var io = require('socket.io')(server);

function updateBTCPrice() {

http.get({

host: 'api.coindesk.com',

path: '/v1/bpi/currentprice.json'

},

function(response) {

var body = '';

response.on('data', function(d) { body += d; });

response.on('end', function() {

var parsed = JSON.parse(body);

console.log(parsed.bpi.USD.rate);

io.emit('btc\_price\_change', parsed.bpi.USD.rate);

});

}

);

}

io.on('connection', function (socket) {

updateBTCPrice();

});

// If wanted

setInterval(updateBTCPrice, 100000);

var bitcoinTransaction = require('bitcoin-transaction');

// BTC

//NEW ADDRESS: cT44ADFtj3HBVzCgAq2CEqPzc7ddFzt5UtFra4FNQA3zWCmcMjTU

//NEW PUBLIC ADDRESS: mjeBxXLF7FBgFMBCNzLuGsaWLZwgemkQtB - Balance = 0.48991 BTC

//New ADDRESS 2: cTmctuuAC9cVScvcAKK1aPx22ZWwpovdbYX9VrLpYHD1351HsQPJ

//NEW PUBLIC ADDRESS2: mkiKq29QCizK7qBn25m6tBZPkBnNZmrqT5 - Balance = 0.04 BTC

//New ADDRESS 3: cU8nsaoRV4NAR1dmtggEvDnuiG64r5kqq8Vs93Q9fv6FQ43cG8TW

//NEW PUBLIC ADDRESS2: mrs26XNm7wZggDdt8yGV8rLXhLWSA6Fcm8 - Balance = 0.9375 BTC

function findConfirmedPayment(){

AM.findConfirmedPayment(foundConfirmedPayment);

}

var tempUser;

function foundConfirmedPayment(user, clientWallet){

console.log("Client Wallet Address: " + clientWallet);

// Save username

tempUser = user;

// Send bitcoins

sendBTC(clientWallet, 0.01);

}

// Gets called with text=tx\_id

function transactionCallback(tx\_id){

console.log("Bitcoins Sent!");

//Save txID in appropriate place in database

AM.saveTXIDWithUsername(tempUser, tx\_id);

}

function sendBTC(to, amount){

//Send all btc

var from = "mrs26XNm7wZggDdt8yGV8rLXhLWSA6Fcm8";

var privKeyWIF = "cU8nsaoRV4NAR1dmtggEvDnuiG64r5kqq8Vs93Q9fv6FQ43cG8TW";

bitcoinTransaction.getBalance(from, { network: "testnet" }).then((balanceInBTC) => {

console.log("amount " + amount + " balance " + balanceInBTC);

if (balanceInBTC > amount){

var txID = bitcoinTransaction.sendTransaction({

from: from,

to: to,

privKeyWIF: privKeyWIF,

btc: amount,

network: "testnet"

}, transactionCallback);

}

else

{

console.log("Not enough funds");

}

});

}

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

console.log("GET ORDERS");

var query = { "orders" : { $exists : true } };

AM.getSpecificRecords(query, function(e, result) {

console.log(result);

res.render('orders', {

title : 'Control Panel',

countries : CT,

udata : req.session.user,

orders: result

});

});

});

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

if(!req.body.confirmCheckbox){

console.log("POST ORDERS: No checkboxes selected");

}

AM.confirmPayment(req.body.confirmCheckbox, function(e, result){

console.log("POST ORDERS: confirmPayment callback");

findConfirmedPayment();

res.status(200).send('ok');

});

});

// Purchase //

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

console.log("GET PURCHASE");

res.render('purchase', {

title : 'Control Panel',

countries : CT,

udata : req.session.user

});

});

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

console.log("POST PURCHASE");

var udata = req.session.user;

AM.placeOrder({

name : udata.name,

email : udata.email,

user : udata.user,

pass : udata.password,

country : udata.country,

orderNumber : udata.orderNumber,

dateIssued : udata.dateIssued,

btc : req.body['btc'],

gbp : req.body['gbp'],

btc\_address : req.body['btc\_address']

}, function(e){

if (e){

console.log("/Purchase POST placeOrder callback SUCCESS: " + e);

res.status(200).send('ok');

} else{

console.log("/Purchase POST placeOrder callback FAILURE: " + e);

res.status(400).send('error');

}

});

});

// logged-in user homepage //

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

if (req.session.user == null){

console.log("Not Logged in");

// //if user is not logged-in redirect back to login page //

if(req.cookies.user)

