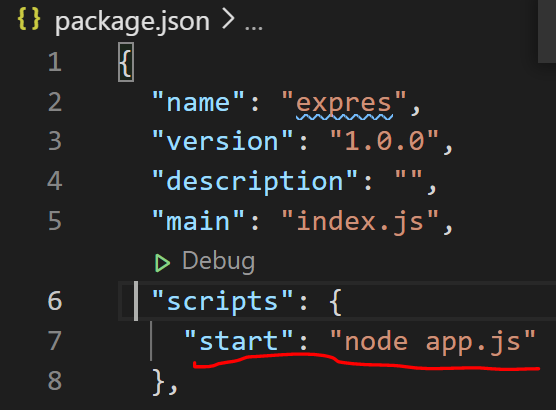
IMP/ => **How to setup express with node**

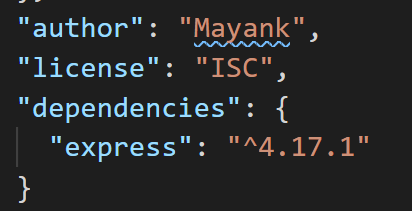
So first we have to write npm init in terminal using git terminal or vs code terminal terminal should be opened in the project

After creating it a json file is created with the name (package.JSON) maybe so to use the npm scripting commands you have to write that command and what it will do inside the script of package.json like npm start you have to give the name of that file to start



After that to install express you have to write npm install --save express

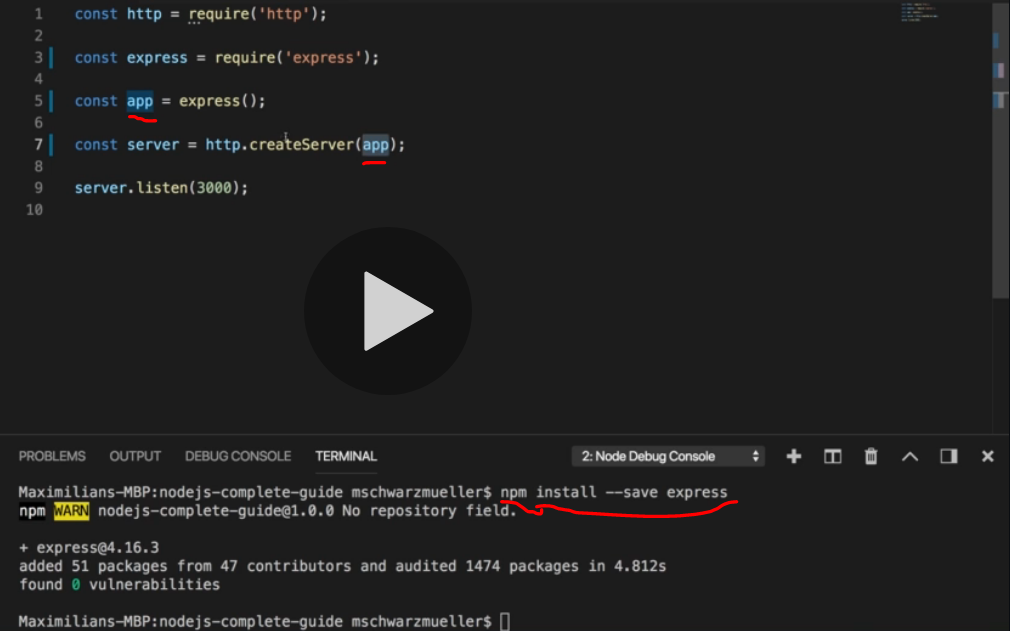
 we are doing it because we are not installing it in separate file instead we are adding dependencies in our package.json file



Useful link of installing express = <https://expressjs.com/en/starter/installing.html>

* EXPRESS JS

So we have to use express js with node js to buid the server side and server side api’s

