Deploy nodejs on digital ocean (May 2019)

The public and private keys of RMS-web droplet didn’t worked, so I created a new droplet called RMS2 with publickey2 and privatekey2

1. Open Putty.exe and type your server/droplet ip address in Host name (RMS2)
2. Click on the connection -> Data tab and add “root” in the auto-login username field
3. Click on the SSH->Auth tab and browse for your privatekey2.ppk file
4. Go back to session tab, name it and click save
5. Click open to connect
6. $ adduser Hassan

$ password

$

Username: hassan

Password: has12345

Install nodejs

$ curl -sL https://deb.nodesource.com/setup\_6.x | sudo -E bash -

$ sudo apt-get install nodejs

Install git

$ sudo apt-get install git

Now we need to push the project from github to the server

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

**Github : hassanalbudairi**

**Git commands**

1- **git clone** (to copy a project from a remote branch to local branch)

2- **git push** (to update a remote branch from a local branch)

3- **git add –A** (to add all new files into the remote branch)

We need to commit the files after you add them, to do both add and commit once use

4- **git add -A && git commit -m “New files added”**

Then you need to push these changes to the remote branch

5- **git push**

6- to check the status, use git status

7- in order to update the server repository from the remote repository do this

I created the **master** branch on my laptop, git hub created an **origin/master branch** on github website. I clone this origin/master branch to digital ocean server and it called **origin/HEAD branch**

Any changes I made on the **master branch** I need first to add it, committed it and push it to the **origin/master branch** using the following commands:

$ **git add -A && commit -m “New files added”**

$ **git push**

Then I need to fetch these changes from **origin/HEAD branch** and merge the tworemote branches to get the changes to my digital ocean server using.

$ **git fetch origin master**

$ **git merge origin/HEAD origin/master**

$ git status

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

Open git push

$ cd ../../my\_files/QTS/RMS\_project/RMS\_web

Username: hasan

Pw: Has@12345

**How to access the db on the server**

1- in C:/ go to putty -> putty.exe -> choose rmsdrop and then load

2- $ pm2 stop all (to stop the server)

3- change directory to the location of the mongod (usr/bin) then use $ mongo

4- > show dbs

> use RMSdb

> show collections

4- > db. S14522.find() or db.S00181.find().pretty

5- > db. S14522.update({ \_id: ObjectId("5b24d16c951bd9748b21315e")},{ $set: { "charge" : 100 }})

6- > cls

db.S14529.remove({\_id: ObjectId("5ab4d2b992e0122f320d561e")})

6- to create a new feild a certian row by adding a new field use

>db.sensorsdb.update({No.:"1"},{No.:"1",lat:"" ...}

>db.sensorsdb.update({No.:"1"},{$set{name:""}}

7- to remove a feild use unset

>db.sensorsdb.update({No.1:"1"},{$unset:{name:1}});

8- If we want to update a record that isnt available in the collection, then use upset to add it to the record

>db.sensorsdb.update({No.:"5"},{$set:{name:1}},{upset:true});

9- to rename a field

>db.sensorsdb.update({No.1:"1"},{$rename:{"name":"title"}});

10- to remove a record

>db.sensorsdb.remove({No.1:"1"}); (this will delete every No.1

>db.sensorsdb.remove({No.1:"1"},{justOne:true}); (this will delete only the first record in the collection with No.1

To remove documents from a collection

db.sensorsdb.remove( { } )

