Module 1

1. Download git bash
2. Make sure node is installed

* Goto nodej.org

1. Create New folder MeanApp
2. Create a file server.js
3. Go to cmd navigate to MeanApp Folder
4. Npm install express
5. In server.js

* Copy Hello World from express site

Module 2 start

1. HTML setup

* Remove app.get
* Add app.use(express.static(\_\_dirname+"/public"));

1. Create Newfolder public
2. Create index.html in public folder

<!Doctype>

<html>

<head>

<title>MEAN Stack App</title>

</head>

<body>

<h1>MEAN App</h1>

</body>

</html>

Module 3 start

1. Setup AngularJs to our App

* Goto Angularjs.org
* Click download
* Copy cdn link
* Paste it below <h1> tag create script tag

1. Edit index.html

* <html ng-app="myApp">
* Below h1 tag type <input ng-model="test">{{test}}
* This is called Angular 2 way binding

Module 4 start

1. Setup Bootstrap to our App

* Goto getBootstrap.com --> Getting Started tab
* Copy the first 2 links javascript we will look into javascript later
* Paste it in the head section below <head> tag

1. Test Bootstrap is working

Module 5 start

1. Create General Layout for index.html

<div class="container">

<h1>MEAN App</h1>

<table class="table">

<thead>

<th>Country</th>

<th>State/Province</th>

<th>District</th>

<th>Area</th>

<th>Pincode</th>

</thead>

</table>

</div>

1. Add ng-controller to <div class="container" ng-controller="AppCntrl">
2. Create a new folder in public folder named controllers
3. Add a file named controller.js
4. Connect controller.js with our index.html and test whether it is working properly

* Add <script> tag with reference to controller.js in index.html below angularjs reference
* In controller.js add the following code

var app = angular.module("myApp",[]);

app.controller('AppCntrl', function ($scope, $http){

console.log("Hello Everybody I'm AppCntrl")

})

1. Test whether nothing is broken

Module 6 Start

1. Populate with Dummy Data

var area1 ={

country:"India",

state:"Karnataka",

district:"Bangalore",

area:"Marathhalli",

pincode:"560100"

};

var area2 ={

country:"India",

state:"Karnataka",

district:"Bangalore",

area:"KRPuram",

pincode:"560101"

};

var area3 ={

country:"India",

state:"Karnataka",

district:"Bangalore",

area:"Ecity",

pincode:"560420"

};

var areas =[area1,area2,area3];

$scope.arealist = arealist

$scope is the glue between controller and view

Add the following below thead in close tag in index.html

<tbody>

<tr ng-repeat="area in arealist">

<td>{{area.country}}</td>

<td>{{area.state}}</td>

<td>{{area.district}}</td>

<td>{{area.area}}</td>

<td>{{area.pincode}}</td>

</tr>

</tbody>

1. Test all the data is displayed on browser without any issues

Module 7 Start

1. Transition to getting dummy data from server (nodejs)
2. Use get request to get data from server
3. In the function app.controller('AppCntrl', function ($scope, $http) will help in talking to server

* below console.log in AppCntrl type $http.get('/arealist')
* In server.js below app.use type app.get('/arealist',function(req,res){

Console.log("I received a Get Request");

})

1. Test server getting get request from client.

Module 8 Start….

1. Copy dummy data from controller to server.js below console.log("I received a get request");

var area1 ={

country:"India",

state:"Karnataka",

district:"Bangalore",

area:"Marathhalli",

pincode:"560100"

};

var area2 ={

country:"India",

state:"Karnataka",

district:"Bangalore",

area:"KRPuram",

pincode:"560101"

};

var area3 ={

country:"India",

state:"Karnataka",

district:"Bangalore",

area:"Ecity",

pincode:"560420"

};

var arealist =[area1,area2,area3];

Add the following line

Res.json(arealist);

1. In the controller.js update the $http.get request as shown below

$http.get('/arealist').success(function(response){

console.log("I got the data I requested");

$scope.arealist = response

});

1. Test the same in browser. This step we made sure we get data from our server.
2. Download and Install Mongodb

Module 9 Start….

1. Npm install mongoose--save / mongojs
2. In server.js add var mongoose = require('mongoose');
3. In server.js add mongoose.connect('mongodb://localhost:27017/darealist');
4. Prepare index.html to post data to mongodb

* Add <th>Action</th> this in thead tag.
* Add below tbody tag

<tr>

<td><input class ="form-control" ng-model="area.country"></td>

<td><input class ="form-control" ng-model="area.state"></td>

<td><input class ="form-control" ng-model="area.district"></td>

<td><input class ="form-control" ng-model="area.area"></td>

<td><input class ="form-control" ng-model="area.pincode"></td>

<td><button class="btn btn-primary" ng-click="addArea()">Add Area</button></td>

</tr>

Module 10 Start….