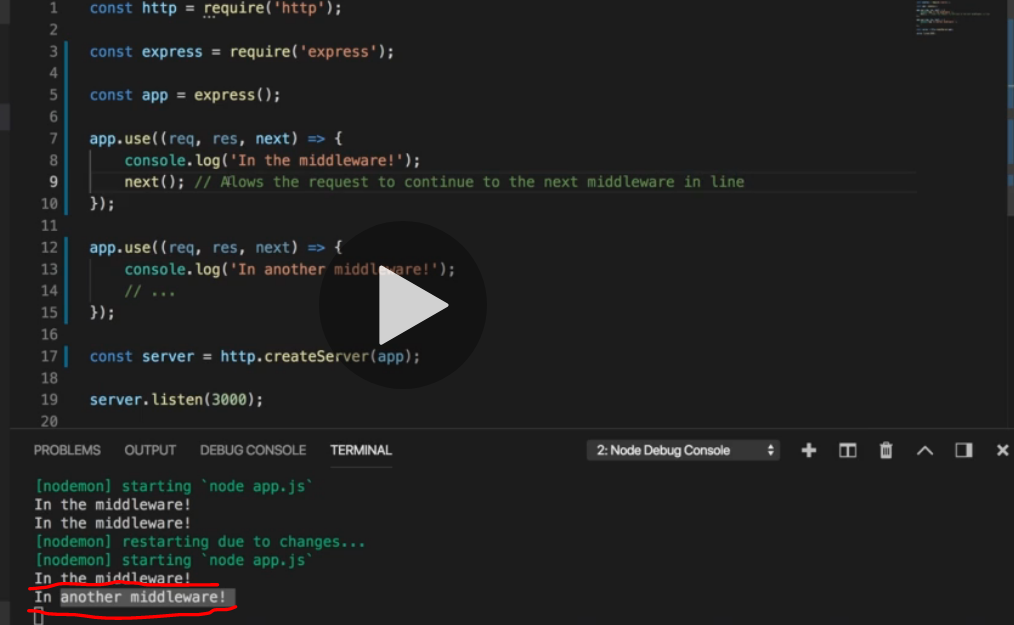


Here we are saving our express in our old file so we can use express from it by require so after saving in const we are saving it to an app as a function so that we can call it in server

So npm install –save will add the dependencies in our package.json file.

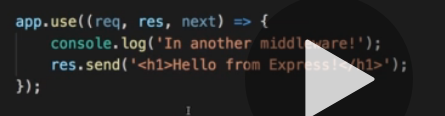
So the use of middleware In express

We can add multiple function and methods in middleware so that we do not have to write whole code at one function instead we can write multiple function in it



So that we can use app as function which containing request, response as we see earlier but the net is used for calling the next function of app if we do not write next it will not run the another function

We can also send the data to server which is by default in html

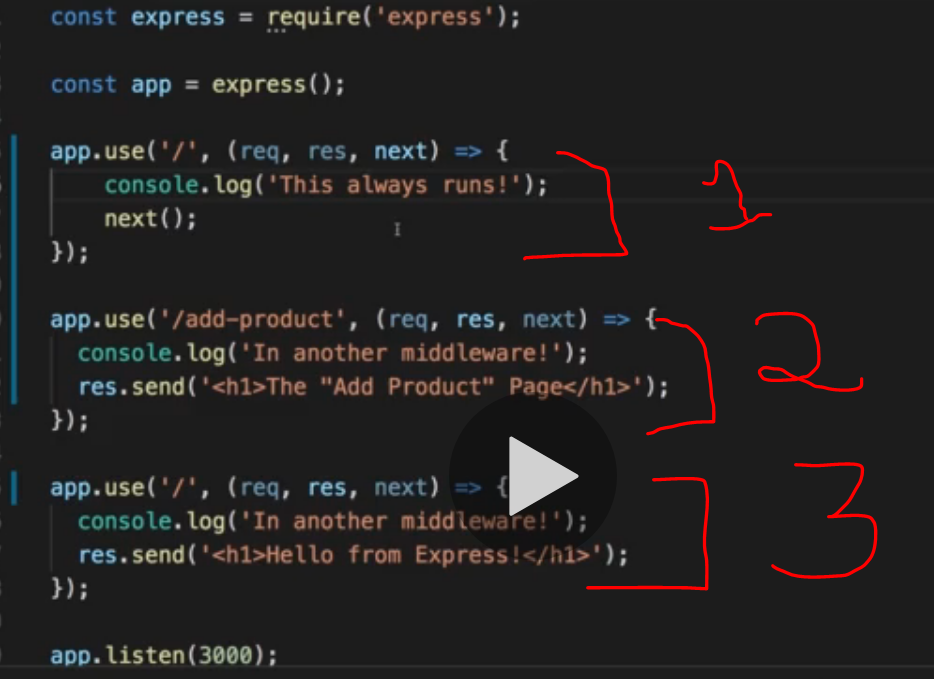


Res. Send is saving a lots of code because previously we are using a buffer and hole lots of code which we do not have to write because it is already written in the express js you can see it on their git repository of express code.

