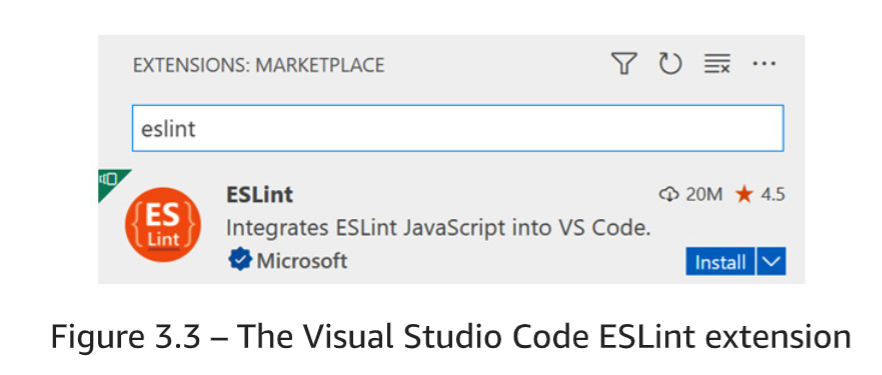
Thiết lập React với Tailwind (v4)

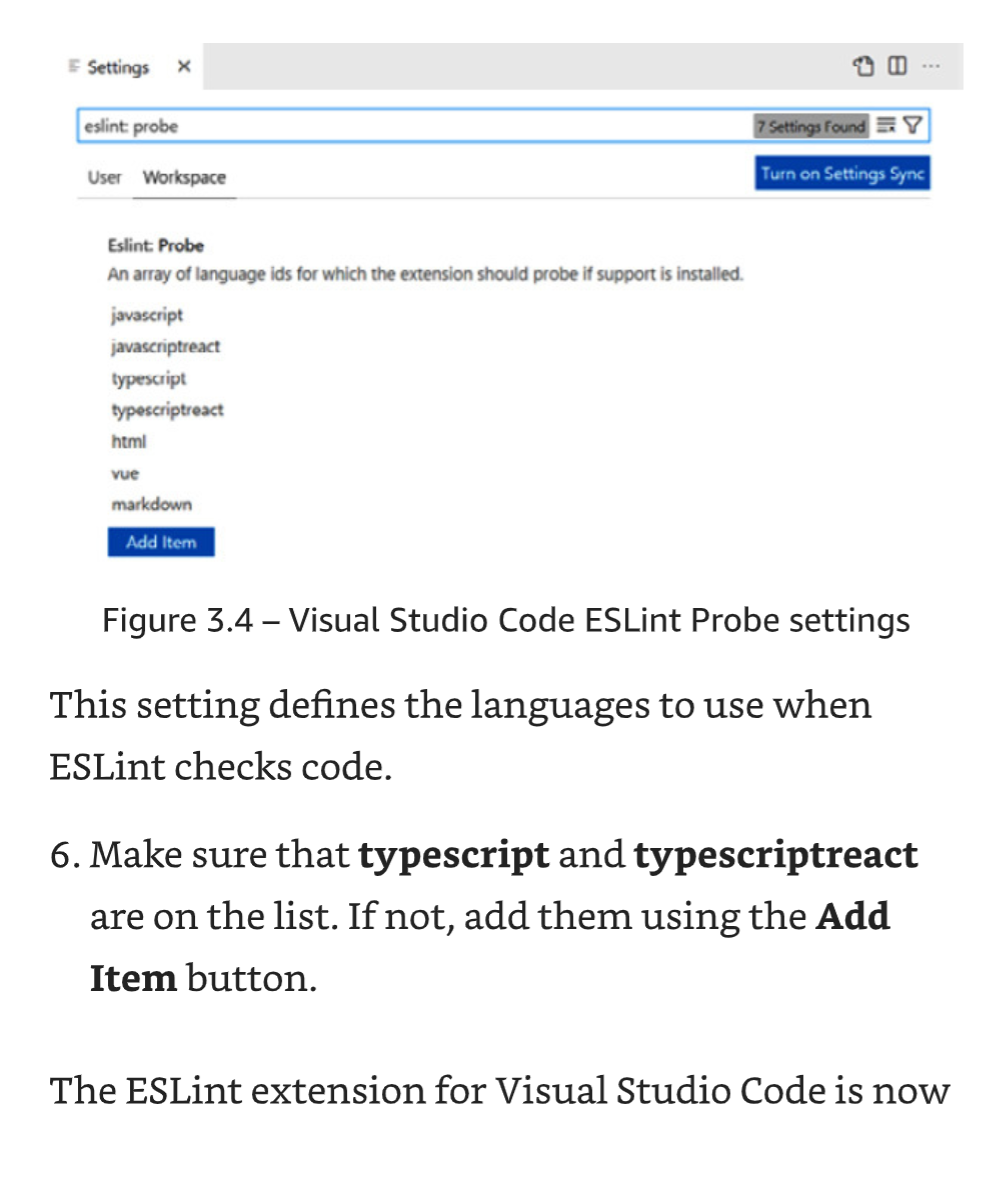
Link tham khảo

1. <https://medium.com/@osamajavaid/setting-up-react-19-with-tailwind-css-v4-using-vite-in-just-two-steps-3748f55b06fd>
2. <https://duthanhduoc.com/blog/tao-du-an-react-vite-typescript-eslint>
3. https://medium.com/@vitor.vicen.te/setting-up-jest-js-for-a-vite-ts-js-react-project-the-ultimate-guide-7816f4c8b738

