response file downloaded from Bloomberg

data Return of RblParse

Examples

```
## Not run:
# Run RblConnect first
x <- RblQuery(fields = c('PX_LAST', 'PX_OPEN', 'PX_HIGH', 'PX_LOW'),
              identifiers = c('SXXE Index', "SX5E Index"),
              from = '2005-01-01')
str(x)

## End(Not run)
```

RblRequestBuilder *Build a request file to query Bloomberg*

Description

The request file is generated according to Bloomberg Data License documentation

Usage

```
RblRequestBuilder(header, fields, identifiers, overrides = NULL)
```

Arguments

header	names list of headers. Ex. list(FIRMNAME = RblUser(), PROGRAMNAME = 'getdata')
fields	vector of Bloomberg fields. Ex. c('PX_LAST', 'PX_OPEN', 'PX_HIGH', 'PX_LOW')
identifiers	vector of Bloomberg identifiers. Ex c('SXXE Index', 'SX5E Index')
overrides	named list of Bloomberg overrides. Ex list('END_DT' = '20100101')

Value

character string representing the request file. Upload it to query Bloomberg (see RblUpload)

Examples

```
## Not run:
# Run RblConnect first
RblRequest <- RblRequestBuilder(header = list(FIRMNAME = RblUser(), PROGRAMNAME = 'getdata'),
                                fields = c('PX_LAST'), identifiers = c('SXXE Index'))
RblRequest

## End(Not run)
```

RblUpload *Upload a request file to Bloomberg*

Description

Upload a request file to query Bloomberg. A response file will be generated by Bloomberg. The request file can be user-defined following the Bloomberg Data License documentation or generated with the RblRequestBuilder function. The response file needs to be downloaded (see RblDownload) and parsed (see RblParse) to make data available in R.

Usage

```
RblUpload(RblRequest, verbose = TRUE)
```

Arguments

RblRequest	character string representing the request file according to Bloomberg Datalicense documentation. Can be generated with the RblRequestBuilder function
verbose	logical. Should R report extra information on progress?

Value

A list with components

req the request filename

out the response filename

Examples

```
## Not run:
# Run RblConnect first
RblRequest <- RblRequestBuilder(header = list(FIRMNAME = RblUser(), PROGRAMNAME = 'getdata'),
                                fields = c('PX_LAST'), identifiers = c('SXXE Index'))
req <- RblUpload(RblRequest)
req

## End(Not run)
```

`RblUrl` *Get Bloomberg connection Url*

Description

Retrieve the Bloomberg connection Url.

Usage

```
RblUrl()
```

Value

Url string

Examples

```
## Not run:
# Run RblConnect first
RblUrl()

## End(Not run)
```

`RblUser` *Get Bloomberg User*

Description

Retrieve the Bloomberg User.

Usage

```
RblUser()
```

Value

User string

Examples

```
## Not run:
# Run RblConnect first
RblUser()

## End(Not run)
```

Index