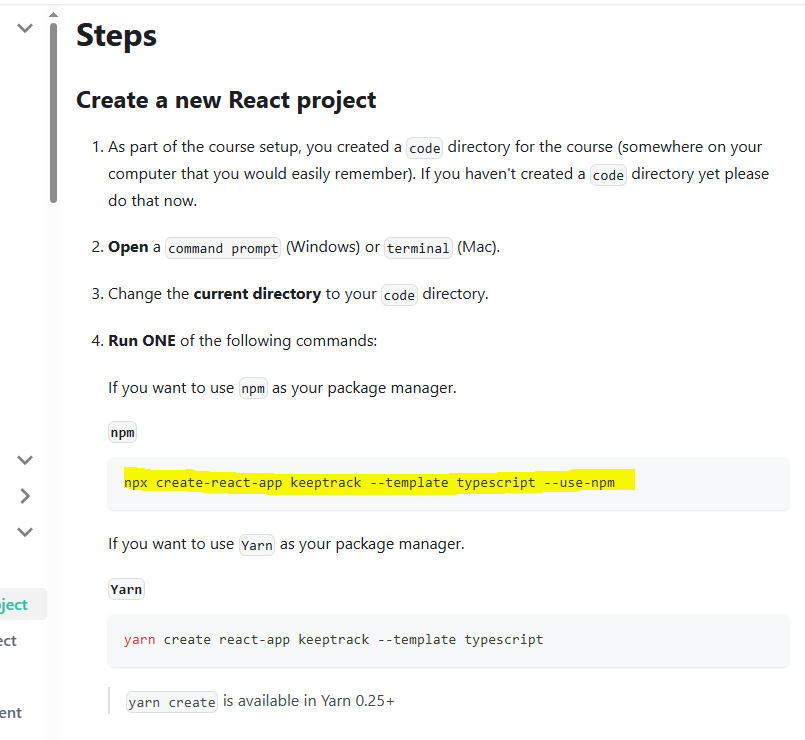
<https://handsonreact.com/docs/labs/ts/CreatingNewProject>

Creating a New Project  
Lab-1

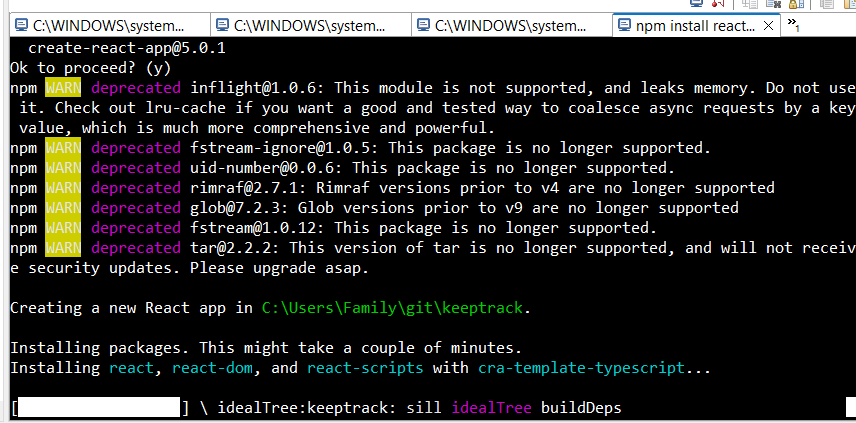
Objectives[**​**](https://handsonreact.com/docs/labs/ts/CreatingNewProject#objectives)