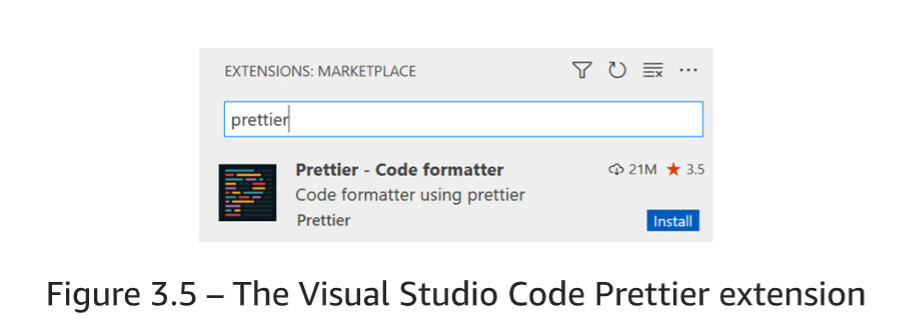
Các Extension cài đặt cho VS Code

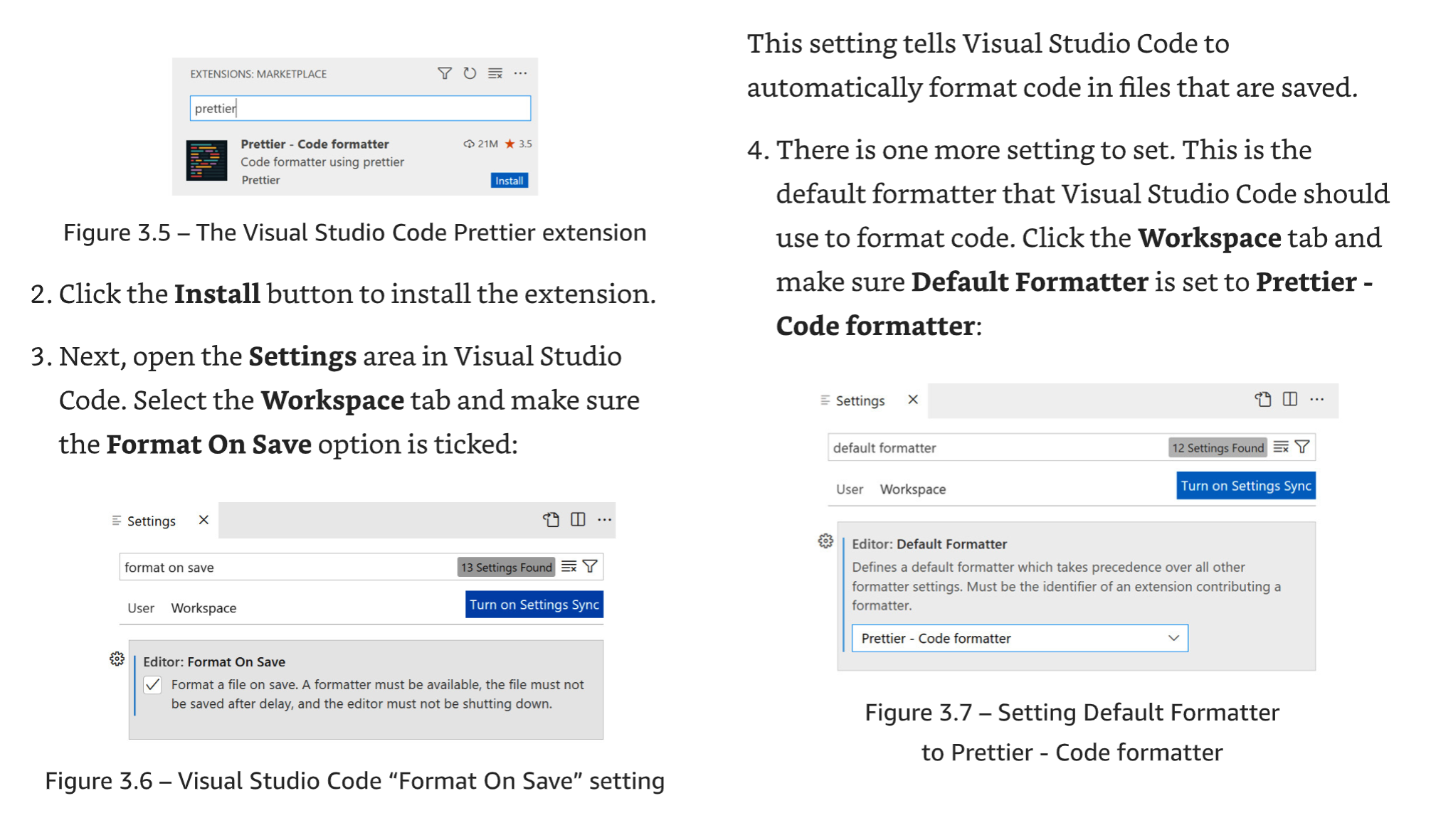
1. Eslint





1. Prettier

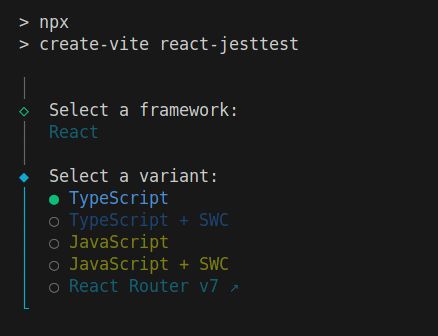




