SIE example with Counterfeit Banknotes and Coins series

Victor Cuspinera

22/10/2020

This is the example shared in the **Usage** section of the main introduction of this Repository.

Overview

To show the usage of the SIE API with `siebanxicor` R-package, we will run through an example using the time series of *Annual counterfeit domestic banknotes detected* (serie SM1255) and *Annual counterfeit domestic banknotes detected* (SM1266).

1. Load library

After the `siebanxicor` package is installed, load this library.

```
library("siebanxicor")
```

2. Use `setToken(token)`

Bring your token and open the SIE API channel with the `setToken` utility function.

```
# bring the token
token_file <- read.csv("../token/SIE_Token.csv", header=FALSE)

# set the token
setToken(token_file$V2)
```

Notes:
- If you don't have a token to use SIE API, ***click here*** *to access the official website and obtain one.*
- I add a csv file where users should paste and save their token to run this example.

3. Get data with `getSeriesData(series, startDate, endDate)`

Get the time series of interest, in this case the **SM1255** and **SM1266** series of annual counterfeit of mexican banknotes and coins, using the `getSeriesData` function.

```
# setting the variables
my_series <- c("SM1255", "SM1266")
my_start <- '2010-01-01'
my_end <- Sys.Date() #looks for today's date

# getting the series
series <- getSeriesData(my_series, my_start, my_end)
```

. . . this is the vector we get as result:

```
## $SM1255
## $SM1255$date
##  [1] "2010-01-01" "2011-01-01" "2012-01-01" "2013-01-01" "2014-01-01"
##  [6] "2015-01-01" "2016-01-01" "2017-01-01" "2018-01-01" "2019-01-01"
##
## $SM1255$value
##  [1] 260419 316565 352625 332946 230530 264372 269099 301075 339655 302930
##
##
## $SM1266
## $SM1266$date
##  [1] "2010-01-01" "2011-01-01" "2012-01-01" "2013-01-01" "2014-01-01"
##  [6] "2015-01-01" "2016-01-01" "2017-01-01" "2018-01-01" "2019-01-01"
##
## $SM1266$value
##  [1]  5065  3423  1532  2435  6352 12606  1308   976  1712  3009
```

Note: to use the `getSeriesData` *function, you should previously call* `setToken`*.*

4. Get the metadata with `getSeriesMetadata(series, locale)`

This function returns the general information of series. To select the language of the metadata, set the *locale* variable as "en" for English, and "es" for Spanish.

```
# getting the metadata
getSeriesMetadata(my_series, locale="en")
```

```
##   idSerie
## 1  SM1266
## 2  SM1255
##                                                                                  title
## 1     Annual counterfeit coins detected per denomination (domestic coins), All denominations
## 2 Annual counterfeit notes detected per denomination (domestic banknotes), All denominations
##    startDate    endDate frequency dataType   unit
## 1 2006-01-01 2019-01-01    Annual    Flows Pieces
## 2 2006-01-01 2019-01-01    Annual    Flows Pieces
```

Note: to use the `getSeriesMetadata` *function, you should previously call* `setToken`*.*

5. Get a data frame of one series using `getSerieDataFrame(series, idSerie)`

This function will be helpful to get a data frame for the annual counterfeit of mexican banknotes **(SM1255)** series, from the vector returned by the `getSerieDataFrame` in the previous point #3.

```
# getting the series
df_SM1255 <- getSerieDataFrame(series, "SM1255")
```

. . . this is the data frame that we get as result:

```
##          date  value
## 1  2010-01-01 260419
## 2  2011-01-01 316565
## 3  2012-01-01 352625
## 4  2013-01-01 332946
## 5  2014-01-01 230530
## 6  2015-01-01 264372
## 7  2016-01-01 269099
## 8  2017-01-01 301075
## 9  2018-01-01 339655
## 10 2019-01-01 302930
```

Note: to use the ***getSerieDataFrame*** *function, you should previously call* ***setToken*** *and* ***getSerieData****.*

6. Get the last value one or many series with `getSeriesCurrentValue(series)`

Get the last value of the series **SM1255** and **SM1266** by using the `getSeriesCurrentValue` function.

```
series_last <- getSeriesCurrentValue(my_series)
```

. . . this is the data frame returned after using `getSeriesData`:

```
series_last
```

```
##   idSerie       date  value
## 1  SM1255 2019-01-01 302930
## 2  SM1266 2019-01-01   3009
```

Note: to use the ***getSeriesCurrentValue*** *function, you should previously call* ***setToken****.*