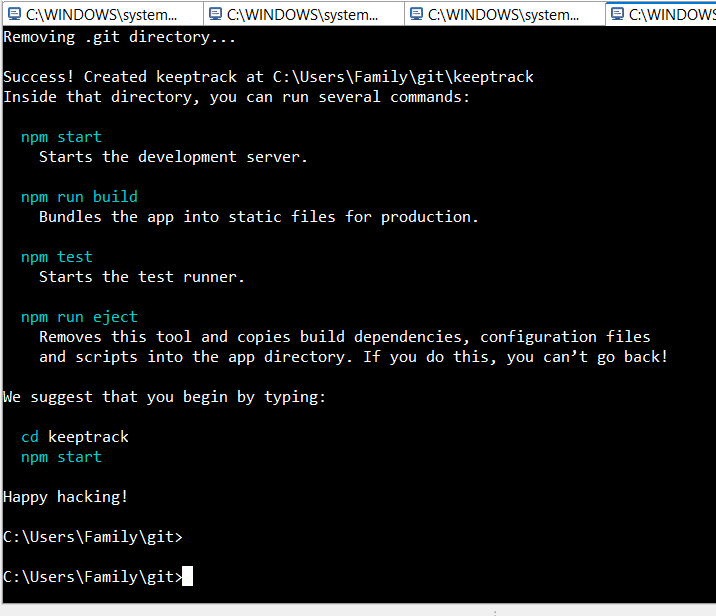
* Create a new React project (app) that uses TypeScript
* Open the new project
* Review the default project structure

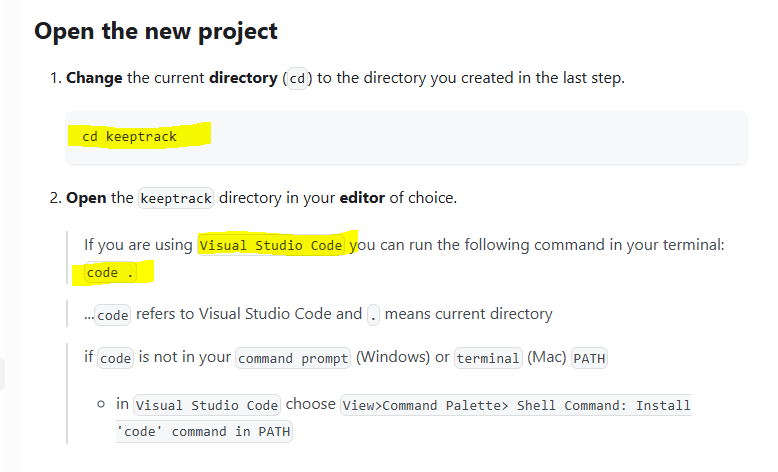


Steps[**​**](https://handsonreact.com/docs/labs/ts/CreatingNewProject#steps)

* **npx create-react-app keeptrack --template typescript --use-npm**

C:\Users\Family\git>npx create-react-app keeptrack --template typescript --use-npm  


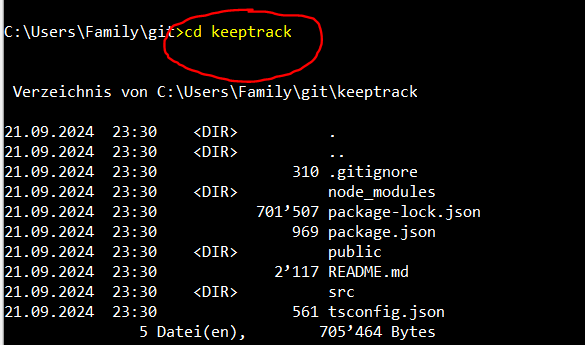




Open the new project[**​**](https://handsonreact.com/docs/labs/ts/CreatingNewProject#open-the-new-project)

Change the current directory (cd) to the directory you created in the last step.

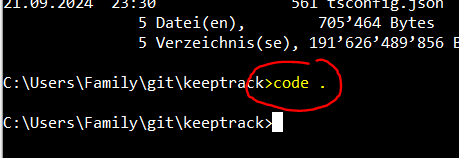
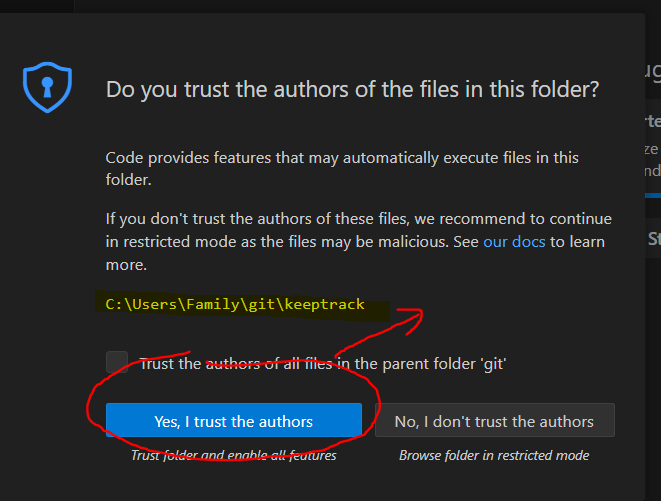
* **cd keeptrack**

C:\Users\Family\git>cd keeptrack  


Open the keeptrack directory in your editor of choice.

