Mean.JS Generator uses Yeoman to create a basic application

<https://www.getpostman.com/> – POSTMAN FOR TESTING ROUTES INSTEAD OF BROWSER

M – M Storage

E – Server Routing and biz Logic

A – Client Side Application

N – Ran by Node

All Applications Data is stored in MongoDB Documents

The Client (Angular) requests data from Express

Express Reads Request

Express Asks MongoDB for the data

Mongo sends data

Express then sends data to the Client

The Client then displays it for User

–

To Do Application

app.js (express)

require express

create instance of express server – var app = express();

will allow for any middleware, handle routes, and start server.

Configure var app to listen for requests on port 3000

app.listen(3000, function () {

console.log(“the server is running on port 3000!”);

});

Test by running node src/app.js

BEFORE APP.LISTEN

We add static Middleware

app.use(express.static(‘public’));

Starts from root of folder.

We can be more explicit by adding app.use(‘/’, express.static(‘public’));

Express GET Routes

GET Request

sends page to be viewed

in app.js

Middleware before listen

app.get(‘/todos’, function(req,res){

res.send(‘These are the todos!’);

});

^// This creates the route localhost:3000/todos

To make sure you do not conflict with root public Routes

app.get(‘/api/todos’, …

to return json

res.json({todos[]})

To save yourself from typing /api on every route..

var router = express.router();

router.get(‘/todos’, function(res, req){

res.json …

});

app.use(‘/api’, router);

Will create more routes to Post Put(update existing) Delete

//

Data.JS

Changed mock data from mock json to /api/todos

WEBPACK – Used to bundle .js files into one main file…

Helps to eliminate number of files and order dependency in index.HTML

Setup MongoDB and Mongoose

Add Require Mongoose to app.js and database file.