4)Installing the lessons code – Learn why it is essential to use NPM 5

Here we cloned the git hub repository into our system. Here we have package-lock.json file.so make sure you use npm version 5 or above. We can update npm like this-

npm install –g npm.

Now before installing dependencies it is important to not to use master branch of repository. So we will switch to our first branch.

Runt his-

git branch –r

this will give list of all remote branches. We will us this branch – **origin/1-signup**. Bramches are numbered in order they should be used.

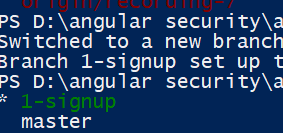
Now run this-

**git checkout -b 1-signup origin/1-signup**

here 1-signup is name of local branch that we are about to create. This can be anything. Now second argument(**origin/1-signup**)t defines which remote branch we want to bring here locally. Now if you run thins command and then run this-

**git branch**

you will see that you are in local 1-signup branch(or new branch that you have created if you gave different name to your local branch).so this will be our starting point for development.



Now run **npm install** to install our dependencies, according to package-lock.josn file. it is important that you do not edit this file manually. This file is equal to yarn.lock file, it provides same fucntinality. Lets open it . it is json file which mentions which exact versions should be downloaded on your machine. This will ensure that we are using same versions as instructor is using. So we will not face semantic versioning issues.

5)How to run node in typescript with hot reloading

Here we will see how to run node processs using typrscript directly, without need of compiling ts into js first. We will see how to run server in hot reloading mode. We will also see how to run small http server in node. All this is n context of starting our smaple application. We are going to need 2 processes for it-one is angular cli development cli server, other is small node server that we will be running in https. Lets first run node server.

Open package.json. here we will see couple of tasks that we will be using.

In script section we can see-

"start-server": "./node\_modules/.bin/ts-node ./server/server.ts --secure",

Here we are running ts-node command-

./node\_modules/.bin/ts-node

and we are using it run our server-

./server/server.ts

We will review our server in a second. ts-node is a command line utility which allow us to run node programs in type script without having to compile ts into js first. So compulation will happen on fly.

--secure means that server is running in http mode.

But we will not use start-server task . we will use server task. Which is-

"server": "./node\_modules/.bin/nodemon -w ./server --ext \".ts\" --exec \"npm run start-server\"",

It uses node utility called nodemon for scanning all the typescript files in server directory, each time we made change to our code, it will run our server again. (it will run start-server tasks).this means when we are adding functionality to our server this will hot reload the server. Now lets run our server. Run this command-

**npm run server**

now our server will run and you can see on which port it is running.

So this is our rest api server. We also need to run our angular cli server. For this we will use start task which is-

"start": "ng serve --proxy-config ./proxy.json

To start it run this command-

**npm start**

Here we are running angular-cli server in proxy mode , we are passing some configuration. proxy configuration is in file proxy.json. lets see this file-

{

"/api": {

"target": "https://localhost:9000",

"secure": false

}

}

What this confirmation file means is that every http request that we do in our application, that starts with /api is going to be proxied or redirected to our rest server which is <https://localhost:9000>. This server is running with self signed certificate which is not recognized by angular cli so we set secure flag to false, which means that angular cli will not validate with self signed certificate.

Now open localhost:4200, this is application that we will secure.

6)Guided tour of sample application

In our application we have some lessons name and description. Right now anyone having this url can see them,. So this data is unprotected.

Then we see how data is coming to our application. Then we see we are making http request to api/lessons, because of proxy .josn.

Lets have a look at our backend server.

"start-server": "./node\_modules/.bin/ts-node ./server/server.ts --secure"

Here we can see that entry point of our server is server.ts file. lets open it. It is simple express app. There we create http or https server depending upon whether secure flag is present or not.default value of thi falf is true. If you want to set http server just do not send tgis flag. In this course we will use https server. Then we set certificates for http server. If you want http server simply remove secure from start-server task in package.json.

We have only one route, lets jump into it and see from where data is coming.Then we use inmemory database to return some value. This data will be reset each time application restart.