**Khởi tạo**

npm create vite@latest my-react-app





**Thêm TailwindCSS**

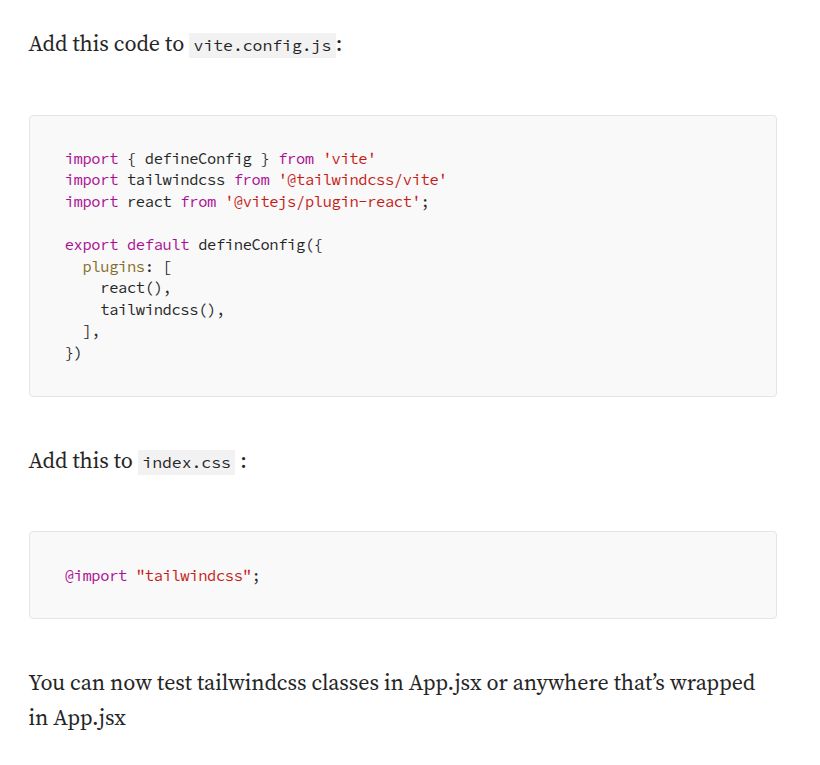
npm install tailwindcss @tailwindcss/vite

