Analytic System for Currency Rates

Symbol = USDEUR;

Queue Rates(size 5);

Queue RatesTimestamp(size 5);

currTimestamp;

*external\_device;*

While True{

setTimeout(getData, 60000) **//system call every minute**

function geData(){

fetch(<https://api.exchangeratesapi.io/v1/latest>

? access\_key = API\_KEY

& base = USD

& symbols = EUR)

Object data = Response{

{

"success": true,

"timestamp": time,

"base": "USD",

"date": "date",

"rates": {

"EUR": rate

}

}

currTimestamp = data.Timestamp;

*external\_device* = data; **//prevent data loss if system restarts**

**//If hard drive has 5 rates delete the first added rate//**

}

If (*external\_device not empty){*

*For (data : external\_device){*

*If ( data.*Timestamp not in RatesTimestamp.timestamp){

RatesTimestamp.add(data.timestamp);

Rates.add(*data.rate*); // **use FIFO to delete first element if size of queue is 5**

**}**

**}**

**}**

}

If (Rates.length == 5){

**// StatisticCalculation**

currentRate = Rates[4]; **// get last/current value**

minNum = Min(rates);

maxNum = Max(rates);

avg = avg(rates);

**//Statistic Persistent Storage**

**Store in file:**

< currTimestamp > < currentRate > \

< minNum > \

< maxNum > \

< avg >

}

}