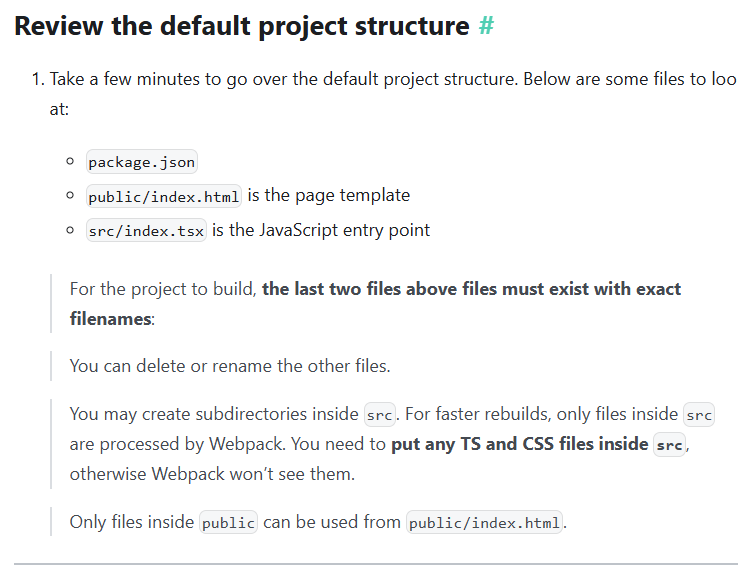
If you are using Visual Studio Code you can run the following command in your terminal: code .

* **code .**

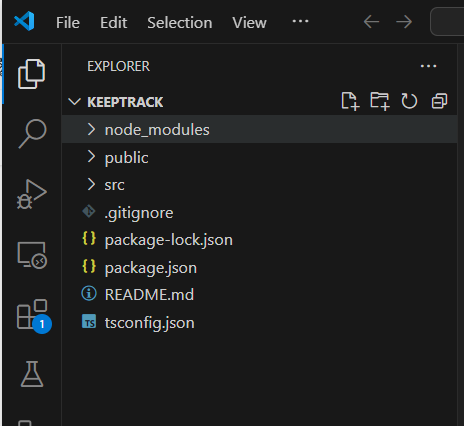
C:\Users\Family\git\keeptrack>code .  
  
  
  
  
...code refers to **Visual Studio Code** and . means current directory

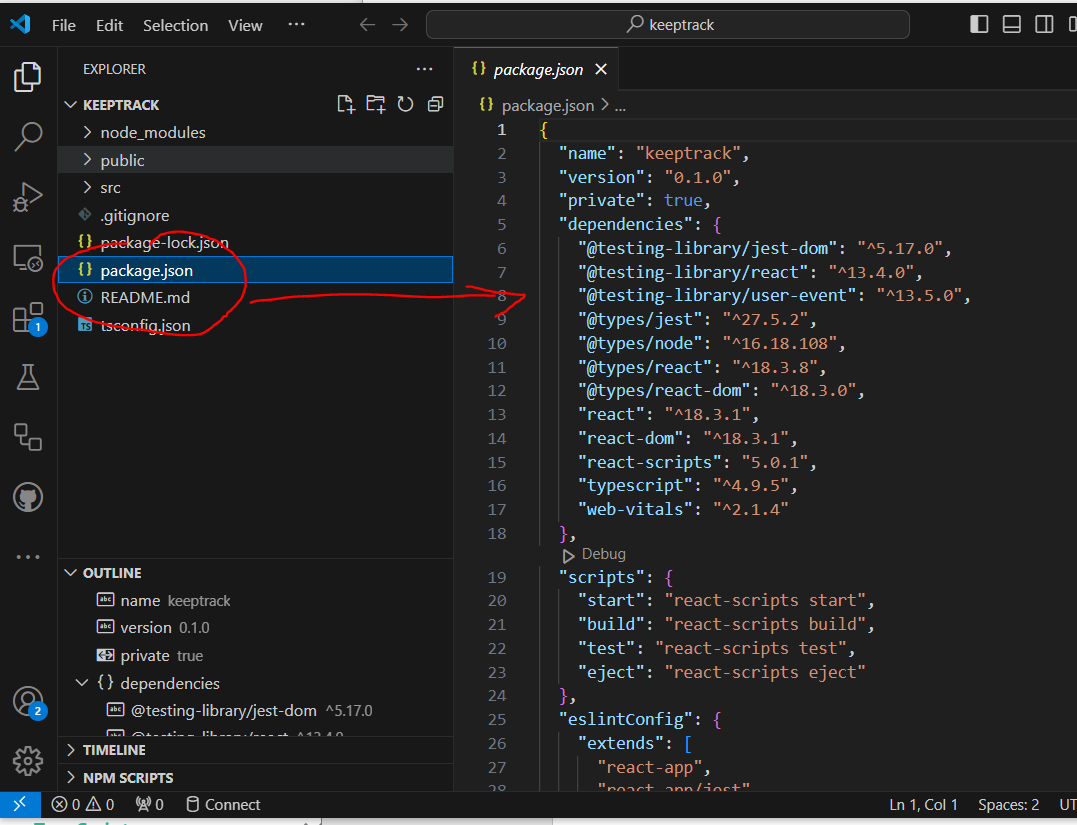
if code is not in your command prompt (Windows) or terminal (Mac) PATH