---

import { defineConfig } from 'vite'  
import tailwindcss from '@tailwindcss/vite'  
import react from '@vitejs/plugin-react';  
  
export default defineConfig({  
plugins: [  
react(),  
tailwindcss(),  
],  
})

---

@import "tailwindcss";



**Test TailwindCSS**

import React from 'react'  
import './App.css'  
  
function App() {  
 return (  
 <h1 class="text-3xl bg-red-700 font-bold underline">  
 React 19 with Tailwind CSS v4  
 </h1>  
 )  
}  
  
export default App

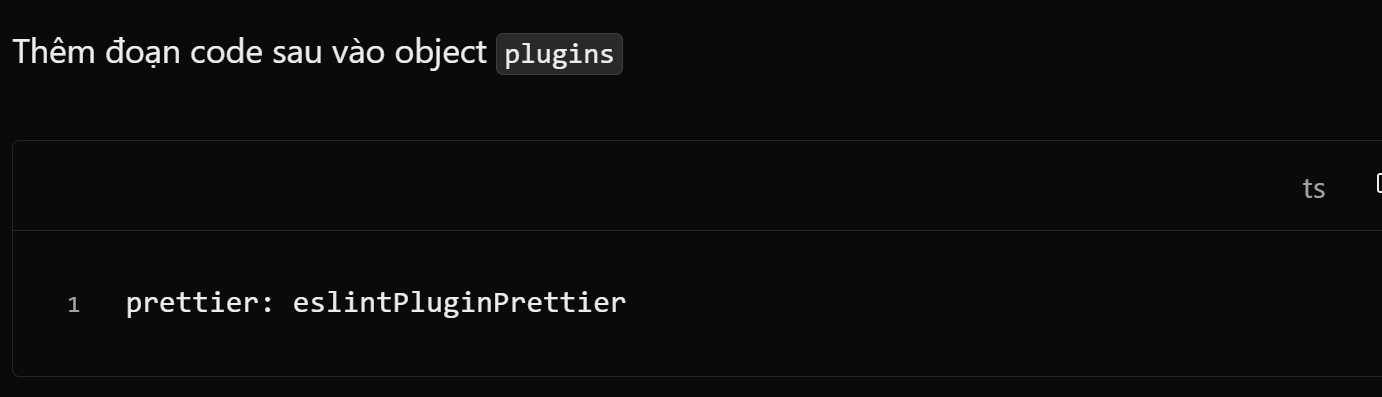
**Cài đặt ESLint và Prettier cho format code**

Add ESLint and Prettier

npm i prettier eslint-config-prettier eslint-plugin-prettier -D

Mở file eslint.config.js lên





Thêm đoạn code sau vào object rules để thêm các rule của Prettier



'vite.config.ts'

---

import eslintPluginPrettier from 'eslint-plugin-prettier'

---

prettier: eslintPluginPrettier

---

"prettier/prettier": [

"warn",

{

arrowParens: "always",

semi: true,

trailingComma: "all",

tabWidth: 2,

endOfLine: "auto",

useTabs: false,

singleQuote: true,

printWidth: 120,

jsxSingleQuote: true,

},

],

Tạo file .prettierrc trong thư trong thư mục root với nội dung dưới đây

{

"arrowParens": "always",

"semi": true,

"trailingComma": "all",

"tabWidth": 2,

"endOfLine": "auto",

"useTabs": false,

"singleQuote": true,

"printWidth": 120,

"jsxSingleQuote": true

}

Tiếp theo Tạo file .prettierignore ở thư mục root

node\_modules/

dist/

Thêm compilerOptions vào tsconfig.json



  "compilerOptions": {

    "baseUrl": ".",

    "paths": {

      "~/\*": ["src/\*"]

