//+------------------------------------------------------------------+

//| Cotacao1.0.mq4 |

//| Copyright 2018, MetaQuotes Software Corp. |

//| https://www.mql5.com |

//+------------------------------------------------------------------+

#property copyright "Copyright 2018, MetaQuotes Software Corp."

#property link "https://www.mql5.com"

#property version "1.00"

#property strict

//--- input parameters

input string GravaSimouNao="Sim";

input string NomeArquivos="";

input double TempoEsperaSeg=20000;

//+------------------------------------------------------------------+

//| Expert initialization function |

//+------------------------------------------------------------------+

#include <hash.mqh>

#include <json.mqh>

int teste = 1;

int a = 0;

string nomeArquivo, dadofile, dateToSave;

string cotacoes[80000];

string webLoaded = "false";

int handle;

double dolar,cont,ctUSDcoinMarket;

int OnInit()

{

//---

string s ;

string sf = '"';

Print( "sf = " + sf);

string data=TimeToStr(TimeLocal(),TIME\_DATE);

string horario = data ;

int replaced=StringReplace(horario,".","\_");

cont = 1000;

// int x;

nomeArquivo = horario + NomeArquivos;

cotacoes[a] = "Timestamp;Moeda;AskVlrBrlxBitCoinBTC\_BRL;AskLoteBrlxBitCoinBTC\_BRL;BidVlrBrlxBitCoinBTC\_BRL;BidLoteBrazileixBTC\_BRL;" +

"Timestamp;Moeda;AskVlrBrlxBitCoinETH-BTC;AskLoteBrazileixETH-BTC;BidVlrBrazileixETH-BTC;BidLoteBrazileixETH-BTC;" +

"Timestamp;Moeda;AskVlrBrlxBitCoinETH\_brl;AskLoteBrazileixETH\_brl;BidVlrBrazileixETH\_brl;BidLoteBrazileixETH\_brl;" +

"Timestamp;Moeda;AskVlrBrlxBitCoinLTC\_brl;AskLoteBrazileixLTC\_brl;BidVlrBrazileixLTC\_brl;BidLoteBrazileixLTC\_brl;" +

"Timestamp;Moeda;AskVlrBrlxBitCoinLTC\_btc;AskLoteBrazileixLTC\_btc;BidVlrBrazileixLTC\_btc;BidLoteBrazileixLTC\_btc;" +

"Timestamp;Moeda;AskVlrNgCnBTCBRL;AskLoteNgCnBTCBRL;BidVlrNgCnBTCBRL;BidLoteNgCnBTCBRL;" +

"Timestamp;Moeda;AskVlrNegCoin\_tlc\_brl;AskLoteNegCoin\_tlc\_brl;BidVlrNgCnLTCBRL;BidLoteNgCnLTCBRL;" +

"Timestamp;Moeda;AskVlrMercBTCBRL;AskLoteMercBTCBRL;BidVlrMercBTCBRL;BidLoteMercBTCBRL;" +

"Timestamp;Moeda;AskVlrMercLTCBRL;AskLoteMercLTCBRL;BidVlrMercLTCBRL;BidLoteMercLTCBRL;" +

"Timestamp;Moeda;AskVlrFoxBTCBRL;AskLoteFoxBTCBRL;BidVlrFoxBTCBRL;BidLoteFoxBTCBRL;" +

"BrokerBtcAsk;BrokerBtcBid;BrokerEthAsk;BrokerEthBid;BrokerLTCAsk;BrokerLTCBid;BTCAsk;BTCBid";

a = a + 1;

//---

return(INIT\_SUCCEEDED);

}

//+------------------------------------------------------------------+

//| Expert deinitialization function |

//+------------------------------------------------------------------+

void OnDeinit(const int reason)

{

//---

}

//+------------------------------------------------------------------+

//| Expert tick function |

//+------------------------------------------------------------------+

void OnTick()

