Thomson Reuters Tick History API for R

Package Version: 0.9.5
Document Version: 1.2
Author: Zhe Zhao
Issue Date: 09/30/2014

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.

1. Please update to the latest version of R.
2. Three more packages are required: XML, base64 and RCurl. They are all available on CRAN.
3. I have tested this package on Windows 7, Windows 8 and Mac OS 10.9. It should also work for Windows 8.1 (please test). If you are using linux, please email me, I have a different package for linux.

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. Like we are opening TRTH webpage and login to the database:

```
library(THAPI)    # of course
rdth <- createCredential(user="username@company.com", password="password")
```

Submit FTP Request

Now we are ready to submit requests. There are two kinds of requests in this package, "submitRequest" and "submitFTPRequest". The latter one sends 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(r=rdth,
 friendlyname="test",
 instrumentList="GOOG.O",
 startdate="2011-11-08", enddate="2011-12-12",
 starttime="00:00:00", endtime="23:59:59.999",
 reqtype="TimeAndSales",
 mktdepth="0",
 messagetypelist="Trade:Price,Volume",
 reqInGMT="false", disInGMT="false")
[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:

1. *friendlyname* is a name defined by user, and it may be used to represent this request in other functions.

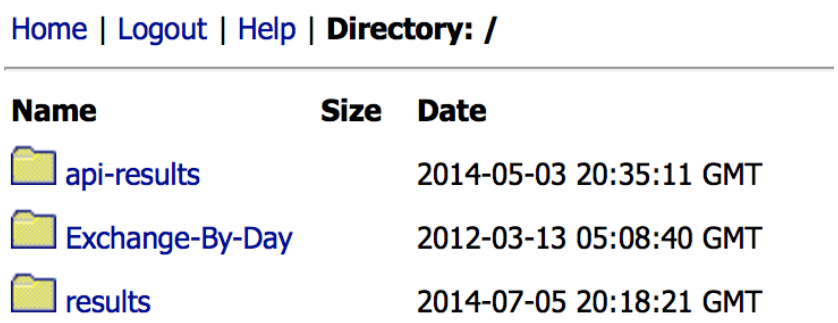
2. *instrumentList* is a favorable argument that allows user use multiple symbols.

3. *reqtype* represents the request data type, i.e "TimeAndSales", "MarketDepth", "NasdaqLevel2", "Intraday", "EndOfDay", "RTCE", "MarketDepth", and "Any".

4. *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.

Two "get" functions

To use the function submitFTPRequest() with high proficiency, you may need two more helper functions: *getRequestResult()* and *getMessageType()*.
With the first one, you will know what is the status of your previously submitted request. Your data is ready to download when the function indicates Complete, just as the preceding example shows. 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 screenshot when downloading, the result should be in folder "api-results":



The other function is *getMessageTypes*, which will return a list of supported message types for a given list of Asset Domains and request type. As the preceding notes refer, *messagetypelist* should exactly matches *reqtype*. And that is where you can use this function to obtain related informations.
Here are more examples about other request types such as intraday data, note that this time we are downloading different types:

```
# To download 1 sec(intraday) data:
> submitFTPRequest(rdth, "test", instrumentList="AAPL.O", "2011-11-08", " 2011-11-12", "00:00:00", "23:59:59.999", reqtype
      = "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")

# To download minute(intraday) data
> submitFTPRequest(rdth, "test", instrumentList="GOOG.O", "2011-11-08", " 2011-11-12", "00:00:00", "23:59:59.999", reqtype
      = "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")
```

The API package support more get functions, use ??rdth for more details.

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 dont 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
```

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 perfor- mance. Some of the readers of this document may already know this webpage:
http: //www.reuters.com/finance/stocks/lookup. Just give it a shot.