in Visual Studio Code choose View>Command Palette> Shell Command: Install 'code' command in PATH



Review the default project structure[**​**](https://handsonreact.com/docs/labs/ts/CreatingNewProject#review-the-default-project-structure)

Take a few minutes to go over the default project structure. Below are some files to look at:  


**package.json**  


**package.json**

{

  "name": "keeptrack",

  "version": "0.1.0",

  "private": true,

  "dependencies": {

    "@testing-library/jest-dom": "^5.17.0",

    "@testing-library/react": "^13.4.0",

    "@testing-library/user-event": "^13.5.0",

    "@types/jest": "^27.5.2",

    "@types/node": "^16.18.108",

    "@types/react": "^18.3.8",

    "@types/react-dom": "^18.3.0",

    "react": "^18.3.1",

    "react-dom": "^18.3.1",

    "react-scripts": "5.0.1",

    "typescript": "^4.9.5",

    "web-vitals": "^2.1.4"

  },

  "scripts": {

    "start": "react-scripts start",

    "build": "react-scripts build",

    "test": "react-scripts test",

    "eject": "react-scripts eject"

  },

  "eslintConfig": {

    "extends": [

      "react-app",

      "react-app/jest"

    ]

  },

  "browserslist": {

    "production": [

      ">0.2%",

      "not dead",

      "not op\_mini all"

    ],

    "development": [

      "last 1 chrome version",

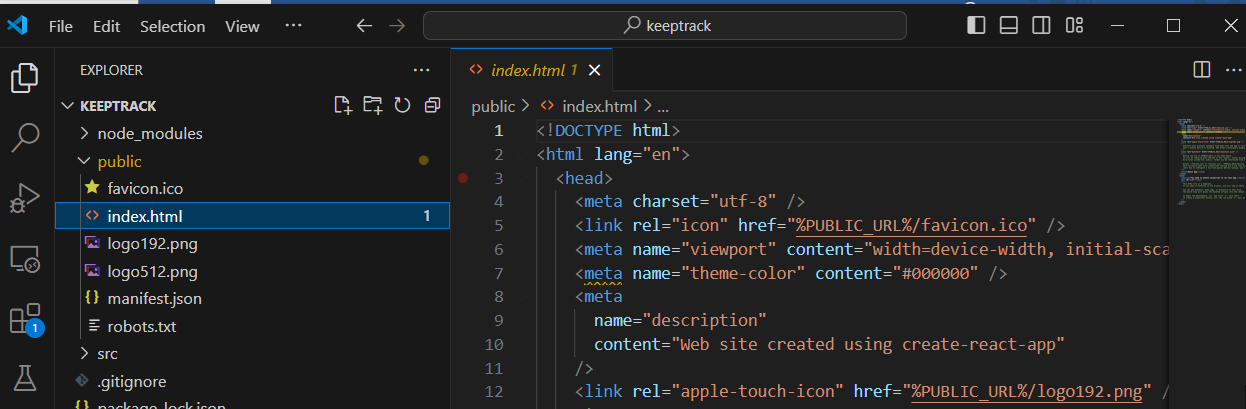
      "last 1 firefox version",

      "last 1 safari version"

