# RClient for the meXBT **DATA API**

## Introduction

R Code API for connecting to the **meXBT - The Mexican Exchange of Bitcoins**, *Public* API. This code is a series of functions for building *GET* *Request Queries* in order to pull information from the **meXBT** system, this can be read at their [HomePage](https://mexbt.com/en/api/http/)

* **License:** GNU General Public License
* **Location:** Mexico City

## R Packages/Libraries used

Some important functions are used to build this API Client/Wrapper. Most of them come from the following packages/libraries, which official documentation is also included in this repossitory:

* **base**: *Base Statistical and data functions in R.*
* **httr**: *Tools for Working with URLs and HTTP.*
* **jsonlite**: *A Robust, High Performance JSON Parser and Generator for R.*
* **lubridate**: *Make dealing with dates a little easier.*
* **plyr**: *Tools for Splitting, Applying and Combining Data.*
* **RCurl**: *General network (HTTP/FTP/...) client interface for R.*
* **xts**: *eXtensible Time Series.*
* **zoo**: *S3 Infrastructure for Regular and Irregular Time Series.*

You can check and download the official documentation for these packages from this repository [Here](https://github.com/FranciscoME/meXBTRClient/tree/master/LibrariesInfo) or from the **CRAN** site [Here](http://cran.r-project.org/src/contrib/Archive/)

## Data API Info Provided

* **Order Book** For every market available, currently two: Btc/Usd and Btc/Mxn.
* **Historical Trades** Every trade executed at the exchange, for both markets.
* **Actual Tick (Price)** Present ticker price of Btc/Usd and Btc/Mxn.

## How to use this RClient ?

All you need is to locate the function which provides the information you required, current supported are the following:

* **Order Book** is requested with: **meXBTOrderBook***(BtcPair)*
* **Historical Trades** is requested with: **meXBTHistoricPrices***(BtcPair,TimeZonePar,InfoSince)*
* **Actual Tick (Price)** is requested with: **meXBTTicker***(BtcPair)*

## Type of entry info and formats

* **BtcPair** : Either **btcusd** (BitCoin Vs American Dollar) or **btcmxn** (BitCoin Vs Mexican Peso)
* **InfoSince**: Parameter that specifies the tick/trade number from which you want to fetch data, 0 is from the very begining of our data and that is "2014-05-12 21:16:34 CDT" for both btcusd and and btcmxn.
* **TimeZonePar**: Fomart as stated by the **IANA** (Internet Assigned Numbers Authority) time zone database, a complete list can be found [**Here**](http://developer.oanda.com/docs/timezones.txt), and more info about **TZ DataBse** in [**Here**](https://en.wikipedia.org/wiki/Tz_database)

## Current Functions in RClient

Eg1 <- meXBTTicker("btcmxn") # meXBTTicker(BtcPair)  
Eg2 <- meXBTOrderBook("btcmxn") # meXBTOrderBook(BtcPair)  
Eg3 <- meXBTHistoricPrices("btcusd","America/Mexico\_City",650) # meXBTHistoricPrices(BtcPair,TimeZonePar,InfoSince)

## Specific HTTP Character String to fetch data manually

Or if you want/need to build your own *http* GET - POST functions, all you need is to generate character strings like the following and receive the response in **JSON** format.

#### Order Book

# HTTP Address to fetch from for Btc/Usd  
HttpAddress <- "https://data.mexbt.com/order-book/btcusd"  
# HTTP Address to fetch from for Btc/Mxn  
HttpAddress <- "https://data.mexbt.com/order-book/btcmxn"

#### Historical Trades

# HTTP Address to fetch from for Btc/Usd  
HttpAddress <- "https://data.mexbt.com/trades/btcusd?since=0"  
# HTTP Address to fetch from for Btc/Mxn  
HttpAddress <- "https://data.mexbt.com/trades/btcmxn?since=0"

#### Actual Tick (Price)

# HTTP Address to fetch from for Btc/Usd  
HttpAddress <- "https://data.mexbt.com/ticker/btcusd"  
# HTTP Address to fetch from for Btc/Mxn  
HttpAddress <- "https://data.mexbt.com/ticker/btcmxn"

## An Easy Example

This code generates a request to fetch Btc/Mxn Exchange Rate, convert the response from *JSON* format to a *data.frame* object, then modify the unix timestamp to a Human readable format to finally re-organize the columns and deliver a tidy *data.frame* ready to use for any computation.

HmeXBTBtcMxn1 <- "https://data.mexbt.com/trades/btcmxn?since=12205" # 12205 an   
HmeXBTBtcMxn2 <- getURL(HmeXBTBtcMxn1,cainfo=system.file("CurlSSL", # arbitrary  
 "cacert.pem",package="RCurl")) # Example  
HmeXBTBtcMxn3 <- data.frame(fromJSON(HmeXBTBtcMxn2))  
  
BtcMxn <- data.frame(HmeXBTBtcMxn3$tid,  
 as.POSIXct(as.numeric(as.character(HmeXBTBtcMxn3$date)), # BTC/MXN  
 origin = '1970-01-01', tz='America/Mexico\_City'), # Date  
 HmeXBTBtcMxn3$price, HmeXBTBtcMxn3$amount) # Formated  
colnames(BtcMxn) <- c("TickerID","TimeStamp","Price","Amount") # Posixct

This should return two *data.frame* objects, first **HmeXBTBtcMxn3** is in raw format, in order to you can change the *TimeStamp* with your current *Time Zone*, **BtcMxn** object is with *'America/Mexico\_City'* *Time Zone* , also the content is reorganized like the following:

|  |  |  |  |
| --- | --- | --- | --- |
| TickerID | TimeStamp | Price | Amount |
| 12205 | 2015-06-25 17:02:21 | 3736.33 | 0.99375858 |
| 12206 | 2015-06-25 17:06:07 | 3739.43 | 0.38670599 |
| 12207 | 2015-06-25 18:51:46 | 3744.02 | 0.20790000 |

**FranciscoME**: *Research & Development* - [franciscome@mexbt.com](mailto:franciscome@mexbt.com)