

Module 8: React Application Testing and Deployment

Demo Document 1

edureka!

edureka!

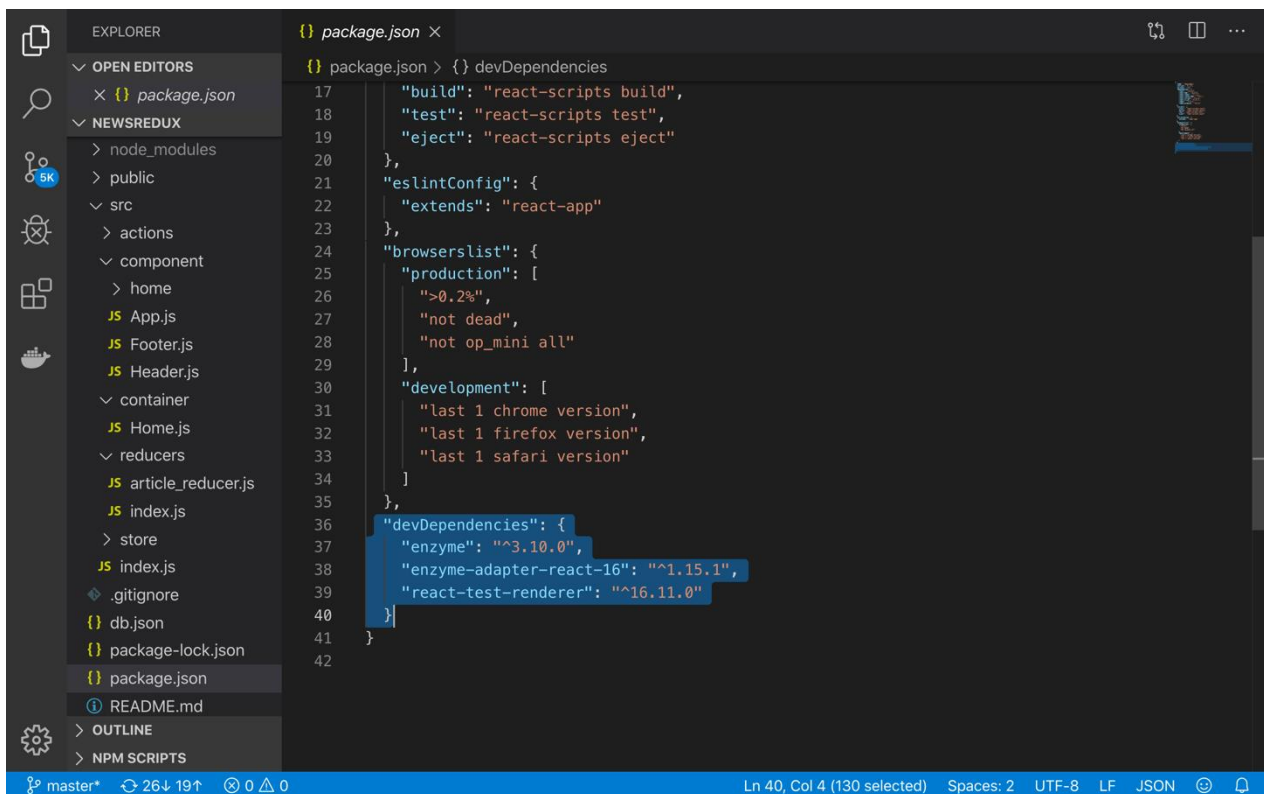
© Brain4ce Education Solutions Pvt. Ltd.

Snapshot Testing and Integrating Test Reducers Using Jest and Enzyme

Step 1: Install **enzyme** and other dependencies for testing. Jest is by default available in the react seed.

```
Avyaans-MacBook-Pro:newsredux avi$ npm i enzyme enzyme-adapter-react-16 react-test-renderer --save-dev
```

Step 2: These dependencies will appear under the **dev dependencies** in package.json file.

