**CS 628 Full-Stack Development II**

**HOS03B Modifying Express for MVC**

July 16, 2020

12/29/2020 Updated by Min

School of Technology & Computing (STC)

City University of Seattle (CityU)

**Before You Start**

* The directory path shown in screenshots may be different from yours.
* Some steps are not explained in the tutorial**.** If you are not sure what to do:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Resources**

* Holmes, S. & Harber, C. (2019). [Getting MEAN with Mongo, Express, Angular, and Node, Second Edition](https://login.proxy.cityu.edu/sso/skillport?context=148121). Manning Publications. (ISBN 9781617294754)
  + Chapter 3: Creating and Setting up a MEAN Project
    - 3.3 Modifying Express for MVC

**Learning Outcomes**

* Understand an architectural design pattern – MVC.
* Apply MVC to your application.

**Activities**

* Preparing for a MEAN project environment
* **Creating the MVC folder structure**
* **Using the New Views and Routers Folder Location**
* **Not using a depreciated feature in ES2015 – ‘var’**
* **Splitting Controllers from Routes**
* Source Code Management and Deployment

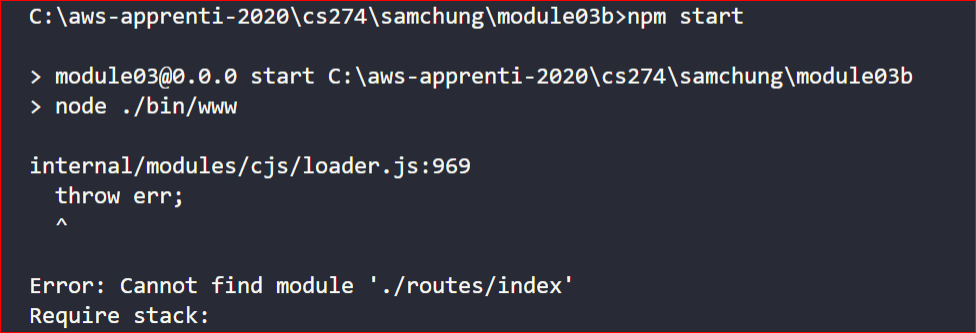
**Preparing for a MEAN project environment**

1. Copy the “module03a” directory to the “module03b” one.
2. Open the “module03b” folder in the VSCode.  
   Close the “Welcome” window.  
   Open the terminal (Ctrl+`).
3. Test whether the application works correctly.  
   >>>npm start  
     
   http://localhost:3000

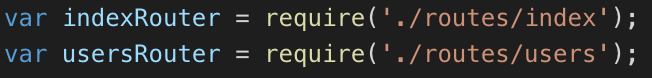
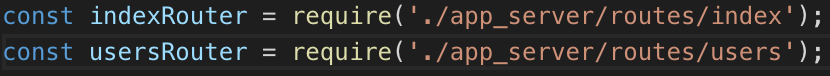
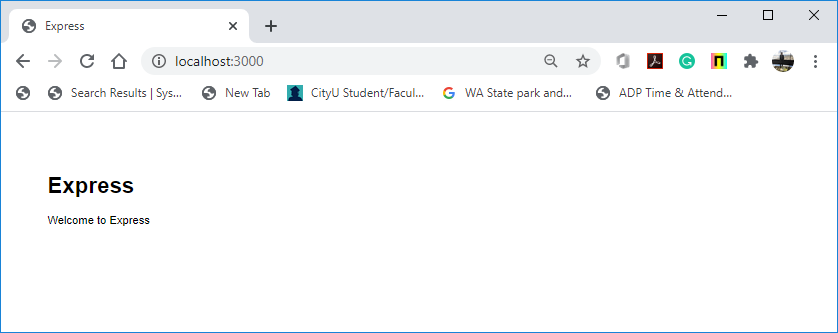
**Creating the MVC folder structure**You will have an obvious MVC setup in the application, which makes it easier to separate your concerns.

1. Under the “module03b” folder, create a new folder “app\_server” for all your MVC architecture.
2. Under “app\_server” create two new folders called “models” and “controllers”.
3. Move the “views” and “routes” folders from the root of the application into the “app\_server” folder.

|  |  |
| --- | --- |
| Before | After |
|  |  |

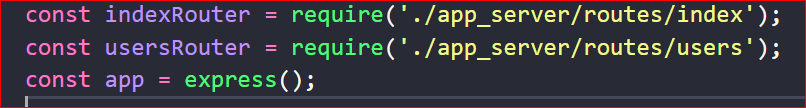
1. But, if you try to run the application now, it won’t work, as you’ve just broken views and routes.   
   >>>npm start  
   

**Using the New Views and Routers Folder Location**

1. As we have moved the views and routes into different location, we need to change the path in “app.js”. Under “app.js,” find the following line.  
     
