

# Package ‘RJS DMX’

September 23, 2014

**Version** 1.1

**Title** R interface to SDMX Web Services

**Author** Attilio Mattiocco, Diana Nicoletti, Gianpaolo Lopez, Banca d'Italia

**Maintainer** Attilio Mattiocco <attilio.mattiocco@bancaditalia.it>

**Description** This package provides functions that can be used to retrieve data and metadata from SDMX compliant data providers.

**Depends** R (>= 2.12.0), rJava (>= 0.8-8), zoo (>= 1.6-4)

**SystemRequirements** Java (>= 1.6)

**License** EUPL

**URL** <https://github.com/amattioc/SDMX/>

**Copyright** Banca d'Italia

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-09-23 15:15:52

## R topics documented:

RJS DMX-package . . . . .	2
addProvider . . . . .	2
getCodes . . . . .	3
getDimensions . . . . .	4
getDSDIdentifier . . . . .	4
getFlows . . . . .	5
getProviders . . . . .	5
getSDMX . . . . .	6
getTimeSeries . . . . .	7
sdmxHelp . . . . .	8
<b>Index</b>	<b>9</b>

---

RJSdmx-package	<i>Gets timeseries from SDMX data Provider</i>
----------------	--

---

### Description

This package provides functions to extract timeseries from an SDMX Provider (e.g. ECB, OECD, EUROSTAT) via SDMX Web Service

### Details

Package: RJSdmx  
Type: Package

Download Timeseries in local environment using Web Services published by SDMX data Providers.

### Author(s)

Attilio Mattiocco, Diana Nicoletti, Bank of Italy, IT Support for the Economic Research <attilio.mattiocco@bancaditalia.it>  
diana.nicoletti@bancaditalia.it

### References

<http://sdmx.org/> <http://bancaditalia.it> <http://sdw.ecb.europa.eu/> <http://stats.oecd.org/>

### See Also

**getProviders, getTimeSeries, sdmxHelp**

### Examples

```
# not run
# my_ts = getTimeSeries('ECB', 'EXR.M.USD.EUR.SP00.A')
```

---

addProvider	<i>Configure a new data provider (only SDMX 2.1 REST providers are supported)</i>
-------------	---

---

### Description

This function can be used to configure a new (SDMX 2.1 compliant, REST based) data provider.

### Usage

```
addProvider(name, agency, endpoint, needsCredentials)
```

**Arguments**

name	the name of the provider
agency	the agency identifier of the provider
endpoint	the URL where the provider resides
needsCredentials	set this to TRUE if the user needs to authenticate to query the provider

**Author(s)**

Attilio Mattiocco

**Examples**

```
#not run
#addProvider('pname', 'pagency', 'pendpoint', F)
```

---

getCodes	<i>Extract the codes of a dimension</i>
----------	---

---

**Description**

This function is used to retrieve the list of codes available for the input dimension and flow

**Usage**

```
getCodes(provider, flow, dimension)
```

**Arguments**

provider	the name of the provider
flow	the identifier of the dataflow
dimension	the identifier of the dimension

**Author(s)**

Attilio Mattiocco, Diana Nicoletti

**Examples**

```
# not run
# dims=getCodes('ECB', 'EXR', 'FREQ')
```

getDimensions *Extract the dimensions of a DataFlow*

---

**Description**

This function is used to retrieve the list of dimensions of the input dataflow

**Usage**

```
getDimensions(provider, dataflow)
```

**Arguments**

provider	the name of the provider
dataflow	the identifier of the dataflow

**Author(s)**

Attilio Mattiocco, Diana Nicoletti

**Examples**

```
# not run  
# dims = getDimensions('ECB', 'EXR')
```

---

getDSDIdentifier *Extract the KeyFamily identifier of a DataFlow*

---

**Description**

This function is used to retrieve the name of the keyfamily of the input dataflow

**Usage**

```
getDSDIdentifier(provider, dataflow)
```

**Arguments**

provider	the name of the data provider
dataflow	the identifier of the dataflow

**Author(s)**

Attilio Mattiocco, Diana Nicoletti

**Examples**

```
# not run
# id = getDSDIdentifier('ECB','EXR')
```

---

getFlows	<i>Extract the list of DataFlows of a provider</i>
----------	--

---

**Description**

This function is used to query the list of dataflows of the provider. A matching pattern can be provided, if needed.