{

//---

double vAskBTC = MarketInfo("BTCUSD",MODE\_ASK);

double vAskETH = MarketInfo("ETH",MODE\_ASK);

double vAskLTC = MarketInfo("LTC\_Mini",MODE\_ASK);

double vBidBTC = MarketInfo("BTCUSD",MODE\_BID);

double vBidETH = MarketInfo("ETH",MODE\_BID);

double vBidLTC = MarketInfo("LTC\_Mini",MODE\_BID);

string r1, r2, r3, r4, r5,r6,r7,r8,r9,r10;

double r0;

// for(x=1;x<=20; x++) {

if ( cont > 99 ) {

//r0 = APIDolar("GET", "https://api.coinmarketcap.com/v1/ticker/bitcoin/?convert=brl");

Print("teste if");

teste = 1;

while(teste == 1){

r0 = APIDolar("GET", "http://api.promasters.net.br/cotacao/v1/valores?moedas=USD&alt=json");

ctUSDcoinMarket = CoinMArketDolarApi("GET", "https://api.coinmarketcap.com/v1/ticker/bitcoin/?convert=brl");

Print("valo dolar uol comercial = " + r0);

dolar = r0 \* 1.05;

Print("valo dolar uol turismo com 5% do comercial = " + dolar);

Print("valo dolar uol coinmarket = " + ctUSDcoinMarket);

if ( dolar > 1 && ctUSDcoinMarket > 1){

teste = 3;

}

}

cont = 1;

}

cont = cont + 1 ;

Print("cotacao dolar uol = " + dolar);

Print("cotacao dolar coinMarket = " + ctUSDcoinMarket);

r1 = WebAPI("GET", "https://braziliex.com/api/v1/public/orderbook/btc\_brl", "BrlxBTCBRL", dolar,ctUSDcoinMarket );

r2 = WebAPI("GET", "https://braziliex.com/api/v1/public/orderbook/eth\_btc", "BrlxETHBTC", dolar,ctUSDcoinMarket);

r3 = WebAPI("GET", "https://braziliex.com/api/v1/public/orderbook/eth\_brl", "BrlxETHBRL", dolar,ctUSDcoinMarket);

r4 = WebAPI("GET", "https://braziliex.com/api/v1/public/orderbook/ltc\_brl", "BrlxLTCBRL",dolar,ctUSDcoinMarket);

r5 = WebAPI("GET", "https://braziliex.com/api/v1/public/orderbook/ltc\_btc", "BrlxLTCBTC", dolar,ctUSDcoinMarket);

r6 = WebAPI("GET", "https://broker.negociecoins.com.br/api/v3/btcbrl/orderbook", "NgCnBTCBRL", dolar,ctUSDcoinMarket);

r7 = WebAPI("GET", "https://broker.negociecoins.com.br/api/v3/ltcbrl/orderbook", "NgCnLTCBRL", dolar,ctUSDcoinMarket);

r8 = WebAPI("GET", "https://www.mercadobitcoin.net/api/btc/orderbook/", "MercBTCBRL", dolar,ctUSDcoinMarket);

r9 = WebAPI("GET", "https://www.mercadobitcoin.net/api/ltc/orderbook/", "MercLTCBRL", dolar,ctUSDcoinMarket);

r10 = WebAPI("GET", "https://api.blinktrade.com/api/v1/BRL/orderbook?crypto\_currency=BTC", "FoxBTCBRL", dolar,ctUSDcoinMarket);

Sleep(TempoEsperaSeg);

// }

}

//+------------------------------------------------------------------+

string WebAPI (string metodo, string url, string coinpar,double ctdolar, double ctUSDMarket){

string cookie=NULL,headers;

char post[],result[];

int res;

double ressss[];

string strResult,nameFile;

string erro, tempo, dados;

double vAskBTC = MarketInfo("BTCUSD",MODE\_ASK);

double vAskETH = MarketInfo("ETH",MODE\_ASK);

double vAskLTC = MarketInfo("LTC\_Mini",MODE\_ASK);

double vBidBTC = MarketInfo("BTCUSD",MODE\_BID);

double vBidETH = MarketInfo("ETH",MODE\_BID);

double vBidLTC = MarketInfo("LTC\_Mini",MODE\_BID);

tempo = TimeLocal();