    }

  }

**Config Jest**

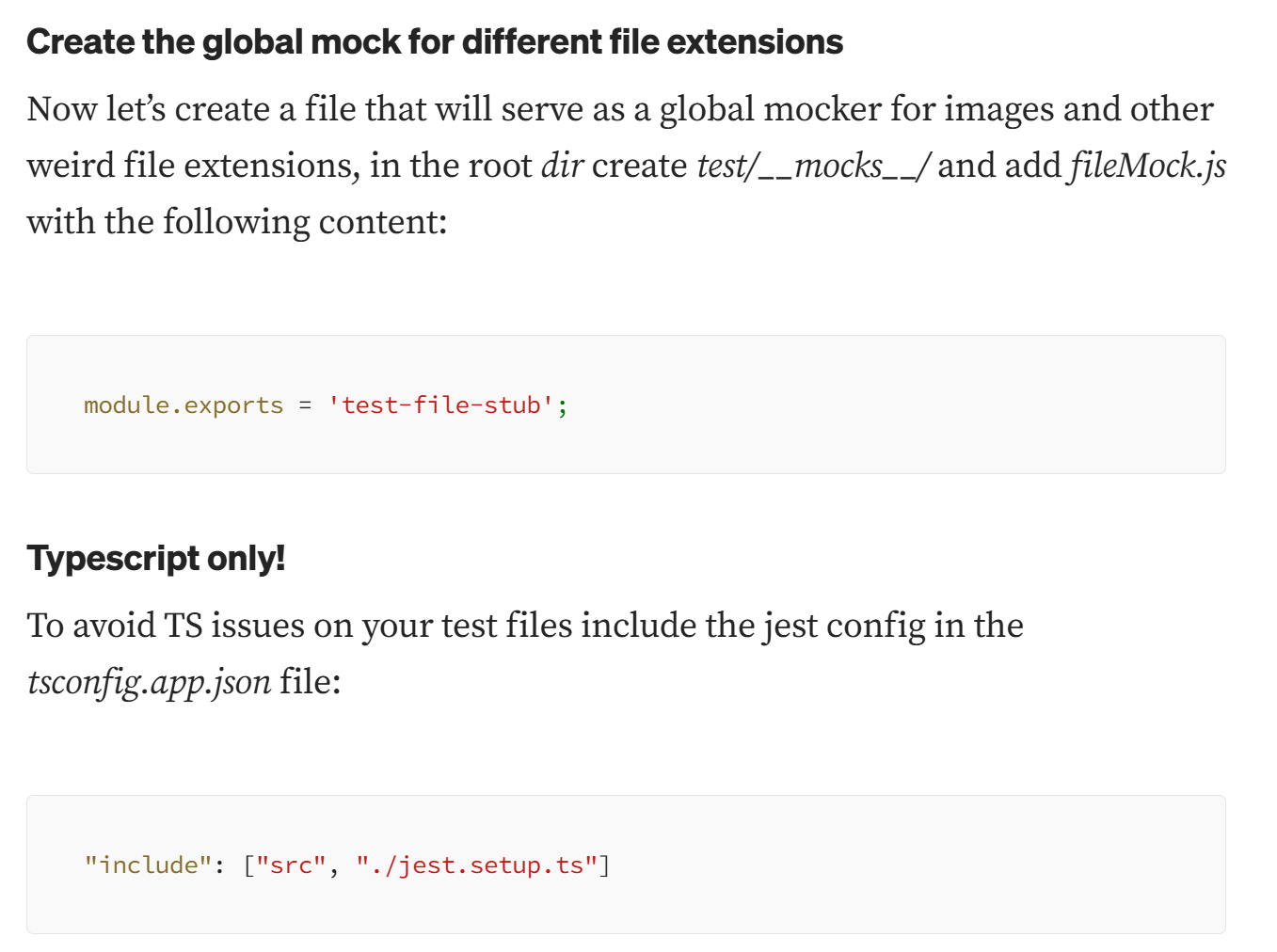
npm install -D jest @types/jest @testing-library/react @testing-library/jest-dom @testing-library/user-event @babel/core @babel/preset-env @babel/preset-react @babel/preset-typescript babel-jest identity-obj-proxy jest-environment-jsdom ts-node