So express js will sort the repetitive coding part which is used more often so we can do our real operations

So we can use app.use in different ways if we do not write anything in that means it is applicable for all and we have to do next() to perform different operation which is below it but if we use app.use() with some path that will run in any case for that path only so we do not have to use next but if we write (./) only it means it applicable to all and it is default too.

 we can have more then 1 callBacks



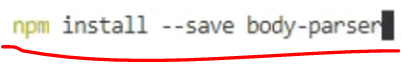
So the first one is applicable to all and we have to write next() to run the next on the list

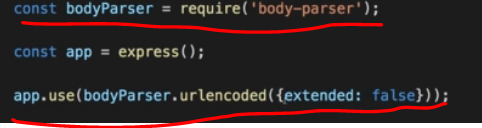
The second one will run for the /add-productpage so we are not writing next because we do not want it to go down we only show that response only

And the third one is for all, so if we write add-product/something it will still open the product page

* Form Handling in express js

Form Handling is done by body parser in js from which we can parse JSON type of data, url encoded type of data and many more ( url encoded data is smoothing that is encode in form like your name)

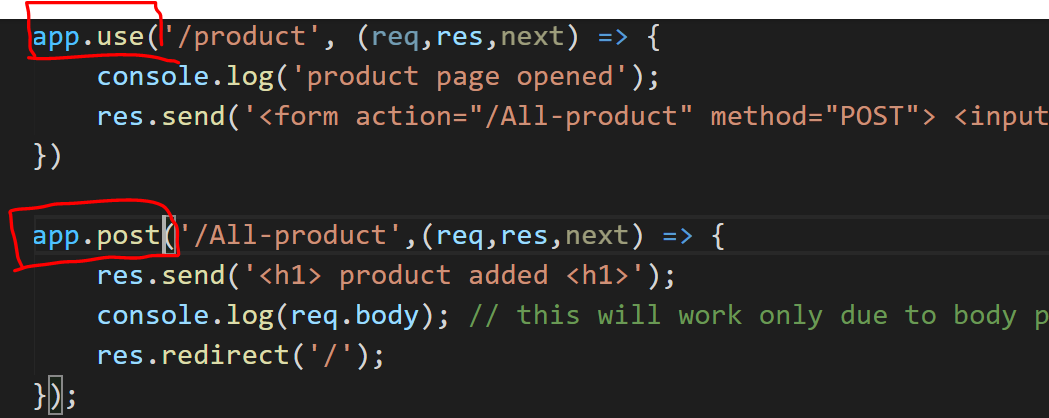
So to enable body parser you have to first install body parser by it will save the parse file in our package dependencies



Just like we are using body parser by require and using it in the app where we are using url encoded method with seting default values as false.

Which can run  so from then you can see your object

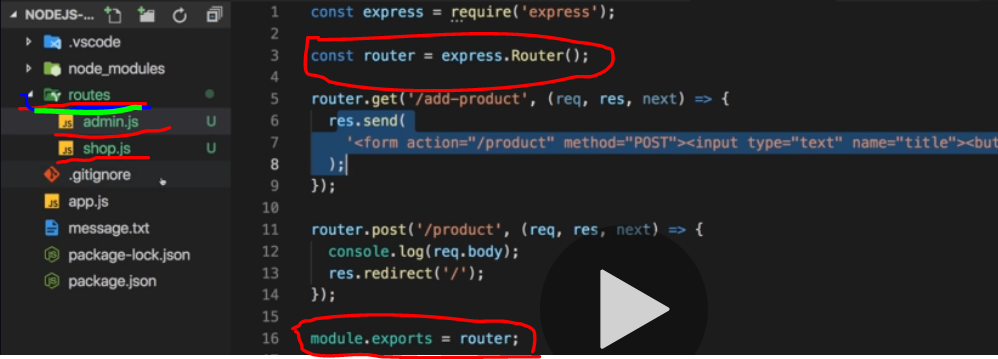
We can also use get,post,delete etc in place of app.use



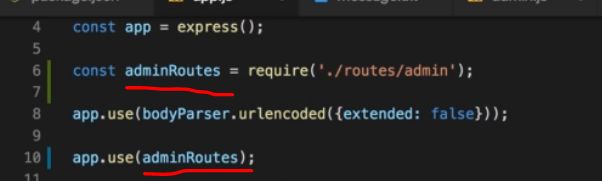
So the advantages of writing get,post ect instead of app.use is that they will run no matter where is the position of them and they will run

