

create-react-app Lab

Let's create the basis for our React movie reservation system. We'll start off by creating the bare-bones Angular app and then add some components to it.

Creating a starter site

1. Open a command window and cd to the root of your project. cd to the directory above server. You'll know you're in the right place if you do a listing of files and you see "server".
2. In this folder, create a new React app called "client". You can either use npx to run create-react-app or use npm to install it, then run create-react-app. Either way, create a new React project called "client". Remember, this command may take a while to run mostly because of the npm installation of the supporting libraries.
3. Notice that there is now a new directory called client. cd to it. Look around at the files created.

Adding static assets to your site

Do you see the folder called public? This is where static assets go, things like fonts, movies, and images.

4. Look in the starters folder and you'll see a sub-folder called img. Move that folder (and all of its sub-folders) under public/img.
5. Take a look at what is in there. You should see some movie posters. Whenever Dinner-and-a-Movie gets a new movie poster, we can put it in there so it'll be accessible to our web application. We'll use these in a future lab exercise.

Using the development server

To make it easy on us as we write code, we won't use the Node/Express server until later in the course. Instead we'll use the built-in server that create-react-app sets up for us.

6. Do this:
`npm start`
7. Look at the message. Make sure it says that it is running a server at a particular port.
8. Point your browser to localhost at that port. Make sure your page comes up. You should see "Welcome to React" in the browser.
9. Keeping your browser open, edit the App.js file in your favorite IDE.
10. Change the title to "Dinner And A Movie". Hit save and watch the browser window. You should see your application reload without refreshing. This is the file watcher in action.

Making a navigation menu

11. In App.js, Remove the logo and replace it with a `<div>`.
12. Inside the `<div>`, create a series of links at the top that will serve as our navigation. Create links to ...
 - Home
 - About
 - Login
 - Checkout
13. If you test these out, they won't work yet. Don't worry about that. We'll make them work later.