Package ‘wquantR’

July 6, 2017

Title Quant Library to Access Timeseries Data

Version 0.0.1

Description Package provides various data access methods for quantitative timeseries data. It utilizes java language to structure objects in point in the timeseries format. The packages should be used by data analysts and scientist to fetch data from SQL/NoSQL databases.

Imports rJava

BugReports `https://wolferesearch.atlassian.net/projects/WQUAN/issues`

Depends R (>= 3.3.0),
rJava(>= 0.9-8)

License Proprietary
Do not distribute outside of Wolferesearch.

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

R topics documented:

`wq.array2Matrix` *2-d Arry to matrix function*

Description

Converts a java Object[][] array containing double values to a simple matrix

Usage

```
wq.array2Matrix(arr)
```

`wq.array2namedMatrix` *Wquant matrix conversion function Converts matrix returned by wquant to*

Description

Wquant matrix conversion function Converts matrix returned by wquant to

Usage

```
wq.array2namedMatrix(res, arr)
```

`wq.countries` *Countries Returns all Contries with its meta data*

Description

Countries Returns all Contries with its meta data

Usage

```
wq.countries()
```

Value

Data Frame containing all countries

`wq.countryMeta` *Returns Country Meta Data*

Description

Returns Country Meta Data

Usage

```
wq.countryMeta(country)
```

Arguments

`country`

`wq.define` *Defines a new expression*

Description

Defines a new expression

Usage

```
wq.define(expr)
```

Arguments

`expr`	expression to be defined

Examples

```
wq.define('VOLPRC=CSHTRD/PRCCD')
wq.define('PRCCHG=PRCCD/PRCCD_L1D')
```

`wq.downloadBasicFactors` *Downloads basic factors to file*

Description

Downloads basic factors to file

Usage

```
wq.downloadBasicFactors(f_download, country, directory, endDate)
```

Arguments

`endDate`

wq.downloadCountryFactors *Download Country's factors*

Description

Download Country's factors

Usage

```
wq.downloadCountryFactors(f_download, country, directory, endDate)
```

Arguments

`f_download` Download function that takes univ and factor name

`country` 2 Digit ISO Country Code

`filename` output file name

wq.downloadFactors *Download Factors*

Description

Download Factors

Usage

```
wq.downloadFactors(f_download, univ, factorNamesAll, country, filename)
```

Arguments

`f_download` Download function to use, basically a wrapper on wq.getdata. The function should take 2 parameters univ,factor list

`univ` Universe Id for which to download data

`factorNamesAll` Mnemonic of all factors to be downloaded

`country` Name of the country, only used for printing error

`filename` Full path of the file where to store the factors

Examples

```
startDate<-'2016-11-30'
endDate<-'2017-02-22'
f_download<-function(univ,factorname){
  wq.getdata(wq.newRequest()$testMode()$runFor(univ)$from(startDate)$to(endDate)$at('d')$a(factorname))
}
wq.downloadFactors(f_download,'TQA_SPCBMICJPUSD',c('RTN21D','RTN1D'),'/mnt/ebs1/data/factor_data/dailyJapa
```

`wq.downloadPriceFactors` *Download Price Factors*

Description

Download Price Factors

Usage

```
wq.downloadPriceFactors(f_download, country, directory, endDate)
```

Arguments

`endDate`

`wq.factor.detail` *Returns details of an attribute as data frame*

Description

Returns details of an attribute as data frame

Usage

```
wq.factor.detail(attr, source)
```

Arguments

`attr`	Attribute for which plan to be printed
`source`	Source of the attribute

Examples

```
wq.factor.detail('PRCCD','COMPUSTAT')
```

`wq.filename` *Returns standardized filename*

Description

Returns standardized filename

Usage

```
wq.filename(directory, prefix, cm, endDate)
```

Arguments

`endDate`

wq.getdata *Get Data*

Description

The main interface to get data from WQuant Library

Usage

```
wq.getdata(req, varname = NA)
```

Arguments

`req` Java Request Object, see example on how to build this

Examples

```
1. Returns output matrix with Compustat Pricing Element PRCCD for IBM
wq.getdata(wq.newRequest()$runFor('i:006066.01')$from('2014-01-21')$to('2015-08-21')$at('1m')$a('CS_PRCCD')

2. Returns output matrix with Compustat Pricing Element PRCCD for a test universe
wq.getdata(wq.newRequest()$runFor('i:006066.01')$from('2014-01-21')$to('2015-08-21')$at('1m')$a('CS_PRCCD')

3. Returns 2 output matrices for Compustat Elements PRCCD and CSHO for a test universe
 out<-wq.getdata(wq.newRequest()$runFor('TEST_UNIV_1')$from('2014-01-21')$to('2015-08-21')$at('1m')$a('CS_P
```

wq.indexMapping *Index Mapping Return primary index for BMI Index*

Description

Index Mapping Return primary index for BMI Index

Usage

```
wq.indexMapping()
```

Value

Return Universe Code

wq.init *WQuant Library Initialization*

Description

This function is called only once, once it is initialized a flag is set to ensure it is called again