**MMS solution**

Using intellisoftware mms reply based service

<https://www.intellisoftware.co.uk>

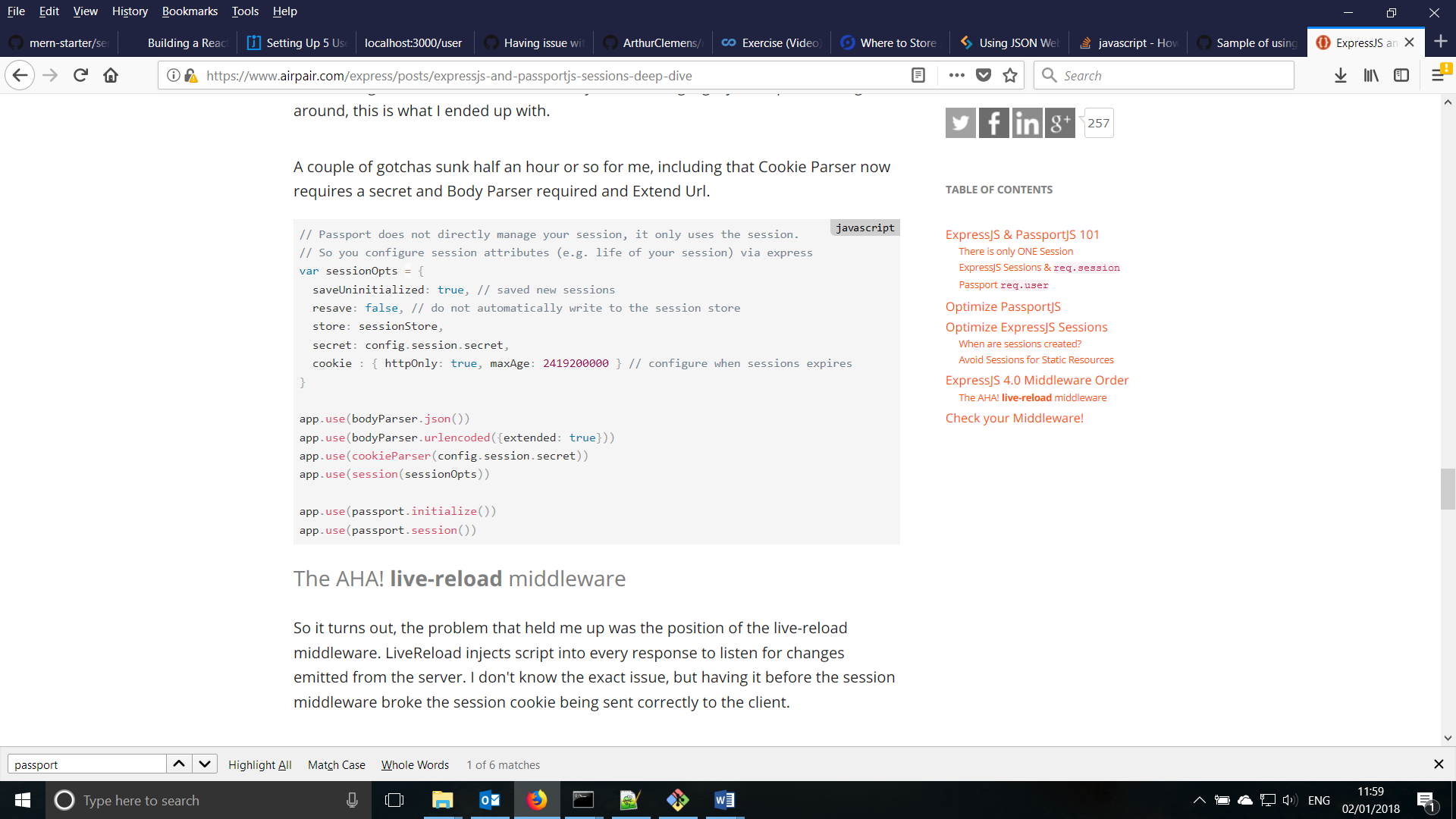
Un: qtsrms

Pw: H\*\*1\*\*\*\*

Adding photos to digital ocean server using this url

/assets/images/img/f250218\_ 00181.png

<https://www.airpair.com/express/posts/expressjs-and-passportjs-sessions-deep-dive>



when trying to view mongodb collection with special charcter, it is only a matter of shell to view them using []

db['W00181-2018-03-12T11-39-33'].drop()

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

**AngularJS v1 (on front end)**

1- Initiate your app on the page, usually use (angular initialise app)

<html ng-app=”app-name”>

Then down in the javascript code use

Var app = angular.module(‘app-name’,[list-of-dependences]);

2- Create controller

Under the var app in the javascript,

App.controller = (‘app-nameCtrl ‘, function ($scope){

$scope.hellos =[

Hello1,

Hello2,

Hello3

];

});

3- use the controller

<body ng-controller=” app-nameCtrl”>

<h3 ng-repeat=”hello in hellos”>

{{ hello}}

4- use ng-model =”name” and ng-repeat or $scope.name=’whateveryouwanttoseee’

5- For button use ng-click=

ng-app = userApp

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

**To build the website on digital ocean server, you need**

WinSCP : to view server files in windows format

Pm2 : to run the website as a process so if the server goes down, it does not affected

>pm2 start

>pm2 list

> pm2 stop all

>pm2 start bin/www

>pm2 delete app

Ubuntu 16.04.3 LTS

db version v2.6.10

//when destroy a droplet and create a new one you need to setup putty again from scratch (i.e. create a new username with password, create a private and public keys

$ adduser hassan

$ usermod –aG sudo

$ sudo su – Hassan

Username hasan

Has@12345

1- install node js

Install mongodb on digital ocean server

$ service mongod status

$ sudo service mongodb stop

Image size: resolution (horizontal \* vertical)\* bit depth (24 bit) /(8 (to convert from bit to byte) \* 1024 (from byte to KB)