//--- to enable access to the server, you should add URL "https://www.google.com/finance"

//--- in the list of allowed URLs (Main Menu->Tools->Options, tab "Expert Advisors"):

//--- Reset the last error code

ResetLastError();

//--- Loading a html page from Google Finance

int timeout=5000; //--- Timeout below 1000 (1 sec.) is not enough for slow Internet connection

res=WebRequest("GET",url,cookie,NULL,timeout,post,0,result,headers);

//--- Checking errors

if(res==-1)

{

Print("Error in WebRequest. Error code =",GetLastError());

//--- Perhaps the URL is not listed, display a message about the necessity to add the address

// MessageBox("Add the address '"+url+"' in the list of allowed URLs on tab 'Expert Advisors'","Error",MB\_ICONINFORMATION);

webLoaded = "false";

}

else

{

//--- Load successfully

PrintFormat(coinpar + " - The file has been successfully loaded, File size =%d bytes.",ArraySize(result));

//--- Save the data to a file

//C:\Users\UserAd\AppData\Roaming\MetaQuotes\Terminal\88A7C6C356B9D73AC70BD2040F0D9829\MQL4\Files

webLoaded = "true";

}

if ( webLoaded == "true") {

printf("funcinouuu");

for(int i=0;i<ArraySize(result);i++)

{

strResult += CharToStr(result[i]);

}

Print("Resultados : " + strResult);

string s = strResult ;

JSONParser \*parser = new JSONParser();

JSONValue \*jv = parser.parse(s);

if (jv == NULL) {

Print("error:"+(string)parser.getErrorCode()+parser.getErrorMessage());

} else {

Print("PARSED:"+jv.toString());

if (jv.isObject()) { // check root value is an object. (it can be an array)

JSONObject \*jo = jv;

//"Timestamp;AskValorBrlxBitCoin;AskLoteBrazileix;BidValorBrazileix;BidLoteBrazileix;AskAvaEth;BidAvaEth"

if( (coinpar == "NgCnBTCBRL") || (coinpar == "NgCnLTCBRL")){

dados = tempo + ";" + coinpar + ";" + jo.getArray("ask").getObject(0).getDouble("price") + ";" + jo.getArray("ask").getObject(0).getDouble("quantity") + ";" +

jo.getArray("bid").getObject(0).getDouble("price") + ";" + jo.getArray("bid").getObject(0).getDouble("quantity");

}

if( (coinpar == "BrlxBTCBRL") || (coinpar == "BrlxETHBTC") || (coinpar == "BrlxETHBRL") || (coinpar == "BrlxLTCBRL") || (coinpar == "BrlxLTCBTC")){

dados = tempo + ";" + coinpar + ";" + jo.getArray("asks").getObject(0).getDouble("price") + ";" + jo.getArray("asks").getObject(0).getDouble("amount") + ";" +

jo.getArray("bids").getObject(0).getDouble("price") + ";" + jo.getArray("bids").getObject(0).getDouble("amount");

}

if( (coinpar == "MercBTCBRL") || (coinpar == "MercLTCBRL") || (coinpar == "FoxBTCBRL") ){

dados = tempo + ";" + coinpar + ";" + jo.getArray("asks").getArray(0).getDouble(0) + ";" + jo.getArray("asks").getArray(0).getDouble(1) + ";" +

jo.getArray("bids").getArray(0).getDouble(0) + ";" + jo.getArray("bids").getArray(0).getDouble(1);

}

a = a + 1;

delete jv;

}

delete parser;

}

dados = dados + ";" + vAskBTC + ";" + vBidBTC + ";" + vAskETH + ";" + vBidETH + ";" + vAskLTC + ";" + vBidLTC + ";" + Ask + ";" + Bid + ";" + ctdolar + ";" + ctUSDMarket;

StringReplace(dados,".",",");

nameFile = nomeArquivo + coinpar + ".csv";

handle=FileOpen(nameFile, FILE\_CSV|FILE\_WRITE|FILE\_READ, ';');

if(handle>0)

{

FileSeek(handle, 0, SEEK\_END);

