Package 'IMFData'

October 29, 2016

Type Package

Title R Interface for International Monetary Fund(IMF) Data API

Version 0.2.0

Date 2016-10-16

Depends R (>= 3.1.0)

Imports httr, jsonlite, plyr

Description Search, extract and formulate IMF's datasets.

License MIT + file LICENSE

LazyData TRUE

KeepSource TRUE

URL https://github.com/mingjerli/IMFData

RoxygenNote 5.0.1

NeedsCompilation no

Author Ming-Jer Lee [aut, cre]

Maintainer Ming-Jer Lee <mingjerli@gmail.com>

Repository CRAN

Date/Publication 2016-10-29 00:27:17

R topics documented:

CodeSearch *Search Available Code in a Dimension of a Given Dataset*

Description

CodeSearch search matching codes in a given dimension of a dataset

Usage

```
CodeSearch(available.codes, code, searchtext, search.value = TRUE,
  search.text = TRUE)
```

Arguments

available.codes	string. Database ID of the dataset.
code	string. Dimension code get from DataStructureMethod.
searchtext	string. String to search in the dimension code.
search.value	logical. If true, it will search searchtext in CodeValue.
search.text	logical. If true, it will search searchtext in CodeText.

Value

A list for each dimension. The name of the list is the dimension name. Each element of the list is a data frame with two columns; dimension code and dimension text(description).

Examples

```
IFS.available.codes <- DataStructureMethod('IFS') # Get dimension code of IFS dataset
names(IFS.available.codes) # Available dimension code
IFS.available.codes[[1]] # Possible code in the first dimension
CodeSearch(IFS.available.codes, 'CLL', 'GDP') # Error (CLL is not a dimension code of IFS dataset)
CodeSearch(IFS.available.codes, 'CL_INDICATOR_IFS', 'GDP') # Search code contains GDP
CodeSearch(IFS.available.codes, 'CL_INDICATOR_IFS', 'GDPABCDE') # NULL
```

CompactDataMethod *Make Query to Get Data from the API*

Description

CompactDataMethod will make an API call with filter (if not NULL) to get data of each available code in the data set. It might only return incomplete data because of API limit. In this case, try to add more constraints in the filter.

Usage

```
CompactDataMethod(databaseID, queryfilter = NULL, startdate = "2001-01-01",
  enddate = "2001-12-31", checkquery = FALSE, verbose = FALSE,
  tidy = FALSE)
```

Arguments

databaseID	A character string. Database ID for the dataset. Can be obtained from DataflowMethod
queryfilter	list. A list that contains filter to use in the API call. If not NULL, it must be a list with the lengtho of dimension of dataset. If NULL, no filter has been set.
startdate	string. Start date in format of "YYYY-mm-dd".
enddate	string. End date in format of "YYYY-mm-dd".
checkquery	logical. If true, it will check the database ID is available or not.
verbose	logical. If true, it will print the exact API call.
tidy	logical. If true, it will return a simple data fram.

Value

A data frame. The last column, Obs, is a time series data described by other columns.