     
   Change the above line to this.  
   
2. For the routes, find the following lines under “app.js”  
     
     
   Change the above line to this.
3. Run the application for testing.  
   >>>npm start  
   

**Not using a depreciated feature in ES2015 – ‘var’**

1. The “app.js” uses the “var” keyword to define variables.   
   Replace the context-level variable initializer ‘var’ with the block-level variable initializer ‘const.’  
     
   A black sign with white text

   Description automatically generated  
     
   
2. Run the application for testing.  
   >>>npm start

**Splitting Controllers from Routes**Controllers should manage the application logic, and routing should map URL requests to controllers.

1. Under the ‘app\_server/routes’ directory, open the ‘index.js’ file and find that controllers are part of the routes.

var express = require('express');

var router = express.Router();

/\* GET home page. \*/

router.get('/', function(*req*, *res*, *next*) {

*res*.render('index', { title: 'Express' });

});

*module*.*exports* = router;

1. Rewrite the ‘router.get’ function by moving the Controller out of the Routes File.

// var express = require('express');

// var router = express.Router();

const express = require('express');

const router = express.Router();

/\* GET home page. \*/

// router.get('/', function(req, res, next) {

// res.render('index', { title: 'Express' });

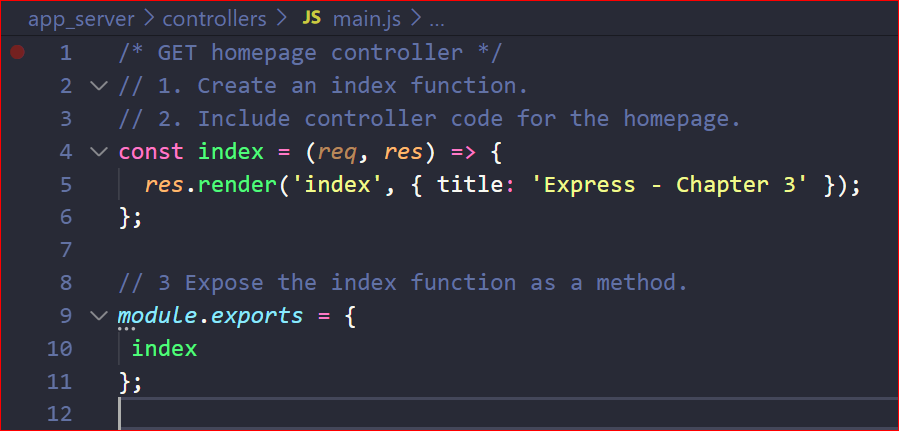
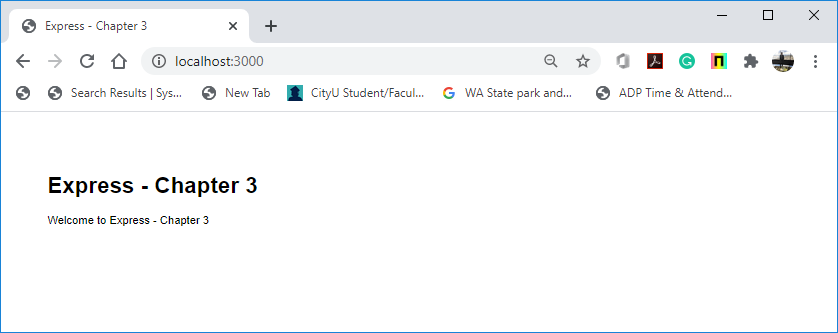
// });

// 1. Require the main controllers file.

const ctrlMain = require('../controllers/main');

// 2. Reference the index method of the controllers in the route definition.

router.get('/', ctrlMain.index);

1. Under the “app\_server/controllers” directory, create a file “main.js” and add the below code to that file. You require this controller module in the routes file so that you can use the exposed method in the route definition.   
   
2. Run the application for testing.  
   >>>npm start  
   

**Source Code Management and Deployment**

**Push your work to GitHub**

Run the following commands to push your work to the GitHub repository:

Open the terminal from the VSCode by hitting the control + ~ key and type the following command:

>>> git add .

>>> git commit -m “Submission for Module 3 --yourname”

>>> git push origin master

**Push your work to Heroku**

>>> git push heroku master

If you cannot remember your branch name, run the command “git status” to check.