

Running tests and producing a production-ready build

With the project we have set up in the previous lessons, we are now going to learn how to run tests within the project and how to create a production-ready build.

WE'LL COVER THE FOLLOWING



- Running tests
- Producing a production-ready build
- Wrap up

Running tests

The `App` component has an example test in the `App.test.tsx`, which checks that the component renders without crashing. This test is the **Jest** test which we will learn about later in this course.

The tests can be run in the Terminal by the following npm command:

```
npm test
```

The tests will then be run and the results output in the Terminal:

```
PASS src/App.test.tsx
  ✓ renders without crashing (21ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        1.894s
Ran all test suites.

Watch Usage
  > Press f to run only failed tests.
  > Press o to only run tests related to changed files.
  > Press q to quit watch mode.
  > Press p to filter by a filename regex pattern.
  > Press t to filter by a test name regex pattern.
  > Press Enter to trigger a test run.
```

Notice that we didn't specify which tests to run in the npm command. So, how do you think Jest knows which tests to run?

 Show Answer

The tests will be automatically run again when changes are made to the app.

How can we just run the tests in a specific test file?

 Show Answer

The *q* key can be pressed to stop the tests from automatically running.

Producing a production-ready build

An optimized build can be created by running the following npm command in the Terminal:

```
npm run build
```

After a few seconds, a **build** folder will appear in the project containing all the app's artifacts. So, this folder can be copied to the web server as part of a deployment.

Where are the files that contain all the bundled JavaScript?

 Show Answer

Wrap up

That concludes this lesson and category on creating projects with the Create React App tool. As we have experienced, this tool adds and configures a lot of really useful dependencies for us. This saves us a lot of time that would be spent having to do this manually. Great stuff!

spent having to do this manually. Great stuff.

Next, let's double-check what we have learned with a quiz.