Example 640x480\*24/(8\*1024) = 900KB (uncompressed)

**Mongodb section**

Mongodb directories on digital ocean

Databases var/lib/mongodb

Mongodb config etc

Mongod usr/bin

To remove a record from mongodb on digital ocean server, go to mongod and type

$ mongo

Db.photosdb.remove({“\_id”:6})

db.S14529.remove({\_id: ObjectId("5ab4d2b992e0122f320d561e")})

To locally run the website with dB

1- Open cmd, then navigate to C:\Mongo\bin and run the executable file of mongod or by using the command promote (as an administrator) inside the bin directory to run:

**> mongod --port 27017 --dbpath C:\mongodb\data\db**

2- To use the db use

> mongo

To switch to mongodb shell and then

> show dbs

>use RMSdb

>show collections

3- To open an http connection, open new cmd and cd to **C:\Users\hdab1q\ngrok** and then

>ngrok http 8080

4- To run the website locally, open file expoloer and direct to where the website files is (C:\my\_files\QTS\RMS\_project\RMS\_web) and then use shift + right click to open Power Shell Window here and then type

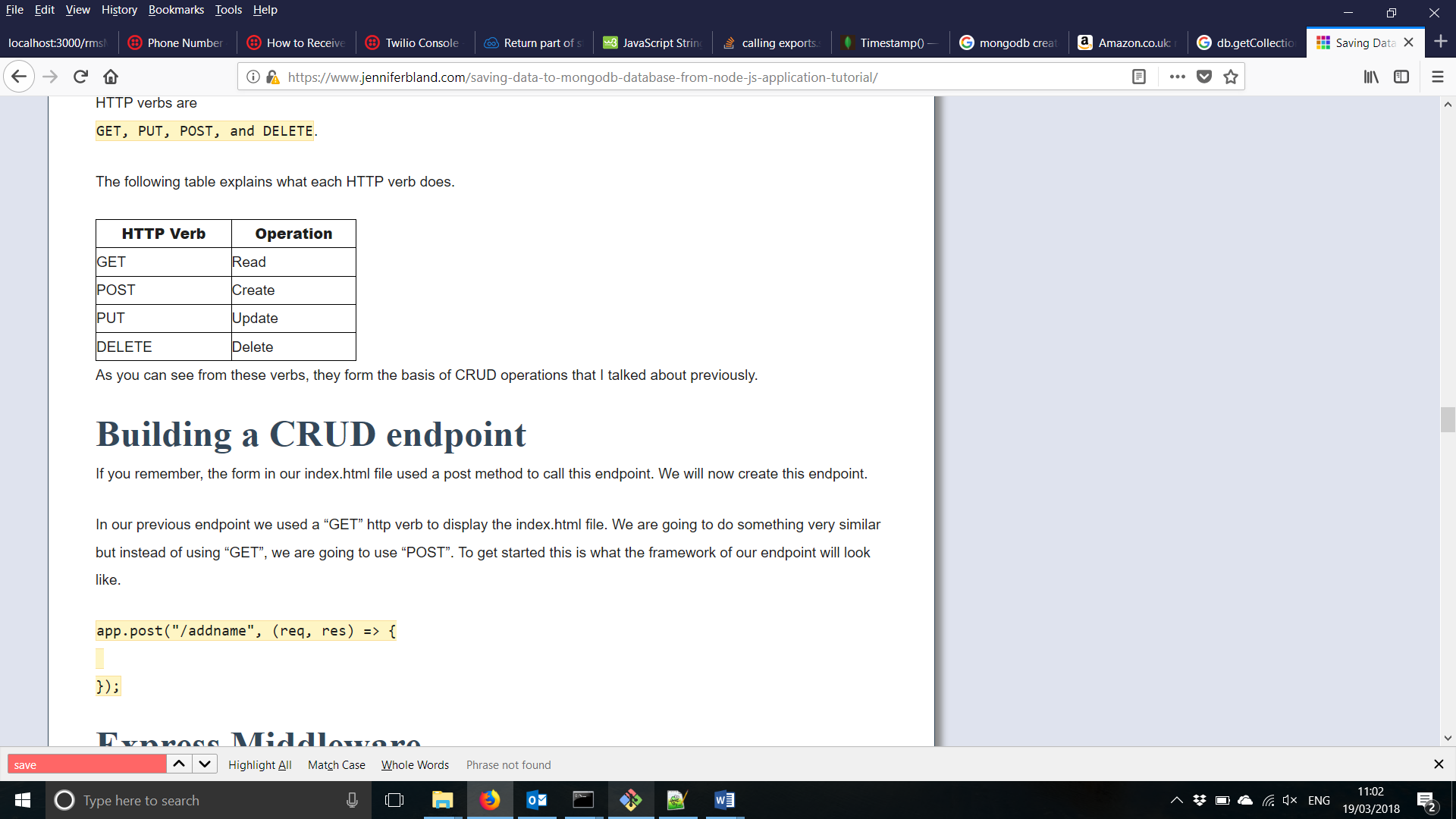
$ node server

5- open interent exploer and type in the search bar

http://localhost/80

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

////////////////////////



UK latitude regular expression (50 to 59)

**/^([5][0-9])\.?\d{0,4}?$/**

UK longitude regular expression (2 to -8)

**/^-[1-8]\.?\d{0,4}?$|^[0-2]\.?\d{0,4}?$/**

**UK phone number**

**/^\+[4][4][7]\d{9}$/**

Regular expression

For javascript should be between two forward slashes //

\d numerical

g at the end means globally

d+ all the number and not only the beginning of the number

^ the start of the string has to match the regex

$ ended the expression so the whole string has to match the whole regex

[a-z] match the first character of the string to this rule

\* means match zero or more characters

+ means match one or more characters

? means zero is optional

{} the length of the chracters in the string {1-5} means between 1 and 5

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

By default require() is not a valid function in client side javascript. I recommend you look into require.js as this does extend the client side to provide you with that function.

CommonJS, or RequireJS, SystemJS, or some other client-side module system unify the module system between the server and the client.

Browserify lets you require('modules') in the browser by bundling up all of your dependencies.

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

To run the website from cmd use

node bin\www

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

Mongo db:

Download Mongodb into C:\Mongodb\

Create two folder inside the directory (data and log)

in the cmd command window, nevigate to the Mongodb directory bin folder (C:\Mongodb\bin\)

Just for the first time

>mongod --directoryperdb --dbpath C:\mongodb\data\db --logpath C:\mongodb\log\mongo.log --logappend --rest --install