Usage

```
wq.init()
```

wq.newRequest *New Request*

Description

Creates an empty data request

Usage

```
wq.newRequest()
```

wq.plan *Prints out the attribute plan*

Description

Prints out the attribute plan

Usage

```
wq.plan(attr)
```

Arguments

`attr`	Attribute for which plan to be printed

Examples

```
wq.plan('PRCCD')
```

wq.plan.json *Returns plan in json tree structure*

Description

Returns plan in json tree structure

Usage

```
wq.plan.json(attr, source)
```

Arguments

attr	Attribute/Factor Name
source	Source of the attribute

Examples

```
wq.plan.json('ACCRUALS','COMPUSTAT')
```

wq.plan.nodes *Provides data frame*

Description

Provides data frame

Usage

```
wq.plan.nodes(attr, source)
```

Arguments

attr	Attribute/Factor Name
source	Source of the attribute

Examples

```
wq.plan.nodes('ACCRUALS','COMPUSTAT')
```

`wq.plan.tree` *Returns attribute tree*

Description

Returns attribute tree

Usage

```
wq.plan.tree(attr, source)
```

Arguments

`attr`	Attribute/Factor Name
`source`	Source of the attribute

Examples

```
wq.plan.tree('ACCRUALS','COMPUSTAT')
```

`wq.port.delete` *Delete a portfolio (universe)*

Description

Delete a portfolio (universe)

Usage

```
wq.port.delete(id)
```

Arguments

`id`	(universe id)

`wq.port.get` *Gets the handle to an uploaded universe*

Description

Gets the handle to an uploaded universe

Usage

```
wq.port.get(id)
```

Arguments

`id`

Value

Portfolio Object (R6 Class)

`wq.port.list` *Lists down all custom portfolios*

Description

Lists down all custom portfolios

Usage

```
wq.port.list(currUserOnly = FALSE)
```

`wq.port.upload` *Uploads a portfolio from the file*

Description

Uploads a portfolio from the file

Uploads the portfolio, see

Usage

```
wq.port.upload(id, header, data, global = TRUE, pitId = FALSE,
  shortFormat = FALSE)

wq.port.upload(id, header, data, global = TRUE, pitId = FALSE,
  shortFormat = FALSE)
```

Arguments

`id` Unique id for the universe

`header` name of the columns in the data

`data` flattened data array as string

`global` Boolean flag indicating whether the portfolio should be mapped to global identifier. If False then it would be mapped to US/Canada identifier

`pitId` Boolean flag indicating whether the identifiers in the file are point in time

`shortFormat` Boolean flag indicating if the data is in short of long format see https://github.com/wolferesearch/docs/

`filen` Full path of the file containing the portfolio

`id` Unique id for the universe

`global` Boolean flag indicating whether the portfolio should be mapped to global identifier. If False then it would be mapped to US/Canada identifier

`pitId` Boolean flag indicating whether the identifiers in the file are point in time

`shortFormat` Boolean flag indicating if the data is in short of long format see https://github.com/wolferesearch/docs/

Value

Portfolio Object (R6 Class)

Portfolio Object (R6 Class)

Examples

```
port<-wq.port.upload('MyPortfolio1','/mnt/ebs1/data/demo/LongPort.csv',global=FALSE,pitId=FALSE,shortFormat

Prints out the summary
port$summary()

Prints out the available attributes
port$attributes()

Deletes the portfolio
port$delete()
data<-read.csv('/mnt/ebs1/data/demo/LongPort.csv')
header<-names(data)
port<-wq.port.upload('MyPort1',names(data),data,global=FALSE,pitId=FALSE,shortFormat=FALSE)

Prints out the summary
port$summary()

Prints out the available attributes
port$attributes()

Deletes the portfolio
port$delete()
```

wq.search *Searches particular entity type*

Description

Searches particular entity type

Usage

```
wq.search(type, query)
```

Arguments

type	Type of Entity (UNIVERSE/ATTRIBUTE/SECURITY)
query	Query Keywords

Examples

```
wq.search('UNIVERSE','TQA BMI United States')
wq.search('ATTRIBUTE','COMPUSTAT EPS')
wq.search('ATTRIBUTE','COMPUSTAT EPS')
```

wq.univflag *Univ Flag*

Description

Returns a TRUE/FALSE matrix. In Flag can be used to remove securities that are either a) Not in the universe on the date b) PRICE > 0 c) Market Cap > 0 d)

Usage

```
wq.univflag(req)
```

Arguments

req	Java Request Object, see example on how to build this

Examples

```
1. Returns output matrix with Compustat Pricing Element PRCCD for IBM
req<-wq.newRequest()$runFor('TSX')$from('2014-01-21')$to('2015-08-21')$at('1m')$a('PRCCD')
flag<-wq.univflag(req);
data<-wq.getdata(req);
```

Index