Thomson Reuters Tick History API for R

Document Version: 1.0
Author: Zhe Zhao
Issue Date: 07/22/2013

Introduction and Installation

This document is a brief introduction about how to use the package of Thomson Reuters Tick History(TRTH) API in R. With this package, one can download multiple symbol data very fast, and don't have to run the browser and log in TRTH. I hope this can save your time.

Here are some notes for you before installing the package.
1) You need R 2.15.x for this package, the new version R 3.x.x will not work.
2) Three more packages are required: **XML**, **base64** and **RCurl**. They are all available on CRAN.
3) This package have been tested on Windows 8 64-bit. It should also work on Win7, but not for other operating system. If you use linux or Mac, please send email to SIRCA and request another package specific to your system.

The most frequently used commands are at the following two section: *createCredential* and *submitFTPRequest*. With the examples listed, you can easily send a series of request to TRTH.

To know more about the package, you can read an official intro from SIRCA, its available on the Lab wiki. Or you can type **??rdth** in R to get more information.

Now let's start.

Credential

The very first step of using this package is to create a variable representing your credential of TRTH.
It works like this:

```
rdth <- createCredential(user="username@company.com", password="password")
```

I have no idea why the guys in SIRCA keep using "rdth", it appears through out the package. Anyway this variable represents the credential of user.

Submit FTP Request

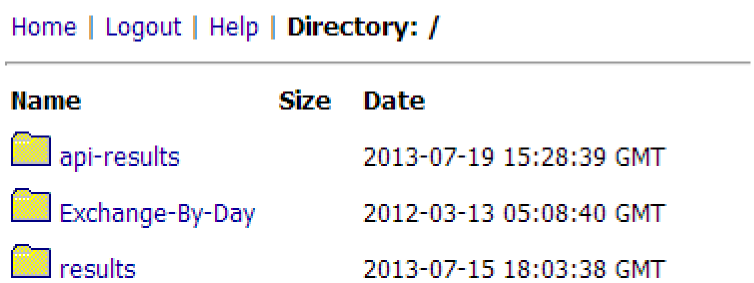
Now we are ready to submit requests. There are two kinds of requests in this package, a small one and a big one. The most frequently used operation is *submitFTPRequest*, it submits a large request to be delivered via FTP pull. FTP Pull stores the results on the TRTH FTPS server and the data can be retrieved either from the FTPS server or HTTP Pull server once GetRequestResult indicates the status is "Complete".
Let me use an example to explain:

```
> submitFTPRequest(rdth, friendlyname="test", instrumentList="GOOG.O; AAPL
   .O", startdate="2011-11-08", enddate="2011-11-12", starttime="00:00:00"
   , endtime="23:59:59.999", reqtype="TimeAndSales", mktdepth="0",
   messagetypelist="Trade:Price,Volume;Quote:Bid Price,Ask Price")
[1] "Your request id is: zzhao6@stevens.edu-test-N55222603"
> getRequestResult(rdth, "zzhao6@stevens.edu-test-N55222603")
[1] "Processing"
> getRequestResult(rdth, "zzhao6@stevens.edu-test-N55222603")
[1] "Complete"
```

In the preceding example, I was using submitFTPRequest() downloading 4 days tick data of AAPL and GOOG. Some arguments need to be explained:
i) *friendlyname* is a name defined by user, and it may be used to represent this request in other functions.
ii) *instrumentList* is a favorable argument that allows user use multiple symbols.
iii) *reqtype* represents the request data type, i.e "TimeAndSales", "MarketDepth", "NasdaqLevel2", "Intraday", "EndOfDay", "RTCE", "RTCEMarketDepth" and "Any".
iv) *messagetypelist*, this argument means what kind data will be downloaded. You need to be very careful about this one. The *messagetypelist* must be compatible with *reqtype*, otherwise

you will fail to run the statement.

After the function *getRequestResult* indicating the status is ”Complete”, you can retrieve your data from HTTP Pull of TRTH. To do that please logging in this website, with your own TRTH username and password:
`https://tickhistory.thomsonreuters.com/HttpPull/`.
Here is my interface when downloading, the result should be in folder *api-results*:



Here are more examples about other request types such as intraday data:

```
# To download "Nasdaqlevel2" data:
> submitFTPRequest(rdth, "test", instrumentList="MSFT.O", "2011-11-08", "
   2011-11-12", "00:00:00", "23:59:59.999", "NasdaqLevel2", mktdepth="0",
   messagetypelist="Market Maker:Bid Price,Total Bid Size,Number of Buyers
   ,Buyers list,Ask Price,Total Ask Size,Number of Sellers,Sellers list")

# To download minutely(intraday) data:
> submitFTPRequest(rdth, "test", instrumentList="GOOG.O", "2011-11-08", "
   2011-11-12", "00:00:00", "23:59:59.999", "Intraday", mktdepth="0",
   messagetypelist="Intraday 1Min:Open,High,Low,Last,Volume,Average
   Execution Price,VWAP,No. of Trades,Correction Qualifiers,Open Bid,High
   Bid,Low Bid,Close Bid,No. Of Bids,Open Ask,High Ask,Low Ask,Close Ask,
   No. Of Asks")

# To download 1 sec data:
 submitFTPRequest(rdth, "test", instrumentList="AAPL.O", "2011-11-08", "
   2011-11-12", "00:00:00", "23:59:59.999", "Intraday", mktdepth="0",
   messagetypelist="Intraday 1Sec:Open,High,Low,Last,Volume,Average
   Execution Price,VWAP,No. of Trades,Correction Qualifiers,Open Bid,High
   Bid,Low Bid,Close Bid,No. Of Bids,Open Ask,High Ask,Low Ask,Close Ask,
   No. Of Asks")
```

The preceding examples are quite long because I list many message types. To get more message types, please run **??getMessageTypes** in R or check the official documents from SIRCA.

Submit Request

You can use the function *submitRequest()* if you want to request data for only one day. The great advantage of this function is that the data will automatically downloaded in your R working directory, you don't have to open browser and log in.

```
> submitRequest(rdth, friendlyname="test", instrument="GOOG.O", date="
   2011-11-11", starttime="00:00:00", endtime="23:59:59.999", "
   TimeAndSales", mktdepth="0", messagetypelist="Trade:Price,Volume;Quote:
   Bid Price,Ask Price")
```

```
[1] "Submitted request: zzhao6@stevens.edu-test-N55206250"
[1] "Polling InFlightStatus to check if request is ready "
[1] "Polling InFlightStatus to check if request is ready "
[1] "Request is completed"
[1] "Getting request result"
```

Cancel and Clean Up Request

By using function *cancelRequest*, the results file will be removed, this can be either a small API request or an FTP request. Also the package has an operation called *cleanUp()*. The CleanUp call aborts all queued and running non-FTP requests, and in addition, removes complete non-FTP requests that have not been downloaded. The operation allows users to start the API with a clean state. This should be called at the start of your programs.

```
> cleanUp(rdth)
OK
 0
> cancelRequest(rdth, friendlyname="test")
OK
 0
```

"Get" Functions

The package also supports a serial of "get" functions, by which user can obtain informations from Thomson Reuters. For example, we can use *getAssetDomains(rdth)* to get a list of supported Asset Domains, Equities, Futures, Options or Indices. Or, as I mentioned before, *geRequestResult(rdth, friendlyname)* indicates whether a request is ready for user downloading.

```
> getAssetDomains(rdth)
   value                longName
1    BON            Fixed Income
2    COM Commodities and Energy
3    ECI     Economic Indicators
4    EQU                Equities
5    FOR        Foreign Exchange
6    FUN                   Funds
7    FUT                 Futures
8    IND                 Indices
9    MON            Money Market
10   OPT                 Options

> submitFTPRequest(rdth, "test", instrumentList="GOOG.O", "2011-11-08", "
   2011-11-12", "00:00:00", "23:59:59.999", "TimeAndSales", mktdepth="0",
   messagetypelist="Trade:Price,Volume;Quote:Bid Price,Ask Price")
[1] "Your request id is: zzhao6@stevens.edu-test-N55262549"
>
>  getRequestResult(rdth, "zzhao6@stevens.edu-test-N55262549")
[1] "Processing"
>  getRequestResult(rdth, "zzhao6@stevens.edu-test-N55262549")
[1] "Complete"
```

Here only two get functions are listed, for more information please type **??rdth** in R and read the document.

About RIC

RIC, an abbreviation for Reuters Instrument Code, is a ticker-like code used by Thomson Reuters to identify financial instruments and indices. The package has two functions about operations on RIC, one is search and the other is verify. With *verifyRICs()*, one can query the validity of an instrument and optionally return its reference data. If you use *searchRICs()*, you can search the reference data by date range and search criteria, it will return a list of instruments that match that criteria; multiple criteria may defined.

I prefer to skip the example in this section, because there exists a webpage from Thomson Reuters that implement almost the same thing but with shorter time and better performance. Some of the readers of this document may already know this webpage: `http://www.reuters.com/finance/stocks/lookup`. Just give it a shot.