>start MongoDB

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

To start working with mongodb after setting up a database as above: we need to run the server (either by going to C:\Mongo\bin and run the executable file of mongod or by using the command promote (as an administrator) inside the bin directory to run:

**> mongod --port 27017 --dbpath C:\mongodb\data\db**

Then use mongodb shell to access to the db and the collections (either by run the excutable mongo or use the command promote (new one) inside bin to run

>mongo

>show dbs

>show collections

>db.sensordb.find()

To create a database/to switch to a db (such as CMSdb)

> use CMSdb

2- create a username

>db.createUser({

user:"user",

pwd:"1234",

roles: ["readWrite","dbAdmin"]

});

3-Create a collection (similar to table)

> db.createCollection('sensorsdb');

>show collections

4- insert data

db.sensorsdb.insert({No.:"", lat:"",log:"",sim\_no:"", location:"",added\_date:"",status:"",add\_info:""});

5- to view the data in the collection

>db.sensorsdb.find()

>db.sensorsdb.find().pretty

6- to create a new feild a certian row by adding a new field use

>db.sensorsdb.update({No.:"1"},{No.:"1",lat:"" ...}

>db.sensorsdb.update({No.:"1"},{$set{name:""}}

7- to remove a feild use unset

>db.sensorsdb.update({No.1:"1"},{$unset:{name:1}});

8- If we want to update a record that isnt available in the collection, then use upset to add it to the record

>db.sensorsdb.update({No.:"5"},{$set:{name:1}},{upset:true});

9- to rename a field

>db.sensorsdb.update({No.1:"1"},{$rename:{"name":"title"}});

10- to remove a record

>db.sensorsdb.remove({No.1:"1"}); (this will delete every No.1

>db.sensorsdb.remove({No.1:"1"},{justOne:true}); (this will delete only the first record in the collection with No.1

To remove documents from a collection

db.sensorsdb.remove( { } )

to delete a collection

db.sensordb.drop()

to delete a db

db.dropDatabase()

///////////////////

To read a file (such as Jason.package) inside a shell use

>cat Jason.package

Regex for pattern matching

List of validator expression is found at:

<https://github.com/chriso/validator.js>

Express-validator to validate the information inside input dom

Web development = Front end (client) and back end (server)

Client side HTML, CSS, JS (frameworks: JQuery, angularJS)

Server side php, perl, python, java, ruby

Static page: ready-made pages with fixed page content. Html, htm

Dynamic-generated on the fly with varying page content. Generated on the web server, interspread with JS, PHP, JSP, ASP (aspx)

For security reasons, Java script cannot :

1- unlimited reading/writing of files from client side

2- writing to the files on the server

3- close a window that was not opened by it

4- read from a web page served by an

other web server

DOM: document object model: stracture and style of a page, access/update content

DOM+JS = Dynamic html (client side)

Dome methods: getElementById(), ….

