**Lesson 9 Practice Hands-On**

**Directions**

Now that you learned how to talk to a Front-End application from an Express application, its time to put that knowledge to the test. In this Hands-On exercise, you will continue working with the two projects from the lesson. The Express project is named ExpressL09 and the React or Angular apps are named accordingly. Please open these two applications within Visual Studio code. Currently, you are able to use the static data from the Express side and render them onto the page from your Front-End app. Now, you are going to render the same data, but from a database. Follow the below setup instruction and be sure to complete the requirements. This Hands-On will not be graded, but we encourage you to complete it. The best way to become a great programmer is to practice!

**Setup**

1. Open up your terminal/command prompt.
2. Navigate to your desktop in your terminal:
3. cd Desktop
4. Then, navigate to the Express-Course directory in your terminal:
5. cd Express-Course
6. Finally, navigate to the express\_lesson\_nine directory:
7. cd express\_lesson\_nine
8. Open your express\_lesson\_nine project within VS Code (or you can open the folder within VS Code directly):
9. code .
10. Run:
11. sequelize init
12. Add the following to the app.js to import the models and to ensure your database and models are the same:
13. var createError = require('http-errors');
14. var express = require('express');
15. var path = require('path');
16. var cookieParser = require('cookie-parser');
17. var logger = require('morgan');
18. var models = require('./models');
19. var indexRouter = require('./routes/index');
20. var usersRouter = require('./routes/users');
21. // ...
22. models.sequelize.sync().then(function () {
23. console.log("DB Sync'd up")
24. });
25. module.exports = app;
26. Add the following route to the routes/index.js file:
27. router.get('/starTrekPlanets', function (req, res, next) {
29. });

**Requirements**

Meet the following requirements within the Express application. You will need to add a small amount of code to your React or Angular application as well, which is located below.

**Step 1**

Write the code for the /starTrekPlanets route so your Front-End application displays information about the following planets in the Star Trek universe:

* Id: 1, Name: Q'onoS, Moons: 1
* Id: 2, Name: Vulcan, Moons: 0
* Id: 3, Name: Earth, Moons: 1
* Id: 4, Name: Rator III, Moons: 0

**Note**: Remember that React and Angular will be case sensitive. The JSON you send needs to match what the front end expects, similar to the format of the /staticPlanets route.

**React Code Needed**

Within your React project, you need to add another component to the App.js file so it will render planets route along with the already existing staticPlanets route. Add the following:

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

import Planets from './components/Planets';

class App extends Component {

render() {

return (

<div className="App">

<header className="App-header">

<img src={logo} className="App-logo" alt="logo" />

<h1 className="App-title">Welcome to React</h1>

</header>

<Planets uri="http://localhost:3001/staticPlanets"/>

//Add the following code

<hr />

<Planets uri="http://localhost:3001/starTrekPlanets"/>

</div>

);

}

}

export default App;

* The <hr /> tag will create a horizontal line.
* The new <Planets /> component will now hit the Express route of /starTrekPlanets

Run nodemon for the Express application and npm start for the React application.

**Angular Code Needed**

Add the following code into the noted files:

**app.component.ts** file:

import { Component } from '@angular/core';

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent {

title = 'Angular 2 Planets';

staticPath: string = 'http://localhost:3001/staticPlanets';

starTrekPath: string = 'http://localhost:3001/starTrekPlanets';

}

**app.component.html** file:

<div style="text-align:center">

<h1>

Welcome to {{ title }}!

</h1>

</div>

<hr/>

<div style="text-align:center">

<div>

<h2>Static Planet Data</h2>

<h3>{{staticPath}}</h3>

<app-planets

[dataPath]="staticPath">

</app-planets>

</div>

<hr/>

<div>

<h2>Live Planet Data</h2>

<h3>{{starTrekPath}}</h3>

<app-planets

[dataPath]="starTrekPath">

</app-planets>

</div>

</div>

**Tip!**

Run nodemon for the **Express** application and ng serve --open for the **Angular** application.

**Caution!**

Be sure to zip and submit your entire express\_lesson\_nine directory when finished!

Drop files to upload, or click to browse.

Submit Project