**Usage**

```
getFlows(provider, pattern)
```

**Arguments**

provider	the name of the provider
pattern	the pattern to match against the dataflow id or description. If a pattern is not provided, all dataflows are returned.

**Author(s)**

Atilio Mattiocco

**Examples**

```
# not run
# get all flows from ECB
# flows = getFlows('ECB')
# get all flows that contain the 'EXR'
# flows = getFlows('ECB','*EXR*')
```

---

getProviders	<i>Extract the list of available Data Providers</i>
--------------	---

---

**Description**

This function is used to query the list of data providers

**Usage**

```
getProviders()
```

**Author(s)**

Attilio Mattiocco

**Examples**

```
# not run
# getProviders()
```

---

getSDMX	<i>Extract a list of time series</i>
---------	--------------------------------------

---

**Description**

This function is used to extract a list of time series identified by the parameters provided in input.

**Usage**

```
getSDMX(provider, id, start, end)
```

**Arguments**

provider	the name of the provider
id	the identifier of the time series
start	the start time - optional
end	the end time - optional

**Author(s)**

Attilio Mattiocco

**Examples**

```
# not run
# get single time series: EXR.A.USD.EUR.SP00.A (alternatively: EXR/A+M.USD.EUR.SP00.A)
# my_ts=getSDMX('ECB', 'EXR.A.USD.EUR.SP00.A')
# get monthly and annual frequency: 'EXR.A|M.USD.EUR.SP00.A' (alternatively: EXR/A+M.USD.EUR.SP00.A)
# my_ts=getSDMX('ECB', 'EXR.A|M.USD.EUR.SP00.A')
# get all available frequencies: 'EXR.*.USD.EUR.SP00.A' (alternatively: EXR/.USD.EUR.SP00.A)
# my_ts=getSDMX('ECB', 'EXR.*.USD.EUR.SP00.A')
```

---

getTimeSeries	<i>Extract a list of time series</i>
---------------	--------------------------------------

---

## Description

This function is exactly the same as **getSDMX**

## Usage

```
getTimeSeries(provider, id, start, end)
```

## Arguments

provider	the name of the provider
id	the identifier of the time series
start	the start time - optional
end	the end time - optional

## Author(s)

Attilio Mattiocco

## See Also

**getSDMX**

## Examples

```
# not run
# get single time series: EXR.A.USD.EUR.SP00.A (alternatively: EXR/A+M.USD.EUR.SP00.A)
# my_ts=getTimeSeries('ECB','EXR.A.USD.EUR.SP00.A')
# get monthly and annual frequency: 'EXR.A|M.USD.EUR.SP00.A' (alternatively: EXR/A+M.USD.EUR.SP00.A)
# my_ts=getTimeSeries('ECB','EXR.A|M.USD.EUR.SP00.A')
# get all available frequencies: 'EXR.*.USD.EUR.SP00.A' (alternatively: EXR/.USD.EUR.SP00.A)
# my_ts=getTimeSeries('ECB','EXR.*.USD.EUR.SP00.A')
```

---

`sdmxHelp`*Open a helper graphical application*

---

**Description**

This function opens a small sdmx metadata browser that can be helpful when building queries.

**Usage**

```
sdmxHelp()
```

**Author(s)**

Attilio Mattiocco

**Examples**

```
# not run  
# sdmxHelp()
```



# Index

## \*Topic **package**

RJSDMX-package, [2](#)

addProvider, [2](#)

getCodes, [3](#)

getDimensions, [4](#)

getDSDIdentifier, [4](#)

getFlows, [5](#)

getProviders, [5](#)

getSDMX, [6](#)

getTimeSeries, [7](#)

RJSDMX (RJSDMX-package), [2](#)

RJSDMX-package, [2](#)

sdmxHelp, [8](#)