**ANUBHAV MISHRA**

**ASSIGNMENT-2**

**BHARATI VIDYAPEETH’s COLLEGE OF ENGINEERING , NEW DELHI**

Q = Create a post request from another JS file/application and hit the /transaction on Localhost:3000

Blockchain.js

-------------------------------------------------------------------------------------------------------------------------------

var sha256 = require('sha256');

function Blockchain(){

this.chain = [];

this.pendingTransactions = [];

this.createNewBlock("adsfsfsfgdghh7675gvbv","4356fhcgcfcvcbvnnnvnv",434427);

}

//creation and addition of the block to the chain

Blockchain.prototype.createNewBlock = function(hashOfBlock,previousHash,nonceOfBlock){

const newBlock = {

index:this.chain.length+1,

timestamp:Date.now(),

transactions:this.pendingTransactions,

hash:hashOfBlock,

previousBlockHash:previousHash,

nonce:nonceOfBlock

};

this.chain.push(newBlock);

this.pendingTransactions=[];

return newBlock;

}

//create mechanism to create and add new transactions

Blockchain.prototype.createNewTransaction = function(sender,recipient,amount){

const newTransaction = {

sender:sender,

recipient:recipient,

amount:amount

};

this.pendingTransactions.push(newTransaction);

return newTransaction;

}

Blockchain.prototype.addTransactionToPendingTransactions=function(transactionObj){

this.pendingTransactions.push(transactionObj);

return this.getLastBlock()['index']+1;

};

//to get getLastBlock information

Blockchain.prototype.getLastBlock=function(){

return this.chain[this.chain.length-1];

};

Blockchain.prototype.hashBlock = function(previousBlockHash,nonce,currentBlockData){

const dataAsString = previousBlockHash + nonce.toString() + JSON.stringify(currentBlockData);

const hash = sha256(dataAsString);

return hash;

}

Blockchain.prototype.proofOfWork = function(previousBlockHash,currentBlockData){

let nonce = 0;

var hash = this.hashBlock(previousBlockHash,nonce,currentBlockData);

//framing algo for math problem

while(hash.substring(0,6)!=="000000"){

nonce++;

var hash = this.hashBlock(previousBlockHash,nonce,currentBlockData);

}

return nonce;

}

//this line is used to export the whole file to any other file which requries it

module.exports = Blockchain;

api.js

var express = require('express') //importing express framework to make this server side app

var app = express()

//importing the whole blockchain functionality here

var Blockchain = require('./blockchain');

var bitcoin = new Blockchain();

const bodyParser=require('body-parser');

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({extended:false}));

//"app" now represents a server side framework

//sample get call to display on browser its request and response

app.get('/', function(req,res){

res.send('the get call for / is working');

});

//created an end point which will give the blockchain info at a node

app.get('/blockchain', function(req,res){

res.send(bitcoin);

});

app.post('/transaction',function(req,res){

const newTransaction=req.body;

var blockIndex=bitcoin.addTransactionToPendingTransactions(newTransaction);

res.json({note:`Transaction will be added in block ${blockIndex}.`});

});

//get call like your mining calling

app.get('/mine',function(req,res){

res.send("this end point is used to mine the blocks");

});

//to make the server listen at 3000like app.listen(port)

app.listen(3000, function(){

console.log('listening at the corresponing port 3000');

});

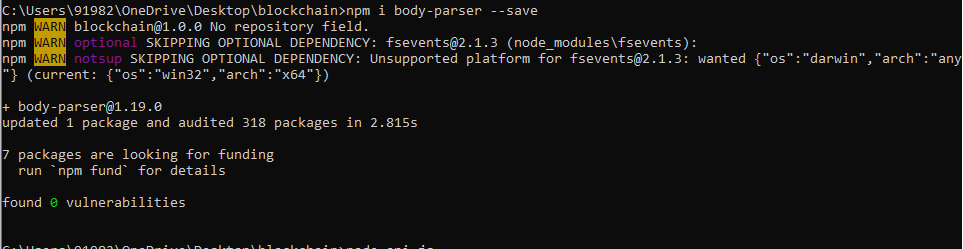
//Create a new get function in your api.js, when you hit localhost:3000/sample

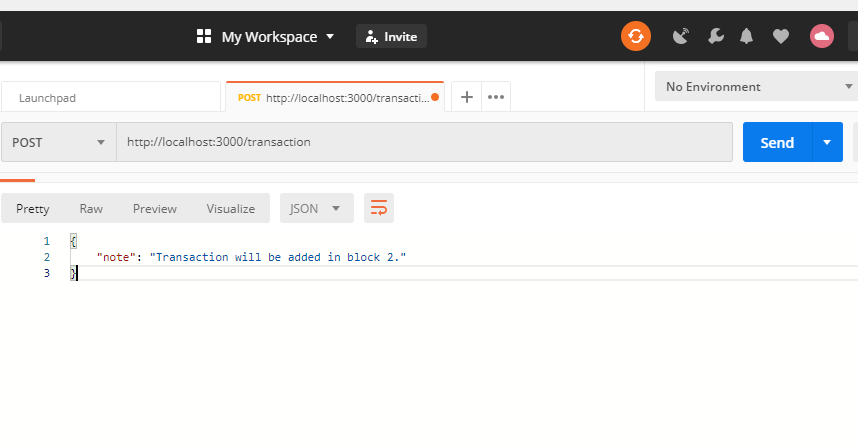
//on the browser it must show "this is my sample get call"

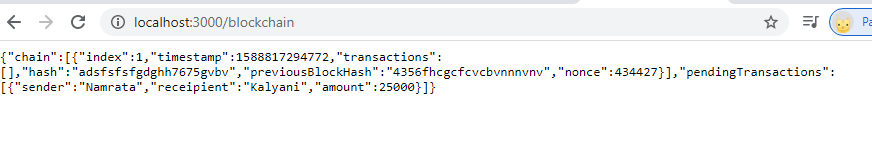
app.get('/sample',function(req,res){

res.send("this is my sample get call");

});







Get all the code files in dev folder and add one script in package.json

{

"name": "blockchain",

"version": "1.0.0",

"description": "blockchain",

"main": "blockchain.js",

"dependencies": {

"body-parser": "^1.19.0",

"express": "^4.17.1",

"install": "^0.13.0",

"nodemon": "^2.0.3",

"sha256": "^0.2.0"

},

"devDependencies": {},

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1",

"start":"nodemon --watch dev -e js dev/api.js"

},

"author": "anubhav",

"license": "ISC"

}

Fire up the command in cmd “npm start”