{

console.log("Cookie found for: " + req.cookies.user);

AM.manualLogin(req.cookies.user, req.cookies.pass, function(o){

if (o != null){

console.log("Autologin success");

req.session.user = o;

res.redirect('/home');

} else{

console.log("Autologin not successful");

res.render('home', {

title : 'Control Panel',

countries : CT

});

}

});

}

else{

res.render('home', {

title : 'Control Panel',

countries : CT

});

}

} else{

console.log("Logged IN!");

res.render('home', {

title : 'Control Panel',

countries : CT,

udata : req.session.user

});

}

});

// When login form submitted

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

console.log("POST Not Logged in");

if(req.body['country'] == undefined){

// Attempt to login

AM.manualLogin(req.body['user'], req.body['pass'], function(e, o){

if (!o){

res.status(400).send(e);

} else{

req.session.user = o;

if (req.body['remember-me'] == 'true'){

res.cookie('user', o.user, { maxAge: 900000 });

res.cookie('pass', o.pass, { maxAge: 900000 });

}

res.status(200).send(o);

}

});

}else{

console.log("1");

AM.addNewAccount({

name : req.body['name'],

email : req.body['email'],

user : req.body['user'],

pass : req.body['pass'],

country : req.body['country']

}, function(e){

if (e){

console.log("creation failed");

res.status(400).send(e);

} else{

console.log("account created");

res.cookie('user', req.body['user'], { maxAge: 900000 });

res.cookie('pass', req.body['pass'], { maxAge: 900000 });

res.status(200).send('ok');

}

});

}

});

exports.placeOrder = function(newData, callback)

{

accounts.findOne({user:newData.user}, function(e, o) {

if (o){

accounts.findOne({email:newData.email}, function(e, o) {

if (o){

var orderNumber = 1;

if (newData.orderNumber) orderNumber += newData.orderNumber;

var order = [{ orderNumber: orderNumber , btc: newData.btc, gbp: newData.gbp, btc\_address: newData.btc\_address, dateIssued: moment().format('MMMM Do YYYY, h:mm:ss a'), confirmed: 'false' }]

accounts.update(o, {$set : {"orders":order}} , function(err, res) {

if (err) throw err;

console.log("1 document updated");

callback(1);

});

} else{

console.log("Email not found");

}

});

} else{

console.log("Username not found");

}

});

}

exports.confirmPayment = function(selectedIndex, callback)

{

console.log("AM");

// var query = { "orders" : { $exists : true } };

// accounts.find(query).toArray(function(e, result) {

// console.log(result[selectedIndex]);

// //result[selectedIndex].orders[0].confirmed = true;

// //accounts.update ({\_id: '123'}, { '$set': {"friends.0.emails.0.email" : '2222'} });

// //result.orders[0].confirmed = true;

// // accounts.update(result, function(err, res) {

// // if (err) throw err;

// // console.log("1 document updated");

// // callback(1);

// // });

// });

var selected\_username = null;

var selected\_email = null;

var selected\_orderNumber = null;

var query = { "orders" : { $exists : true } };

accounts.find(query).toArray(function(e, result){

//First - identify order by username, email and ordernumber

if(!selectedIndex) {

console.log("nothing selected");

return;

}

selected\_username = result[selectedIndex].user;

selected\_email = result[selectedIndex].email;

selected\_orderNumber = result[selectedIndex].orders[0].orderNumber;

accounts.findOne({user:selected\_username}, function(e, o) {

if (e){

callback(e, null);

} else{

console.log(o);

o.orders[0].confirmed = "true";

accounts.save(o, {safe: true}, callback);

}

});

});

}

exports.findConfirmedPayment = function(callback)

{

var query = { "orders" : { $exists : true } };

// Find confirmed payment

accounts.findOne(query, function(e, o) {

if (e){

callback(e, null);

} else{

// return username if the order has been confirmed and if tx\_id field doesnt exist

if(o.orders[0].confirmed == "true" && !o.orders[0].tx\_id ){

callback(o.user, o.orders[0].btc\_address);

}

}

});

}

// Return username of user with confirmed payment

exports.saveTXIDWithUsername = function(tempUser, tx\_id)

{

console.log("Saving tx\_id :" + tx\_id + " For user: " + tempUser);

var query = { "user" : tempUser };

// Find confirmed payment

accounts.findOne(query, function(e, o) {

if (e){

callback(e, null);