FileWrite(handle, dados);

FileClose(handle);

}

return dados;

}else {

erro = tempo + ";" + " " + ";" + " " + ";" + " " + ";" + " " + ";" + " ";

return erro;

}

}

double APIDolar (string metodo, string url){

string cookie=NULL,headers;

char post[],result[];

int res;

double ressss[];

string strResult,nameFile;

string erro, tempo, dados;

tempo = TimeLocal();

//--- to enable access to the server, you should add URL "https://www.google.com/finance"

//--- in the list of allowed URLs (Main Menu->Tools->Options, tab "Expert Advisors"):

//--- Reset the last error code

ResetLastError();

//--- Loading a html page from Google Finance

int timeout=5000; //--- Timeout below 1000 (1 sec.) is not enough for slow Internet connection

res=WebRequest("GET",url,cookie,NULL,timeout,post,0,result,headers);

//--- Checking errors

if(res==-1)

{

Print("Error in WebRequest. Error code =",GetLastError());

//--- Perhaps the URL is not listed, display a message about the necessity to add the address

// MessageBox("Add the address '"+url+"' in the list of allowed URLs on tab 'Expert Advisors'","Error",MB\_ICONINFORMATION);

webLoaded = "false";

}

else

{

//--- Load successfully

PrintFormat(" - The file has been successfully loaded, File size =%d bytes.",ArraySize(result));

//--- Save the data to a file

//C:\Users\UserAd\AppData\Roaming\MetaQuotes\Terminal\88A7C6C356B9D73AC70BD2040F0D9829\MQL4\Files

webLoaded = "true";

}

if ( webLoaded == "true") {

printf("funcinouuu");

for(int i=0;i<ArraySize(result);i++)

{

strResult += CharToStr(result[i]);

}

Print("Resultados : " + strResult);

string s ;

StringReplace(strResult,"valores","dados");

dados = StringSubstr(StringSubstr(strResult,StringFind(strResult,"valor",10),13),7,6);

Print("dados =" + dados + "demais");

return dados;

}else {

erro = tempo + ";" + " " + ";" + " " + ";" + " " + ";" + " " + ";" + " ";

return erro;

}

}

double CoinMArketDolarApi (string metodo, string url){

string cookie=NULL,headers;

char post[],result[];

int res;

double ressss[];

string strResult,nameFile;

string erro, tempo, dados;

tempo = TimeLocal();

//--- to enable access to the server, you should add URL "https://www.google.com/finance"

//--- in the list of allowed URLs (Main Menu->Tools->Options, tab "Expert Advisors"):

//--- Reset the last error code

ResetLastError();

//--- Loading a html page from Google Finance

int timeout=5000; //--- Timeout below 1000 (1 sec.) is not enough for slow Internet connection

res=WebRequest("GET",url,cookie,NULL,timeout,post,0,result,headers);

//--- Checking errors

if(res==-1)

{

Print("Error in WebRequest. Error code =",GetLastError());

//--- Perhaps the URL is not listed, display a message about the necessity to add the address

// MessageBox("Add the address '"+url+"' in the list of allowed URLs on tab 'Expert Advisors'","Error",MB\_ICONINFORMATION);

webLoaded = "false";

}

else

{

//--- Load successfully

PrintFormat(" - The file has been successfully loaded, File size =%d bytes.",ArraySize(result));

//--- Save the data to a file

//C:\Users\UserAd\AppData\Roaming\MetaQuotes\Terminal\88A7C6C356B9D73AC70BD2040F0D9829\MQL4\Files

webLoaded = "true";

}

if ( webLoaded == "true") {

printf("funcinouuu");

for(int i=0;i<ArraySize(result);i++)

{

strResult += CharToStr(result[i]);

}

Print("Resultados : " + strResult);

JSONParser \*parser = new JSONParser();

JSONValue \*jv = parser.parse(strResult);

if (jv == NULL) {

Print("error:"+(string)parser.getErrorCode()+parser.getErrorMessage());

} else {

Print("PARSED:"+jv.toString());

string ds = jv.toString();

Print("ds:"+ ds);