Html5 geolocation api, html drag and drop api, html web storage api, html web workers, html5 server-sent event (push application: send update from server to clients).

FTP/SSH tool (filezilla, putty) to access your server

Upload file from client side to server side using ftp

For server side, learn

1- language (php, ruby, node.js)

Javascript frameworks (react, angular)

Databases (MySQL, PostgreSQL)

1- Need experience in HTML, CSS, and java script (.js)

Chart.js is a community maintained open-source library. It supports 8 different chart types (including bars, lines, & pies), and they’re all responsive.

CDN (content delivery network)

HTML is not a programming language, per se. It’s a mark-up language for determining the layout of a webpage. Along with CSS, they are essential to web development.

Node.js is an open source server framework that allows you to run JavaScript on the server.

Secret key base

Twilio account sid

Twilio token

Twilio messaging service sid

Secret key base

1- Install git from <https://git-scm.com/> this gave me a command window (git bash) more interactive than the default command window for windows 10

2- Install twilio node.js from <https://github.com/twilio/twilio-node> use the following commands

in the project directroy

$ git clone <https://github.com/twilio/twilio-node.git>

$ npm install twilio

3- To setup an environment variables (i.e. your credientail will not be shown in the code), install dotenv from <https://github.com/motdotla/dotenv> . on the project directory use

$ git clone <https://github.com/motdotla/dotenv.git>

$ npm install dotenv

Create .env file (using Notepad ++ or any editor and save it in your project directory, add the credential information into this file and close it)

When you need to use these hidden credential into your code, use

require('dotenv').config()

var accountSid=process.env.Twilio\_account\_sid;

4- make sure you don’t push .env to the git hub by creating .gitignore file in the project directory and then inside this file just add .env

5- in order to install any node.js (such as express, express-session), that we need to use in our webpage code,

A- decide whether you need it for private (i.e. inside some code files that use var x = require (…), or public (i.e. need to use it directly from the git bash terminal (or cmd terminal).

For private, cd to the project directory and use

$ npm install express-sessions

This will install this jsnode from npm webpage inside node-modules folder in your project.

For public, use

$ npm install express-sessions –g

6- to save data from the website to the local machine we use cookies, (each cookie need a key and a value)

To create a cookie use

Regular expressions

[0-9]{5}

[0-9] a number between 0 and 9

{5} occurring 5 times.

‘^’ the beginning of values (not needed by html5).

‘$’ the ending of values (not needed by html5).

The <tr> tag defines a row in an HTML table.

The <th> tag defines a header cell in an HTML table. (contain header info)

The <td> tag defines a standard cell in an HTML table. (contain data)

function function1() {

//do your AJAX stuff here

}

setInterval(function1, 300000);

Java script is a function scoping and not block-scoping, means the variable of certain function cannt be excess by other function but variables inside blocks (like {}) can be excesses elsewhere inside the same function.

What I am using so far as a node modules

1- fs

2- node js

3- connect

4- D3

5-

Express app structure

Based on Model-View-Controller (MVC) design pattern

routers/ – defines your app routes and their logic

Public/ – contains all static files like images, styles and JavaScript

Views/ – provides templates which are rendered and served by your routes

package.json – remembers all packages that your app depends on and their versions

Models/ – represents data, implements business logic and handles storage

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

**Github**

**hassanalbudairi**

**Git commands**

1- **git clone** (to copy a project from a remote branch to local branch)

2- **git push** (to update a remote branch from a local branch)

3- **git add –A** (to add all new files into the remote branch)

We need to commit the files after you add them, to do both add and commit once use

4- **git add -A**

**git commit -m “New files added”**

Then you need to push these changes to the remote branch

5- **git push**

6- to check the status, use git status

7- in order to update the server repository from the remote repository do this

I created the **master** branch on my laptop, git hub created an **origin/master branch** on github website. I clone this origin/master branch to digital ocean server and it called **origin/HEAD branch**

Any changes I made on the **master branch** I need first to add it, committed it and push it to the **origin/master branch** using the following commands:

$ **git add -A && commit -m “New files added”**

$ **git push**

Then I need to fetch these changes from **origin/HEAD branch** and merge the tworemote branches to get the changes to my digital ocean server using.

$ **git fetch origin master**

$ **git merge origin/HEAD origin/master**