Package 'dataonderivatives'

February 10, 2018

Type Package

Title Easily Source Publicly Available Data on Derivatives

Version 0.3.1

Description Post Global Financial Crisis derivatives reforms have lifted the veil off over-the-counter (OTC) derivative markets. Swap Execution Facilities (SEFs) and Swap Data Repositories (SDRs) now publish data on swaps that are traded on or reported to those facilities (respectively). This package provides you the ability to get this data from supported sources.

License GPL-2

Depends R (>= 3.3.0)

Imports assertthat (>= 0.1), httr (>= 1.2.1), lubridate (>= 1.3.3), readr (>= 1.1.0), tibble (>= 1.3.0), utils (>= 3.3.0)

Suggests testthat (>= 1.0.0), covr

URL https://imanuelcostigan.github.io/dataonderivatives, https://github.com/imanuelcostigan/dataonderivatives

BugReports https://github.com/imanuelcostigan/dataonderivatives/issues

RoxygenNote 6.0.1

NeedsCompilation no

Author Imanuel Costigan [aut, cre]

Maintainer Imanuel Costigan <i.costigan@me.com>

Repository CRAN

Date/Publication 2018-02-10 07:45:42 UTC

R topics documented:

bsdr *Get Bloomberg SDR data*

Description

The Bloomberg Swap Data Repository (BSDR) is a registered U.S. swap data repository that allows market participants to fulfil their public disclosure obligations under U.S. legislation. BSDR is required to make publicly available price, trading volume and other trading data reported to its U.S. repository. It publishes this data on its website in real-time and also on a historical basis. I have reverse engineered the JavaScript libraries used by its website to call the Bloomberg Application Service using `POST` requests to a target URL.

Usage

```
bsdr(dates, asset_class, currency = NULL)
```

Arguments

`dates`	the dates for which data is required as Date or DateTime object. It will use all date-time elements including year, month, day, hour, minute, second (up to milliseconds) and time zone information to determine the set of trades to return. It will return the set of trades for the day starting on `dates` if `dates` is of length one or the set of trades between the first and second elements of `dates` if `dates` has a length greater than one.
`asset_class`	the asset class for which you would like to download trade data. Valid inputs are `"CR"` (credit), `"IR"` (rates), `"EQ"` (equities), `"FX"` (foreign exchange), `"CO"` (commodities).
`currency`	the currency for which you would like to get trades for. These should be the currency's ISO code

Value

a tibble containing the requested data, or an empty tibble if data is unavailable. Note that fields containing notional information are not necessarily numeric values are capped in public data to meet CFTC requirements.

References

BSDR search Bloomberg SDR API

Examples

```
## Not run:
library (lubridate)
# Interest rate trades for day starting 19 May 2017
bsdr(ymd(20170519), "IR")
# Interest rate trades for the period between 19 May 2017 and 23 May 2017
```

```
bsdr(ymd(20170519, 20170523), "IR")

## End(Not run)
```

`bsef` *Get Bloomberg SEF data*

Description

The Bloomberg Swap Execution Facility (SEF) offers customers the ability to execute derivative instruments across a number of different asset classes. It is required to make publicly available price, trading volume and other trading data. It publishes this data on its website. I have reverse engineered the JavaScript libraries used by its website to call the Bloomberg Application Service using `POST` requests to a target URL.

Usage

```
bsef(date, asset_class)
```

Arguments

`date`	the date for which data is required as Date or DateTime object. Only the year, month and day elements of the object are used. Must be of length one.
`asset_class`	the asset class for which you would like to download trade data. Valid inputs are `"CR"` (credit), `"IR"` (rates), `"EQ"` (equities), `"FX"` (foreign exchange), `"CO"` (commodities) and must be a string.

Value

a tibble containing the requested data, or an empty tibble if data is unavailable

References

Bloomberg SEF data

Examples

```
## Not run:
library (lubridate)
# All asset classes
bsef(ymd(20140528), "IR")

## End(Not run)
```

cme *Get CME SDR data*

Description

The CME Swap Data Repository (SDR) is a registered U.S. swap data repository that allows market participants to fulfil their public disclosure obligations under U.S. legislation. CME is required to make publicly available price, trading volume and other trading data. It publishes this data on an FTP site.

Usage

```
cme(date, asset_class, field_specs = NULL)

cme_field_specs(asset_class)
```

Arguments

`date`	the date for which data is required as Date or DateTime object. It will only use the year, month and day elements to determine the set of trades to return. It will return the set of trades for the day starting on `date`.
`asset_class`	the asset class for which you would like to download trade data. Valid inputs are `"IR"` (rates), `"FX"` (foreign exchange), `"CO"` (commodities). This must be a string.
`field_specs`	a valid column specification that is passed to `readr::read_csv()` with a default value provided by `cme_field_specs()`. Note that you will likely need to set your own spec as the CME file formats have changed over time.

Value

a tibble containing the requested data, or an empty tibble if data is unavailable

References

CME SDR

Examples

```
## Not run:
library(lubridate)
cme(ymd(20150506), "CO")

## End(Not run)
```

ddr *Get DDR data*

Description

The DTCC Data Repository is a registered U.S. swap data repository that allows market participants to fulfil their public disclosure obligations under U.S. legislation. This function will give you the ability to download trade-level data that is reported by market participants. The field names are (and is assumed to be) the same for each asset class.

Usage

```
ddr(date, asset_class, field_specs = ddr_field_specs())

ddr_field_specs()
```

Arguments

`date`	the date for which data is required as Date or DateTime object. Only the year, month and day elements of the object are used and it must of be length one.
`asset_class`	the asset class for which you would like to download trade data. Valid inputs are `"CR"` (credit), `"IR"` (rates), `"EQ"` (equities), `"FX"` (foreign exchange), `"CO"` (commodities). This must be a string.
`field_specs`	a valid column specification that is passed to `readr::read_csv()` with a default value provided by `ddr_field_specs()`

Value

a tibble that contains the requested data. If no data exists on that date, an empty tibble is returned.

References

DDR Real Time Dissemination Platform

Examples

```
## Not run:
library("lubridate")
ddr(ymd(20170525), "IR") # Not empty

## End(Not run)
```

Index