Package 'RJDProcessor'

October 3, 2024

Type Package

Title RJDProcessor

Version 0.1.4

Author Alessandro Piovani

Maintainer <alessandro.piovani@istat.it>

Description The rjdverse libraries are the officially recommended R software for seasonal adjustment in the European Central Bank and Statistical System.

The RJDProcessor library integrates the rjdverse packages into a fully R-based production pipeline, ready to be used and easily extendable by methodologists.

It offers the capability to manage the entire seasonal adjustment process: acquisition, processing, storage, automation, and not just seasonal adjustment of the data.

Processing of multiple time series is possible by storing their specifications in JSON files, and interoperability with other JDemetra+ software is guaranteed because RJDProcessor can read workspaces and is able to produce them as an output.

RJDProcessor also provides functions to manage workspaces, such as splitting a workspace containing multiple time series into individual single-series workspaces, which are suitable for storing in databases with single time series records. Functions to merge workspaces are also available.

License EUPL

Encoding UTF-8

LazyData true

Imports RJDemetra (>= 0.2.5),
rjson (>= 0.2.21)

Suggests rjd3providers (>= 3.2.3),
readxl (>= 1.4.3),
roxygen2 (>= 7.2.3)

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Collate import_and_interface_definition.R
RJDProcessor.R
Data_reader_csv.R
Data_reader_csv_istat_format.R
Data_reader_ext_reg_tsplus.R
Data_reader_ext_reg_xlsx.R
Data_reader_ext_reg_csv.R
Data_reader_xlsx.R
Data_reader_list.R

Data_reader_xml.R
Extended_tramoseats_spec.R
JD_JSON.R
JD_JSON_file_processor.R
basic_spec.R
utility_functions.R
workspaces_manager.R

R topics documented:

`Data_reader_csv` *Constructor (R-like) of the Data_reader object*

Description

This function creates a Data_reader object capable of reading data from CSV files and returning it using the `read_data()` function.

Usage

```
Data_reader_csv(input_source = NA, ...)
```

Arguments

`input_source`	A string with file name (also with path).

Value

The Data_reader_csv object

Examples

```
input_data_file_name <- "CSV-FAS/grezzi_trim_FAS.csv"
input_data_reader <- Data_reader_csv(input_source = input_data_file_name)
#input_data_readerATread_data() # uncomment and replace AT with its symbol
```

Data_reader_list *Constructor (R-like) of the Data_reader object*

Description

This function creates a Data_reader object capable of reading data from a list and returning it using the `read_data()` function.

Usage

```
Data_reader_list(input_source = NA, ...)
```

Arguments

`input_source`	A string with file name (also with path).

Value

The Data_reader_csv object

Examples

```
FATEXP_10_list <- list("series_name"="FATEXP_10", "dates"=c("2005-01-01","2005-02-01","2005-03-01"), "values
C_DEFL_list   <- list("series_name"="C_DEFL",   "dates"=c("2001-01-01","2001-02-01","2001-03-01"), "values"=
# ...
input_data_list  <- list(FATEXP_10_list, C_DEFL_list)
input_data_reader <- Data_reader_list(input_source = input_data_list)
#input_data_readerATread_data() # uncomment and replace AT with its symbol
```

read_data,Data_reader_csv-method *Get the data from a Data_reader_csv*

Description

This function returns the data from the input_source of the object.

Usage

```
## S4 method for signature 'Data_reader_csv'
read_data(object, ...)
```

Value

data in form of numeric matrix, with rownames = dates (in string format, YYYY-MM-DD) end colnames = time series names (string)

Examples

```
input_data_file_name <- "CSV-FAS/grezzi_trim_FAS.csv"
input_data_reader <- Data_reader_csv(input_source = input_data_file_name)
#input_data_readerATread_data() # uncomment and replace AT with its symbol
```

read_data,Data_reader_list-method
Get the data from a Data_reader_list

Description

This function returns the data from the input_source of the object.

Usage

```
## S4 method for signature 'Data_reader_list'
read_data(object, ...)
```

Value

data in form of numeric matrix, with rownames = dates (in string format, YYYY-MM-DD) end colnames = time series names (string)

Examples

```
FATEXP_10_list <- list("series_name"="FATEXP_10", "dates"=c("2005-01-01","2005-02-01","2005-03-01"), "values
C_DEFL_list   <- list("series_name"="C_DEFL",   "dates"=c("2001-01-01","2001-02-01","2001-03-01"), "values"=
#...
input_data_list  <- list(FATEXP_10_list, C_DEFL_list)
input_data_reader <- Data_reader_list(input_source = input_data_list)
#input_data_readerATread_data() # uncomment and replace AT with its symbol
```

RJDProcessor *RJDProcessor*

Description

A fully RJDemetra-based production pipeline for official statistics.

Details

RJDProcessor

The RJDProcessor library integrates the rjdverse packages into a fully R-based production pipeline, ready to be used and easily extendable by methodologists. It offers the capability to manage the entire seasonal adjustment process: acquisition, processing, storage, automation, and not just seasonal adjustment of the data. Processing of multiple time series is possible by storing their specifications in JSON files, and interoperability with other JDemetra+ software is guaranteed because RJDProcessor can read workspaces and is able to produce them as an output. RJDProcessor also provides functions to manage workspaces, such as splitting a workspace containing multiple time series into individual single-series workspaces, which are suitable for storing in databases with single time series records. Functions to merge workspaces are also available.

Author(s)

Alessandro Piovani alessandro.piovani@istat.it, alessandro.piovani13@gmail.com;

Index