const Block = require('./block');

const cryptoHash = require('./crypto-hash');

class Blockchain {

constructor(){

this.chain = [Block.genesis()];

}

addBlock({data}){

const newBlock = Block.mineBlock({

lastBlock: this.chain[this.chain.length - 1],

data

});

this.chain.push(newBlock);

}

replaceChain(chain) {

if (chain.length <= this.chain.length){

console.error('incoming chain must be longer');

return;

}

if (!Blockchain.isValidChain(chain)){

console.error('incoming chain must be valid');

return;

}

console.log('replacing chain with', chain);

this.chain= chain;

}

static isValidChain(chain) {

if (JSON.stringify(chain[0]) !== JSON.stringify(Block.genesis())){

return false

};

for (let i=1; i<chain.length; i++){

//const block = chain[i];

const {timestamp, lastHash, hash, data} = chain[i];

const actualLastHash = chain[i-1].hash;

//const {timestamp, lastHash, hash, data} = block;

if (lastHash !== actualLastHash) return false;

const validatedHash = cryptoHash(timestamp, lastHash, data);

if (hash !== validatedHash) return false;

}

return true;

}

}

module.exports = Blockchain;