} else{

// save tx\_id

o.orders[0].tx\_id = tx\_id;

accounts.save(o, {safe: true});

console.log("Saved tx\_id!");

}

});

}

head

title Massively by HTML5 UP

meta(charset='utf-8')

meta(name='viewport', content='width=device-width, initial-scale=1, user-scalable=no')

link(rel='stylesheet', href='/css/font-awesome.min.css')

link(rel='stylesheet', href='/css/main.css')

link(rel='stylesheet', href='/css/noscript.css')

nav

.top

| Welcome to Bitcoin!

if udata

| Logged in as,

p= udata.user

else

a(href='/home#login') Click here to login

html

body.is-loading

#wrapper.fade-in

include navigation.jade

#main

#myModal.modal

.modal-content

span.close ×

p Confirmed order

a(href="")

if !user.orders

- user.orders = orders

if user.orders

form(method="post")#confirmOrder-form

if user.admin == "true"

h2.h2\_small.text-center Orders [Admin]

else

h2.h2\_small.text-center Orders

.orderMainHolder

- var iteration = 0;

if user.admin == "true"

table.tg

tr

th.tg-u6aq Username

th.tg-u6aq Email

th.tg-u6aq Order Number

th.tg-u6aq BTC

th.tg-u6aq GBP

th.tg-u6aq Date Issued

th.tg-u6aq Confirm

each field in orders

if field.orders[0].confirmed == "false"

tr

td.tg-us36= field.user

td.tg-us36= field.email

td.tg-us36= field.orders[0].orderNumber

td.tg-us36= field.orders[0].btc

td.tg-us36= field.orders[0].gbp

td.tg-us36= field.orders[0].dateIssued

td.tg-us36

input(type="radio", name="confirmCheckbox", value= iteration++)

if iteration == 0

tr

td.tg-us36

td.tg-us36

td.tg-us36

td.tg-us36

td.tg-us36

td.tg-us36

td.tg-us36

else

table.tg

tr

th.tg-u6aq Order Number

th.tg-u6aq BTC

th.tg-u6aq GBP

th.tg-u6aq Rate

th.tg-u6aq Date Issued

th.tg-u6aq Blockchain Transaction ID

th.tg-u6aq.orderPaidColomn Paid?

tr

td.tg-us36= user.orders[0].orderNumber

td.tg-us36= user.orders[0].btc

td.tg-us36= user.orders[0].gbp

td.tg-us36 -

td.tg-us36= user.orders[0].dateIssued

td.tg-us36= user.orders[0].tx\_id

td.tg-us36= user.orders[0].confirmed

input(type='submit', value='Submit')

#account-form-container.center-vertical

form(method="post")#purchase-form.form-horizontal.well

.bottom\_article\_header

.form-group.article\_input

label.control-label.col-sm-2 BTC

.col-sm-3

input.form-control#btc\_input.article\_input\_field(type='text', name='btc', value= user.btc)

.form-group.article\_input

label.control-label.col-sm-2 GBP

.col-sm-9

input.form-control#gbp\_input.article\_input\_field(type='text', name='gbp', value= user.gbp)

.form-group.article\_input

label.control-label.col-sm-2 Your Bitcoin Wallet Address

.col-sm-9

input.form-control#btcAddress\_input.article\_input\_field(type='text', name='btc\_address', value= user.btc\_address)

hr

.form-buttons.article\_buttons

button(type='button')#account-form-btn1.btn.btn-default.\_purchase\_cancle\_Button

button(type='submit')#account-form-btn2.btn.btn-primary.\_purchase\_submit\_Button

.clearfix

var socket = io();

var btc\_price = 1;

socket.on('connect', function(){

$('.btc\_price').html("1BTC = " + btc\_price);

});

socket.on('disconnect', function(){});

socket.on('btc\_price\_change', function(msg){

console.log(msg);

btc\_price = msg;

$('.btc\_price').html(btc\_price);

});

$('#confirmOrder-form').ajaxForm({

beforeSubmit : function(formData, jqForm, options){

console.log("Confirm Order before submit");

return 1;

},

success : function(responseText, status, xhr, $form){

console.log("Confirm Order Success");

$('.modal').show();

// Display confirmation

},

error : function(e){

console.log("Confirm Order Error");

}

});

(function($) {

$(document).ready(function() {

console.log("test");

// Set navigation

$('#homeNavButton').removeClass('active');

$('#purchaseNavButton').removeClass('active');