1. Insert the Data to Mongodb.

* Create a new folder inside public folder named models
* Create a new file areaSchema.js inside models folder
* Type the following code inside areaSchema.js

var mongoose = require('mongoose');

var Schema = mongoose.Schema;

var areaSchema = new Schema({

country:String,

state:String,

district:String,

area:String,

pincode:String

});

var areaCollection = mongoose.model('areaCollection',areaSchema);

module.exports = areaCollection;

1. In server.js add the following require line code

* Var areaSchema= require('./public/models/areaSchema')
* Npm install body-parser --save

var bodyParser = require('body-parser')

var jsonParser = bodyParser.json()

* add the app.post as below

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

console.log(req.body);

var area = new areaSchema(req.body);

area.save(function (err, docs) {

if (err) {

console.log(err);

}

else {

res.json(docs);

}

})

})

1. Define and test Add Area Button

* Add the following code

$scope.addArea = function(){

console.log($scope.area);

$http.post('/arealist',$scope.area)

}

Test the same in browser using debug mode and check mongodb for verifying data is getting saved in mongo.

Module 11 Start

1. In controller.js add the following success code to check post is happening successfully

$http.post('/arealist',$scope.area).success(function(response){

console.log(response);

})

1. Lets work on getting the posted data on UI
2. Go to Server.js

* Remove dummy data
* Update app.get code as below

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

console.log("I received a Get request");

areaSchema.find({}, function (err, docs) {

if (err) { console.log("Error getting Area List") }

console.log("Result:" + docs);

res.json(docs);

})

})

1. Automate refresh after posting
2. Put the app.get method in controller.js inside refresh function

* Steps
* Var refresh=function(){
* App.get();
* }
* $scope.area ='' inside get
* Call refresh at the end in controller.js
* Call refresh inside post method

Module 12 Start…

1. Lets work on Delete
2. Add remove button in index.html

<td><button class="btn btn-danger" ng-click="remove(area.\_id)">Remove</button></td>

1. Define and test Remove button

$scope.remove = function(id){

console.log(id);

}

1. Test in the browser to see id is getting returned without any issues..
2. In the controller.js inside remove function type the following code.

$http.delete('/arealist/'+id).success(function (response) {

refresh();

});

1. In server.js define the function for delete

app.delete('/arealist/:id', function (req, res) {

var id = req.params.id;

console.log(id);

})

1. Lets now delete the content from the mongoDB Database.

areaSchema.remove({ \_id: req.params.id }, function (err, docs) {

if (err) { console.log("Error removing Area") }

console.log(docs);

res.json(docs);

});

Test the same…..

Module 13 start….

1. Lets work on Update
2. Add update & edit button in index.html

<td><button class="btn btn-warning" ng-click="edit(area.\_id)">Edit</button></td>

<td><button class="btn btn-info" ng-click="update()">Update</button></td>

1. Define Edit function

$scope.edit = function(id){

console.log(id);

}

1. Test in the browser to see id is getting returned without any issues..
2. In the controller.js inside edit function type the following code.

$http.get('/arealist/'+id).success(function (response) {

$scope.area=response

});

1. In server.js define the function for get

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

var id = req.params.id;

console.log(id);

})

1. Now lets bring one document corresponding to the id back to the UI

areaSchema.findOne({ \_id: req.params.id }).exec(function (err, docs) {

if (err) { console.log("Error getting particular area") }

console.log("Result:" + docs);

res.json(docs);

})

1. Test the same to check it is working….

Module 14 start…

1. Define Update function in controller.js

$scope.update = function(){

console.log($scope.area.\_id);

}

1. Next insert the http put command below console.log in update function

$http.put('/arealist/' + $scope.area.\_id, $scope.area).success(function (response) {

refresh();

});

1. In server.js define the function for updating the document

app.put('/arealist/:id', jsonParser, function (req, res) {

var id = req.params.id;

console.log(req.body);

})

1. Now lets update the document in mongodb by using the below code

areaSchema.findById(req.params.id, function (err, doc) {

if (err) {

console.log("Error getting document");

return;

}

console.log("found for update" + doc);

doc.country = req.body.country;

doc.state = req.body.state;

doc.district = req.body.district;

doc.area = req.body.area;

doc.pincode = req.body.pincode;

doc.save(function (err, docs) {

if (err) {

console.log(err);

}

else {

res.json(docs);

}

})

})

1. End of Crud Operation….