    ]

  }

}

**public/index.html is the page template**

**Public/index.html**  
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <meta name="theme-color" content="#000000" />

    <meta

      name="description"

      content="Web site created using create-react-app"

    />

    <link rel="apple-touch-icon" href="%PUBLIC\_URL%/logo192.png" />

    <!--

      manifest.json provides metadata used when your web app is installed on a

      user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/

    -->

    <link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

    <!--

      Notice the use of %PUBLIC\_URL% in the tags above.

      It will be replaced with the URL of the `public` folder during the build.

      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will

      work correctly both with client-side routing and a non-root public URL.

      Learn how to configure a non-root public URL by running `npm run build`.

    -->

    <title>React App</title>

  </head>

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div id="root"></div>

    <!--

      This HTML file is a template.

      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.

      The build step will place the bundled scripts into the <body> tag.

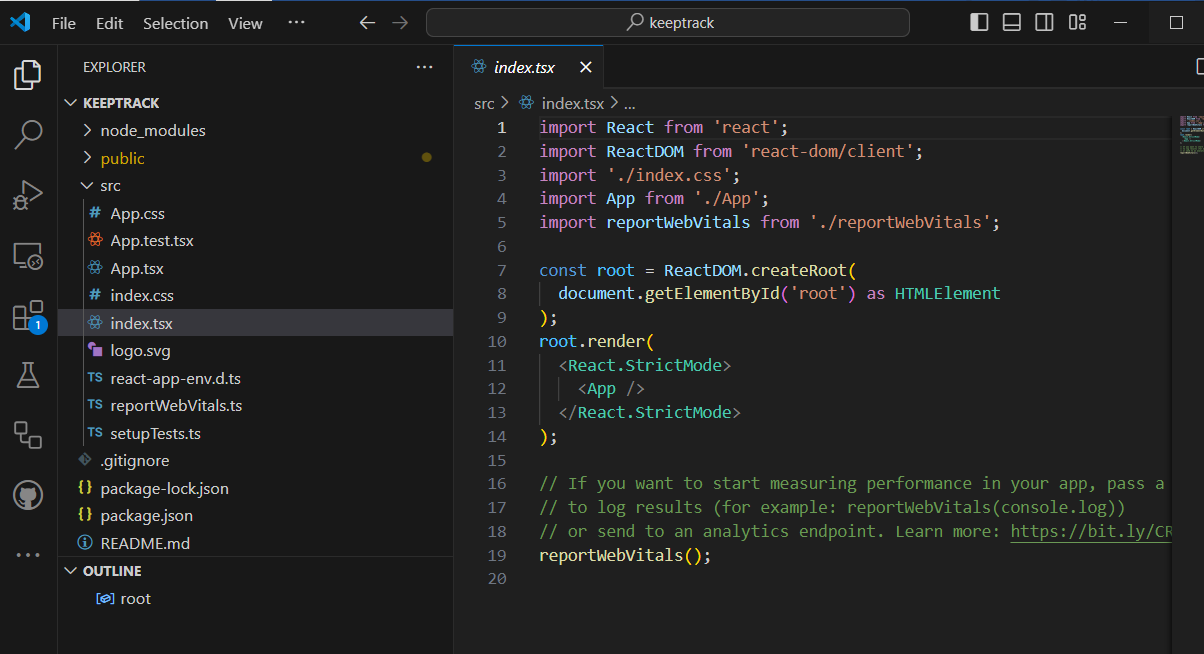
      To begin the development, run `npm start` or `yarn start`.

      To create a production bundle, use `npm run build` or `yarn build`.

    -->

  </body>

</html>

**src/index.tsx is the JavaScript entry point**  


**src/index.tsx**

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(

  document.getElementById('root') as HTMLElement

);

root.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>

);

// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals

reportWebVitals();