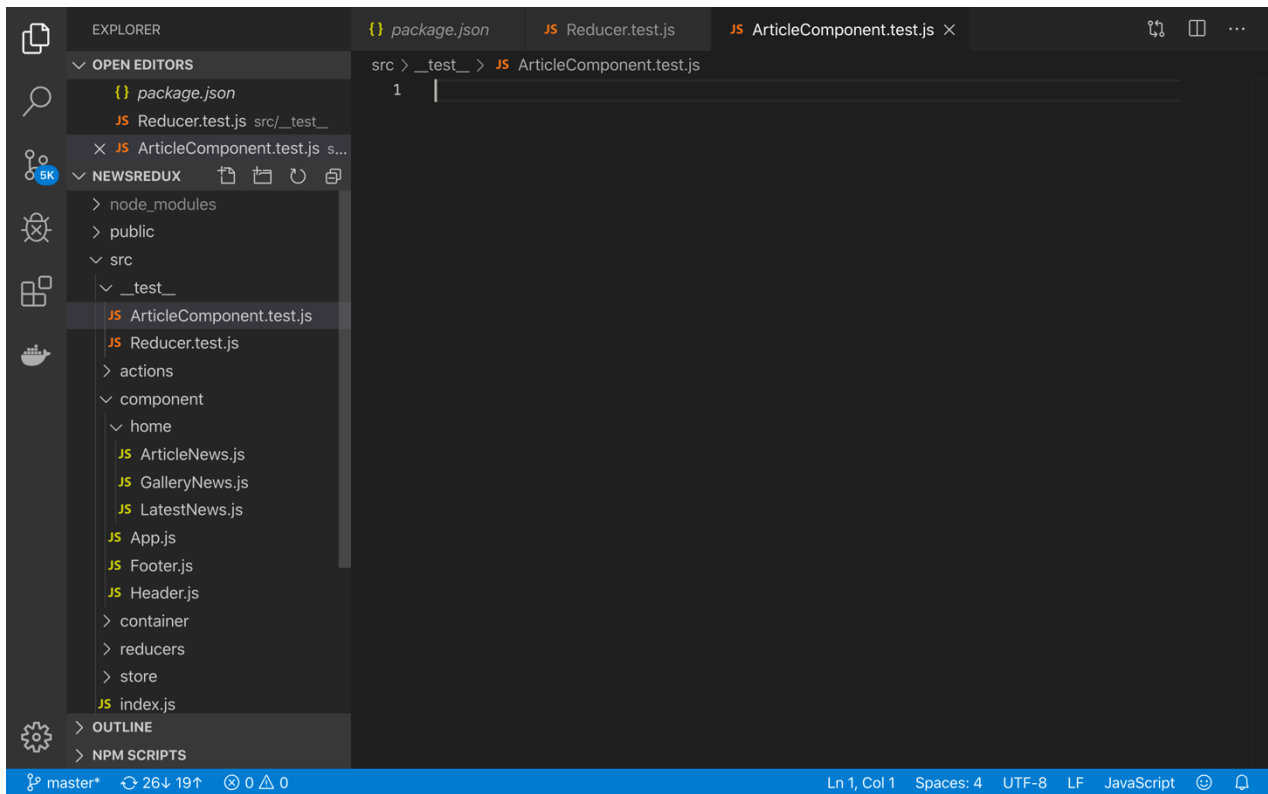


The screenshot shows the Visual Studio Code editor with the `package.json` file open. The `devDependencies` section is highlighted in blue, showing the following dependencies:

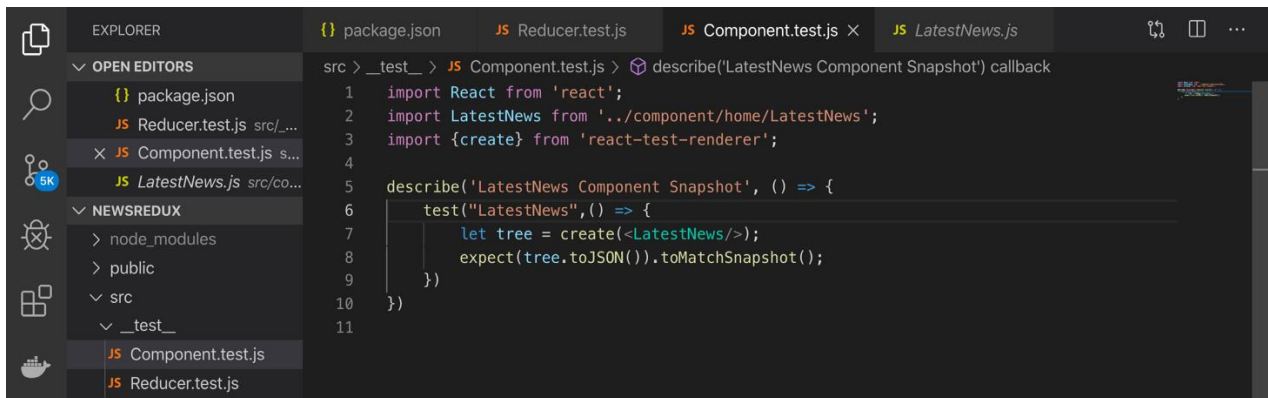
```
{
  "build": "react-scripts build",
  "test": "react-scripts test",
  "eject": "react-scripts eject",
  "eslintConfig": {
    "extends": "react-app"
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  },
  "devDependencies": {
    "enzyme": "^3.10.0",
    "enzyme-adapter-react-16": "^1.15.1",
    "react-test-renderer": "^16.11.0"
  }
}
```

Step 3: Create **__test__** folder in the source folder and all the files in this folder should end with **.test.js**.

Module 8: React Application Testing and Deployment



Step 4: In the **component.test.js** file, add the snapshot test where we will compare the old snapshot with new snapshot.