$('#ordersNavButton').addClass('active');

console.log($("#main").length);

$( ".close" ).click(function() {

$(".modal").hide();

location.reload();

});

})

$(document).ready(function() {

// Set navigation

$('#homeNavButton').removeClass('active');

$('#purchaseNavButton').addClass('active');

})

$('#purchase-form').ajaxForm({

beforeSubmit : function(formData, jqForm, options){

console.log("purchase before submit");

return 1;

},

success : function(responseText, status, xhr, $form){

console.log("purchase success");

$('.modal').show();

// Display confirmation

},

error : function(e){

console.log("purchase error");

}

});

$( ".close" ).click(function() {

$(".modal").hide();

});

$('#btc\_input').on('input', function() {

var btc\_price = parseFloat($(".btc\_price").text().replace(',', ''));

var user\_btc\_input = $("#btc\_input").val();

console.log(btc\_price);

$('#gbp\_input').val(btc\_price \* user\_btc\_input);

});

$('#gbp\_input').on('input', function() {

var btc\_price = parseFloat($(".btc\_price").text().replace(',', ''));

var user\_gbp\_input = $("#gbp\_input").val();

console.log(btc\_price);

$('#btc\_input').val(user\_gbp\_input / btc\_price);

});

var myLatLngs = [{lat: 51.507389, lng: -0.127755}

, {lat: 51.607389, lng: -0.127755}

, {lat: 51.407389, lng: -0.127755}

, {lat: 51.407389, lng: -0.437755}

, {lat: 51.407389, lng: -0.317755}

, {lat: 51.507389, lng: -0.237755}

];

function initAutocomplete() {

var map = new google.maps.Map(document.getElementById('map'), {

center: {lat: 51.507389, lng: -0.127755},

zoom: 11,

mapTypeId: 'roadmap'

});

map.data.loadGeoJson('../Google/London.json');

map.data.setStyle({

fillColor: 'green',

strokeWeight: 1

});

// Create the search box and link it to the UI element.

var input = document.getElementById('pac-input');

var searchBox = new google.maps.places.SearchBox(input);

map.controls[google.maps.ControlPosition.TOP\_LEFT].push(input);

// Bias the SearchBox results towards current map's viewport.

map.addListener('bounds\_changed', function() {

searchBox.setBounds(map.getBounds());

});

var markers = [];

// Clear out the old markers.

markers.forEach(function(marker)

{

marker.setMap(null);

});

markers = [];

var icon = 'http://maps.google.com/mapfiles/ms/icons/green-dot.png';

for (var i = 0, len = myLatLngs.length; i < len; i++) {

markers.push(new google.maps.Marker({

map: map,

icon: icon,

position: myLatLngs[i]

}));

}

// Listen for the event fired when the user selects a prediction and retrieve

// more details for that place.

searchBox.addListener('places\_changed', function() {

var places = searchBox.getPlaces();

if (places.length == 0) {

return;

}

// For each place, get the icon, name and location.

var bounds = new google.maps.LatLngBounds();

places.forEach(function(place) {

if (!place.geometry) {

console.log("Returned place contains no geometry");

return;

}

icon = 'http://maps.google.com/mapfiles/ms/icons/red-dot.png';

// Create a marker for each place.

markers.push(new google.maps.Marker({

map: map,

icon: icon,

title: place.name,

position: place.geometry.location

}));

if (place.geometry.viewport) {

// Only geocodes have viewport.

bounds.union(place.geometry.viewport);

} else {

bounds.extend(place.geometry.location);

}

});

if (map.getZoom() > 0) {

console.log(map.getZoom());

}

map.fitBounds(bounds);

map.setZoom(12);

});

}

if !udata

article.login\_form

#login

form.well(id='form-login', method='post')

.bottom\_article\_header

h2 Hello!

h6 Please Login To Your Account

.form-group.article\_input

label.article\_input\_field Username

input.article\_input\_field.form-control(type='text', id='user-tf', name='user')

.form-group.article\_input

label.article\_input\_field Password

input.article\_input\_field.form-control(type='password', id='pass-tf', name='pass')

.controls.article\_buttons

button(type='submit').btn.btn-primary.button-sign-in

span.glyphicon.glyphicon-lock

| Sign In

.clearfix

hr

.btm-links.article\_footer

#forgot-password

a(href='#') Forgot Your Password?

#create-account.createAccountButton

a(href='/signup') Create An Account

.clearfix

include modals/alert

article.signup\_form.hide\_me

include account