Print ("price\_usd 1 ="+ StringSubstr(ds,StringFind(ds,"price\_usd",0),20)) ;

Print ("price\_brl 1 ="+ StringSubstr(ds,StringFind(ds,"price\_brl",0),20)) ;

Print ("price\_usd ="+ StringSubstr(StringSubstr(ds,StringFind(ds,"price\_usd",0),30),14,8)) ;

Print ("price\_brl ="+ StringSubstr(StringSubstr(ds,StringFind(ds,"price\_brl",0),30),14,7)) ;

dados = StrToDouble(StringSubstr(StringSubstr(ds,StringFind(ds,"price\_brl",0),30),14,8)) / StrToDouble( StringSubstr(StringSubstr(ds,StringFind(ds,"price\_usd",0),30),14,7));

Print ("dados divido ="+ dados) ;

delete parser;

delete jv;

}

return dados;

}else {

erro = tempo + ";" + " " + ";" + " " + ";" + " " + ";" + " " + ";" + " ";

return erro;

}

}

**Cenário 1 :** Captura cotações MercadoBitcoin e AvaFX moedas BTCxBRL, BTCxDolar

**Quando** o sistema acionar a API MercadoBitcoin

**Quando** o sistema acionar a base de dados Avafax

**Quando** o sistema acionar a API Reuters

**Dado que** foi informado o par de moeda BTCxBRL e BTCxUSD e USD respectivamente

**Então** o sistema deve capturar o valor da cotação BTCxBRL, BTCxUSD e Cotação Dolar Comercial

**Ação** gravar na base de dados as informações de cotação de valor do dólar, valor do real, valor do BTC em real e valor do BTC em dólar.

Cenário 2 : Captura cotações Negcoin e AvaFX moedas BTCxBRL, BTCxDolar

**Quando** o sistema acionar a API Negcoin

**Quando** o sistema acionar a base de dados Avafax

**Quando** o sistema acionar a API Reuters

**Dado que** foi informado o par de moeda BTCxBRL e BTCxUSD e USD respectivamente

**Entao** o sistema deve capturar o valor da cotação BTCxBRL, BTCxUSD e Cotação Dolar Comercial

**Ação** gravar na base de dados as informações de cotação de valor do dólar, valor do real, valor do BTC em real e valor do BTC em dólar.

Cenário 3 : Captura cotações Braziliex e AvaFX moedas BTCxBRL, BTCxDolar

**Quando** o sistema acionar a API Braziliex

**Quando** o sistema acionar a base de dados Avafax

**Quando** o sistema acionar a API Reuters

**Dado que** foi informado o par de moeda BTCxBRL e BTCxUSD e USD respectivamente

**Entao** o sistema deve capturar o valor da cotação BTCxBRL, BTCxUSD e Cotação Dolar Comercial

**Ação** gravar na base de dados as informações de cotação de valor do dólar, valor do real, valor do BTC em real e valor do BTC em dólar.

Cenário 4 : Falha de conexão a API

**Quando** o sistema acionar a API das corretoras

**Dado que** foi informado o par de moeda BTCxBRL

**Entao** repetir em até 3 tentativas a chamada da api que não respondeu

**Ação** o sistema deve gravar o erro na base de dados, com data, hora , código da correta que apresentou o erro .

**Cenário 5 :** Falha de conexão base de Dados Exchange

**Quando** o sistema acionar a base de dados da exchange

**Dado que** foi informado o par de moeda BTCxUSD

**Entao** repetir em até 3 tentativas a chamada da api que não respondeu

**Ação** o sistema deve gravar o erro na base de dados, com data, hora , código da correta que apresentou o erro .

**Cenário 6 :** Falha de conexão base de dados da Reuters

**Quando** o sistema acionar a API da Reuters e houver falha de acesso

**Dado que** foi informado a cotação do Dolar em relação ao Real - BRLxUSD

**Entao** o sistema deve recuperar o ultima cotação armazenada e utiliza-la como cotação do momento

**Ação** o sistema deve gravar o erro na base de dados, com data, hora , código da correta que apresentou o erro .