Step 5: To run the test case use command '*npm run test*', this will execute the test code from the source folder.

```
Avyaans-MacBook-Pro:newsredux avi$ npm run test
```

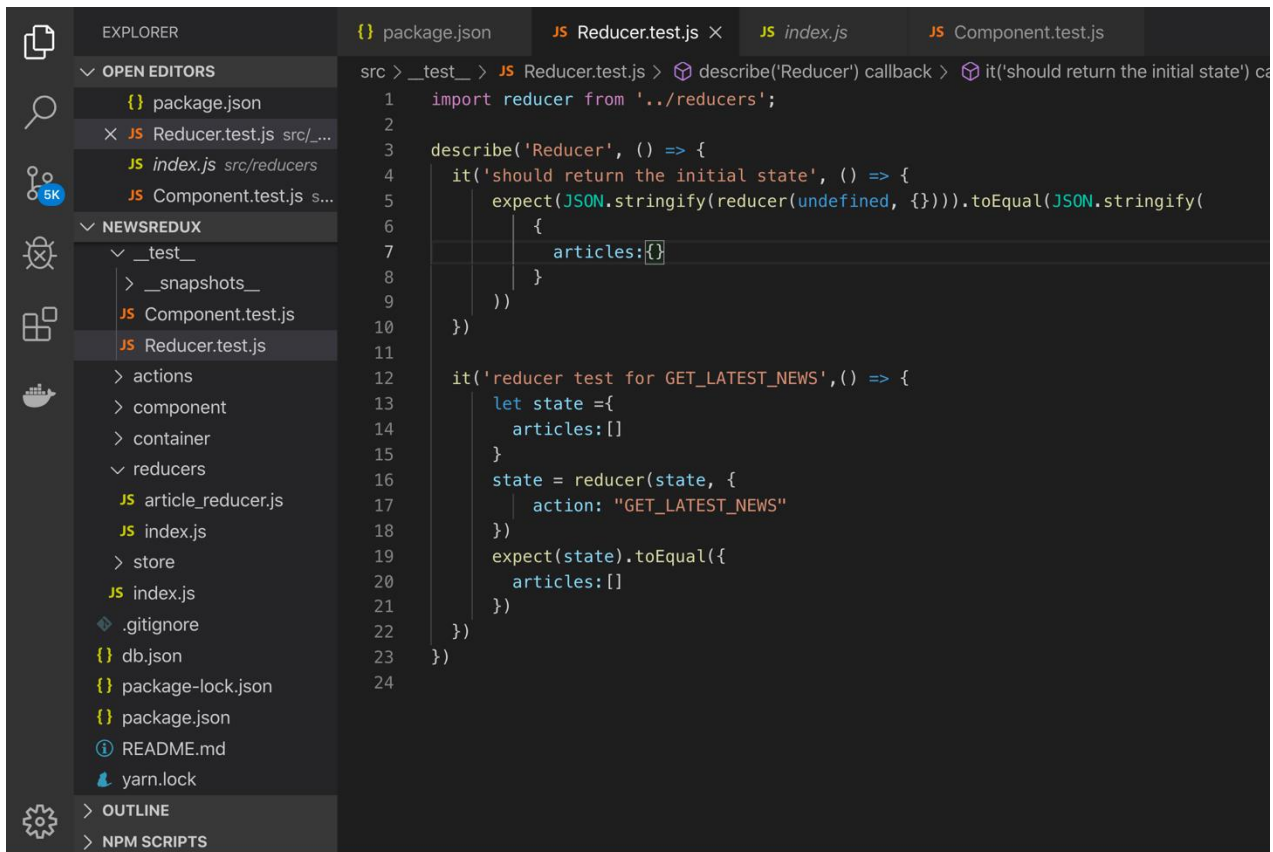
Step 6: In the console we will get the test response.

```
PASS src/__test__/Component.test.js
  LatestNews Component Snapshot
    ✓ LatestNews (14ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  1 passed, 1 total
Time:        1.152s
Ran all test suites related to changed files.

Watch Usage
  > Press a to run all tests.
  > Press f to run only failed tests.
  > 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.
```

Step 7: Now we will write the test case of the *reducer*. Here we have 2 test cases, one for default state check and second for the output after check type.



The screenshot shows the VS Code interface with the Explorer sidebar on the left and the Editor view on the right. The Explorer sidebar is expanded to show the 'reducers' directory, which contains 'article_reducer.js', 'index.js', and 'store'. The Editor view displays the 'Reducer.test.js' file, which contains the following code:

```
src > __test__ > JS Reducer.test.js > describe('Reducer') callback > it('should return the initial state') ca
1  import reducer from '../reducers';
2
3  describe('Reducer', () => {
4    it('should return the initial state', () => {
5      expect(JSON.stringify(reducer(undefined, {}))).toEqual(JSON.stringify(
6        {
7          articles: {}
8        }
9      ))
10   })
11 })
12
13 it('reducer test for GET_LATEST_NEWS', () => {
14   let state = {
15     articles: []
16   }
17   state = reducer(state, {
18     action: "GET_LATEST_NEWS"
19   })
20   expect(state).toEqual({
21     articles: []
22   })
23 })
24 }
```

Step 8: Test framework will automatically detect the change and run all the new added test cases.

```
PASS src/__test__/Reducer.test.js
PASS src/__test__/Component.test.js

Test Suites: 2 passed, 2 total
Tests:       3 passed, 3 total
Snapshots:   1 passed, 1 total
Time:        2.505s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.
```

Conclusion:

We have successfully performed the testing on React application using Jest and Enzyme.

edureka!