Step 5: Compile and Run the Application

As we installed `webpack` locally, we have to configure a script to execute it. In your `package.json` file, the following key was generated by default:

```
"scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
```

Replace the `test` script with a `webpack` script:

```
"scripts": {
    "webpack": "webpack"
  },
```

Enter `npm run webpack` in your terminal to run all the tasks specified in Step 2. The script should run without errors. Alternatively, if you would like to continue developing this demo and you don't want to bother executing Webpack after every change, configure a script in your `package.json` that will run `webpack` with a `--watch` flag:

```
"scripts": {
    "webpack": "webpack",
    "webpack:watch": "webpack --watch"
  },
```

Execute the `webpack:watch` script:

```
npm run webpack:watch
```

If you make a change in one of your JavaScript files, the distribution file is automatically recompiled. Press `Ctrl + C` to exit the `webpack:watch` task whenever you finish development.

Once `myaccount.dist.js` is created, view it in a browser window.

If you are interested in a slightly more complex setup of a React boilerplate with a development server, read this article.

If you would like to add ESLint support to your application, this article will help you.