npm install localforage

unpkg-path-plugin.tsx

import \* as esbuild from 'esbuild-wasm';

import axios from 'axios';

import localForage from 'localforage';

const fileCache = localForage.createInstance({

    name: 'filecache',

});

(async () => {

    await fileCache.setItem('color', 'red');

    const color = await fileCache.getItem('color');

    console.log(color);

})();

Go to the Application tab in dev tools browser and there you will see the IndexedDB and the database.

A screenshot of a computer

Description automatically generated

Here we can see that we stored a key of ‘color’ and value of ‘red’.

A screenshot of a computer

Description automatically generated

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Note:**

**Legacy Peer Deps Flag to Avoid Errors**

In the upcoming lecture, we will be installing the Monaco editor. Since almost all students' projects will be using React v18, you will need to pass the **legacy-peer-deps** flag to the install command:

npm install --save-exact @monaco-editor/react@3.7.5 **--legacy-peer-deps**

**IMPORTANT - You will now need to pass the --legacy-peer-deps flag to every npm install command that is made going forward in this course.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Prettier Installation Required Update**

In the upcoming lecture, we will be installing Prettier to use with our code editor. We will need to make sure to install very specific versions to be compatible with the code we are writing and the other dependencies that have already been installed:

npm install prettier@2 @types/prettier@2 --legacy-peer-deps

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Required Webpack v5 Polyfill Fixes**

In the upcoming lecture, we will be installing some more dependencies for our editor. Unfortunately, Create React App v5 has many conflicts with some of these libraries even with the latest versions. To avoid ejecting, or, downgrading to CRA v4 we can instead make use of the react-app-rewired library.

**Required steps:**

1. Install dependencies

npm install --save-dev react-app-rewired constants-browserify assert os-browserify --legacy-peer-deps

2. In the root of your project (where the package.json file is located) create a **config-overrides.js** file and add the following code:

1. module.exports = function override(config, env) {
2. config.resolve.fallback = {
3. ...config.resolve.fallback,
4. constants: require.resolve('constants-browserify'),
5. assert: require.resolve('assert/'),
6. os: require.resolve('os-browserify/browser'),
7. };
8. config.ignoreWarnings = [
9. {
10. message:
11. /Critical dependency: the request of a dependency is an expression/,
12. },
13. ];
14. return config;
15. };

3 Update your package.json scripts to the following:

1. "start": "react-app-rewired start",
2. "build": "react-app-rewired build",
3. "test": "react-app-rewired test",
4. "eject": "react-app-rewired eject"

These changes will provide the missing native Node polyfills that Webpack v5 removed.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**New Versions of React Resizable to Support React 18**

In the upcoming lecture, we will be installing specific versions of React Resizable and its types definitions. These will only work with older versions of React, so, we must install the latest versions to work with React v18.

The updated install command should now be:

npm install --save-exact react-resizable@**3.0.4** @types/react-resizable@**3.0.2** **--legacy-peer-deps**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Adding children to Interface for React 18**

In the upcoming lecture, we will be implementing our Resizable component. Similar to what was discussed [earlier in the course](https://www.udemy.com/course/react-and-typescript-build-a-portfolio-project/learn/lecture/33361854#questions), we will need to include children in the interface along with our other props.

Find the **resizable.tsx** file and make sure to update your ResizableProps interface to the following:

1. interface ResizableProps {
2. direction: "horizontal" | "vertical";
3. children?: React.ReactNode;
4. }

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Small Update for Try / Catch Block**

In the upcoming lecture, we will be adding some error handling to our bundler. You will likely see a TS error in the catch block that says *Object is of type 'unknown'*.

To address this, we will need to refactor the catch block of our **src/bundler/index.ts**:

1. ...
2. } catch (err) {
3. if (err instanceof Error) {
4. return {
5. code: "",
6. err: err.message,
7. };
8. } else {
9. throw err;
10. }
11. }

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Updated Install Command and Versions for React 18**

Due to some breaking changes in the recent release of React v18, the versions shown in the project setup will no longer work. We will need to install the latest versions of Redux, React Redux, and @types/react-redux:

npm install --save-exact react-redux redux @types/react-redux redux-thunk --legacy-peer-deps

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Property 'order' does not exist on type 'CellsState | undefined'.**

In the upcoming lecture, we will be working on adding our CellList component. You may see the following TS error:

Property 'order' does not exist on type 'CellsState | undefined'.

Property 'data' does not exist on type 'CellsState | undefined'.

As raised in the QA [here](https://www.udemy.com/course/react-and-typescript-build-a-portfolio-project/learn/lecture/24357106#questions/13899618), you will need to add initialState as a second argument to the produce function of **cellsReducer.ts**:

1. const reducer = produce((state: CellsState = initialState, action: Action) => {
2. ...
3. }
4. }, initialState);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*