Examples

```
databaseID <- 'IFS'
startdate='2001-01-01'
enddate='2016-12-31'
checkquery = FALSE

IFS.available.codes <- DataStructureMethod('IFS')

## Germany, Norminal GDP in Euros, Norminal GDP in National Currency
queryfilter <- list(CL_FREA="", CL_AREA_IFS="GR", CL_INDICATOR_IFS =c("NGDP_EUR","NGDP_XDC"))
GR.NGDP.query <- CompactDataMethod(databaseID, queryfilter, startdate, enddate, checkquery)
GR.NGDP.query[,1:5]
GR.NGDP.query$Obs[[1]]
GR.NGDP.query$Obs[[2]]
```

```
## Example for verbose
GR.NGDP.query <- CompactDataMethod(databaseID, queryfilter, startdate, enddate, verbose=TRUE)

## Example for tidy
GR.NGDP.query <- CompactDataMethod(databaseID, queryfilter, startdate, enddate, tidy=TRUE)
head(GR.NGDP.query)

## Quarterly, Germany, Norminal GDP in Euros, Norminal GDP in National Currency
queryfilter <- list(CL_FREA="Q", CL_AREA_IFS="GR", CL_INDICATOR_IFS =c("NGDP_EUR","NGDP_XDC"))
Q.GR.NGDP.query <- CompactDataMethod(databaseID, queryfilter, startdate, enddate, checkquery)
Q.GR.NGDP.query[,1:5]
Q.GR.NGDP.query$Obs[[1]]

## Quarterly, USA
queryfilter <- list(CL_FREA="Q", CL_AREA_IFS="US", CL_INDICATOR_IFS = "")
Q.US.query <- CompactDataMethod(databaseID, queryfilter, startdate, enddate, checkquery)
Q.US.query[,1:5]
CodeSearch(IFS.available.codes, "CL_INDICATOR_IFS", "FITB_3M_PA") # Reverse look up meaning of code

## Quarterly, USA, GDP related
IFS.available.codes <- DataStructureMethod('IFS')
ALLGDPCodeValue <- CodeSearch(IFS.available.codes, "CL_INDICATOR_IFS","GDP")$CodeValue
queryfilter <- list(CL_FREA="Q", CL_AREA_IFS="US", CL_INDICATOR_IFS =ALLGDPCodeValue[1:10])
Q.US.GDP.query <- CompactDataMethod(databaseID, queryfilter, startdate, enddate, checkquery)
Q.US.GDP.query[,1:5]
Q.US.GDP.query$Obs[[1]]
Q.US.GDP.query$Obs[[2]]

## Quarterly, US, NGDP_SA_AR_XDC
queryfilter <- list(CL_FREA="Q", CL_AREA_IFS="US", CL_INDICATOR_IFS ="NGDP_SA_AR_XDC")
Q.US.NGDP.query <- CompactDataMethod(databaseID, queryfilter, startdate, enddate, checkquery)
Q.US.NGDP.query[,1:5]
Q.US.NGDP.query$Obs[[1]]

## Monthly, US, NGDP_SA_AR_XDC
queryfilter <- list(CL_FREA="M", CL_AREA_IFS="", CL_INDICATOR_IFS ="NGDP_SA_AR_XDC")
M.NGDP.query <- CompactDataMethod(databaseID, queryfilter, startdate, enddate, checkquery)
M.NGDP.query$Obs # NULL

## Example for DOT dataset
DOT.available.codes <- DataStructureMethod('DOT')
names(DOT.available.codes)
queryfilter <- list(CL_FREQ = "", CL_AREA_DOT="US",
                    CL_INDICATOR_DOT = "", CL_COUNTERPART_AREA_DOT="")
US.query <- CompactDataMethod('DOT', queryfilter, startdate, enddate, FALSE)
US.query[1:5,1:(length(US.query)-1)]
US.query$Obs[[1]] # Monthly. US. TMG_CIF_USD CH
```

DataflowMethod *Get List of Available Datasets from API*

Description

DataflowMethod returns a data frame with availble datasets

Usage

```
DataflowMethod()
```

Value

DataflowMethod returns a data frame object to describe available datasets from IMF data API. This data frame includes the following two columns:

DatabaseID - Database ID uses for making API call
DatabaseText - Database description

Examples

```
availableDB <- DataflowMethod()
availableDB
availableDB$DatabaseID[1]
```

DataStructureMethod *Get List of Dimension of a Given Dataset*

Description

DataStructureMethod get the data structure of dataset with available code and description for each dimension.

Usage

```
DataStructureMethod(databaseID, checkquery = FALSE)
```

Arguments

databaseID	string. Database ID of the dataset. Obtained from DataflowMethod.
checkquery	logical. If true, it will check the database ID is available or not.

Value

DataStructureMethod returns a list of data frame to describe available dimensions in the dataset. The name of the list is the dimension name. Each element of the list is a data frame with two columns;

CodeValue - dimension code used for CompactDataMethod.
CodeText - dimension description.

Examples

```
available.codes <- DataStructureMethod('IFS')
names(available.codes)
available.codes[[1]]#'
```

Index