* OUT Source the routs

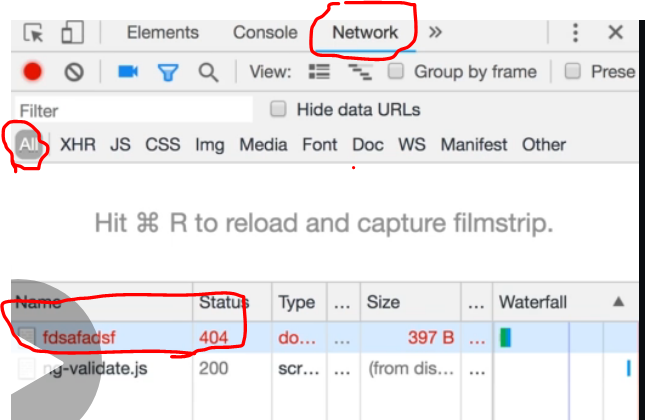
We can also out source the routs which means nothing but creating all the routs which we are creating as app.use,get or post in the different file that we can name it as routs and exporting them so that we can use it in our app.js files



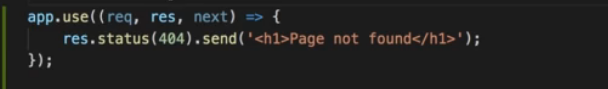
Here we are exporting routs in our admin,shop.js so we can export it into our app.js file



So their we are exporting admin.js by require in adminRoutes so that we can call it inside our app.use() which work same as it working so that we can listen our server by it.

 http://localhost:3500/fdgdfgdfg

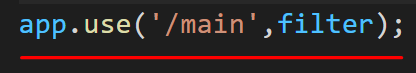
We can also use it to display message for some status like error and we are doing it by



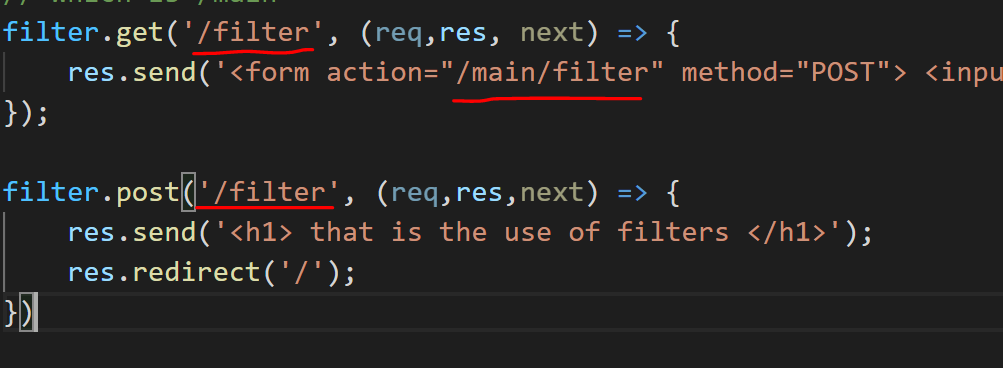
So we ca use these code at last to send h1 tag for the particular status obviously we have to use next or routs file before it so it can go down and check for every call best practice to calling by routs

* FILTER IN NODE JS

Filter is used to set up the common starting points so that instead of writing the /page address agin and agin we can write it a once.

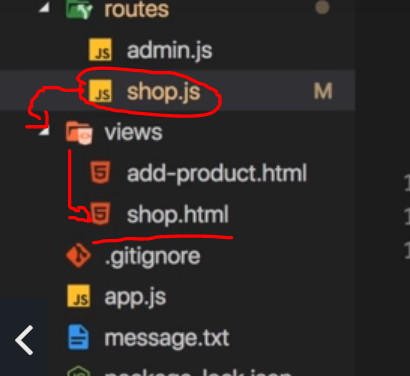


So that we can write export our filter without adding /main all the time we can write things ahed of it like /main/filter = /filter but we have to provide action as full address so that it can open that file /main/filter



* HOW TO SEND HTML FILES TO USER

So instead of writing html in res.send above we can write a html page separately

* So we can write it by using res.sendFile() instead of using res.send() and give the file path.
* But if we try to put a path that will give error because sendFile takes the path in form of linux system which is( /fff ) but window have a path with ( \fff )
* So o over came this error we have to require(‘path’) 
* And then used is as  path.join will join the path so that any user can use it.
* \_\_dirname in it is holding the absolute path of the folder so we can pass path from it
* But we have to give (.//) first because we are calling sendfile from some inside file so we are doing (.//) so that we can go to the root first and then give the path
* So we do not have to write (/) or (\) before any path
* The file from we are calling is shop.js
* 

If we try to add css file into our html page that file we are sending it will not send with it because we are sending a html file only but not the css in above code so for that we have to pass the static location of the folder so the html can use the css file in it

