

/express-locallibrary-tutorial //the project root

/routes

index.js

users.js

catalog.js

Open **/routes/catalog.js** and copy in the code below:

var express = require('express');

var router = express.Router();

// Require controller modules.

var book\_controller = require('../controllers/bookController');

var author\_controller = require('../controllers/authorController');

var genre\_controller = require('../controllers/genreController');

var book\_instance\_controller = require('../controllers/bookinstanceController');

/// BOOK ROUTES ///

// GET catalog home page.

router.get('/', book\_controller.index);

// GET request for creating a Book. NOTE This must come before routes that display Book (uses id).

router.get('/book/create', book\_controller.book\_create\_get);

// POST request for creating Book.

router.post('/book/create', book\_controller.book\_create\_post);

// GET request to delete Book.

router.get('/book/:id/delete', book\_controller.book\_delete\_get);

// POST request to delete Book.

router.post('/book/:id/delete', book\_controller.book\_delete\_post);

// GET request to update Book.

router.get('/book/:id/update', book\_controller.book\_update\_get);

// POST request to update Book.

router.post('/book/:id/update', book\_controller.book\_update\_post);

// GET request for one Book.

router.get('/book/:id', book\_controller.book\_detail);

// GET request for list of all Book items.

router.get('/books', book\_controller.book\_list);

/// AUTHOR ROUTES ///

// GET request for creating Author. NOTE This must come before route for id (i.e. display author).

router.get('/author/create', author\_controller.author\_create\_get);

// POST request for creating Author.

router.post('/author/create', author\_controller.author\_create\_post);

// GET request to delete Author.

router.get('/author/:id/delete', author\_controller.author\_delete\_get);

// POST request to delete Author.

router.post('/author/:id/delete', author\_controller.author\_delete\_post);

// GET request to update Author.

router.get('/author/:id/update', author\_controller.author\_update\_get);

// POST request to update Author.

router.post('/author/:id/update', author\_controller.author\_update\_post);

// GET request for one Author.

router.get('/author/:id', author\_controller.author\_detail);

// GET request for list of all Authors.

router.get('/authors', author\_controller.author\_list);

/// GENRE ROUTES ///

// GET request for creating a Genre. NOTE This must come before route that displays Genre (uses id).

router.get('/genre/create', genre\_controller.genre\_create\_get);

//POST request for creating Genre.

router.post('/genre/create', genre\_controller.genre\_create\_post);

// GET request to delete Genre.

router.get('/genre/:id/delete', genre\_controller.genre\_delete\_get);

// POST request to delete Genre.

router.post('/genre/:id/delete', genre\_controller.genre\_delete\_post);

// GET request to update Genre.

router.get('/genre/:id/update', genre\_controller.genre\_update\_get);

// POST request to update Genre.

router.post('/genre/:id/update', genre\_controller.genre\_update\_post);

// GET request for one Genre.

router.get('/genre/:id', genre\_controller.genre\_detail);

// GET request for list of all Genre.

router.get('/genres', genre\_controller.genre\_list);

/// BOOKINSTANCE ROUTES ///

// GET request for creating a BookInstance. NOTE This must come before route that displays BookInstance (uses id).

router.get('/bookinstance/create', book\_instance\_controller.bookinstance\_create\_get);

// POST request for creating BookInstance.

router.post('/bookinstance/create', book\_instance\_controller.bookinstance\_create\_post);

// GET request to delete BookInstance.

router.get('/bookinstance/:id/delete', book\_instance\_controller.bookinstance\_delete\_get);

// POST request to delete BookInstance.

router.post('/bookinstance/:id/delete', book\_instance\_controller.bookinstance\_delete\_post);

// GET request to update BookInstance.

router.get('/bookinstance/:id/update', book\_instance\_controller.bookinstance\_update\_get);

// POST request to update BookInstance.

router.post('/bookinstance/:id/update', book\_instance\_controller.bookinstance\_update\_post);

// GET request for one BookInstance.

router.get('/bookinstance/:id', book\_instance\_controller.bookinstance\_detail);

// GET request for list of all BookInstance.

router.get('/bookinstances', book\_instance\_controller.bookinstance\_list);

module.exports = router;