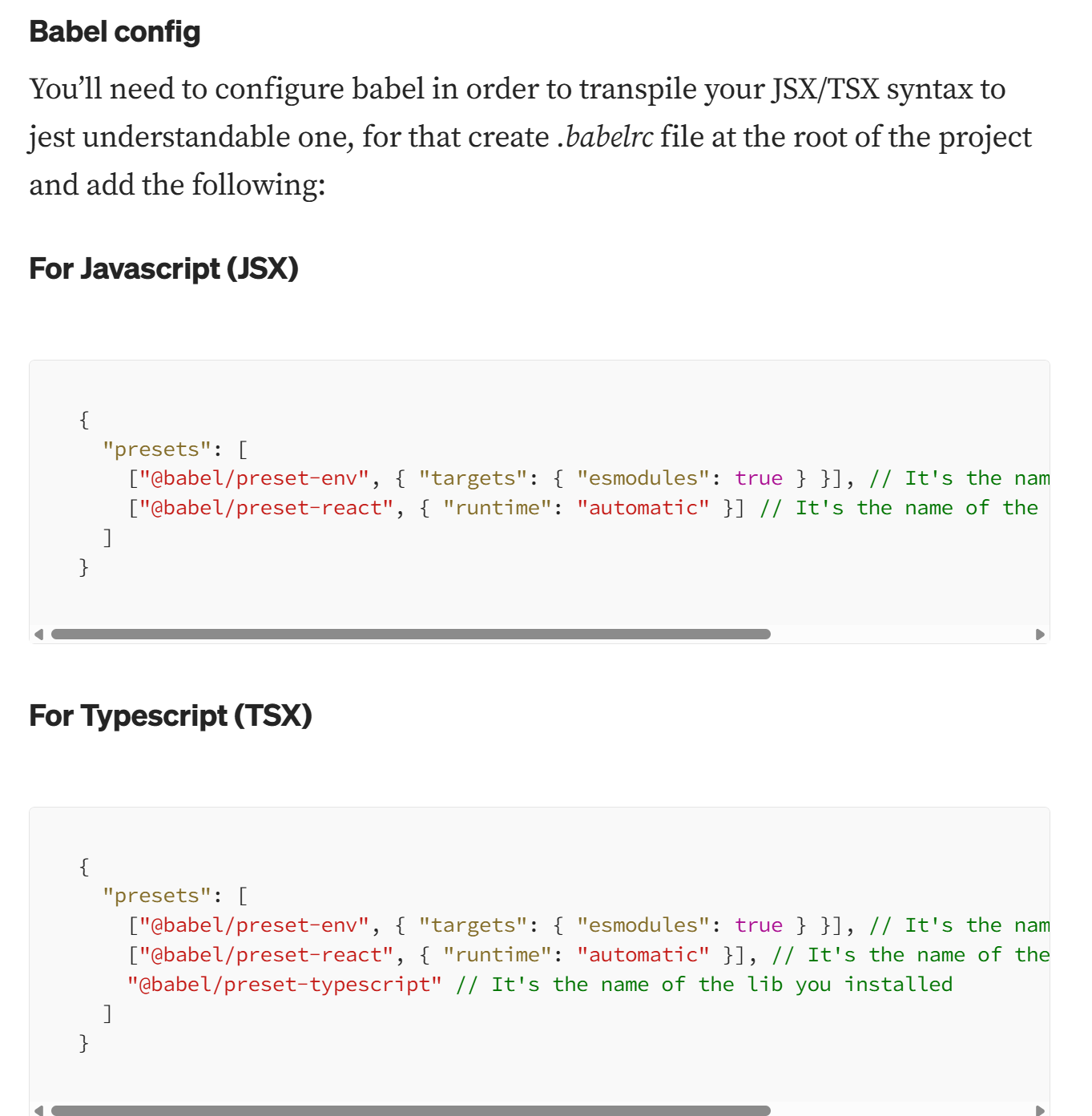


import "@testing-library/jest-dom";

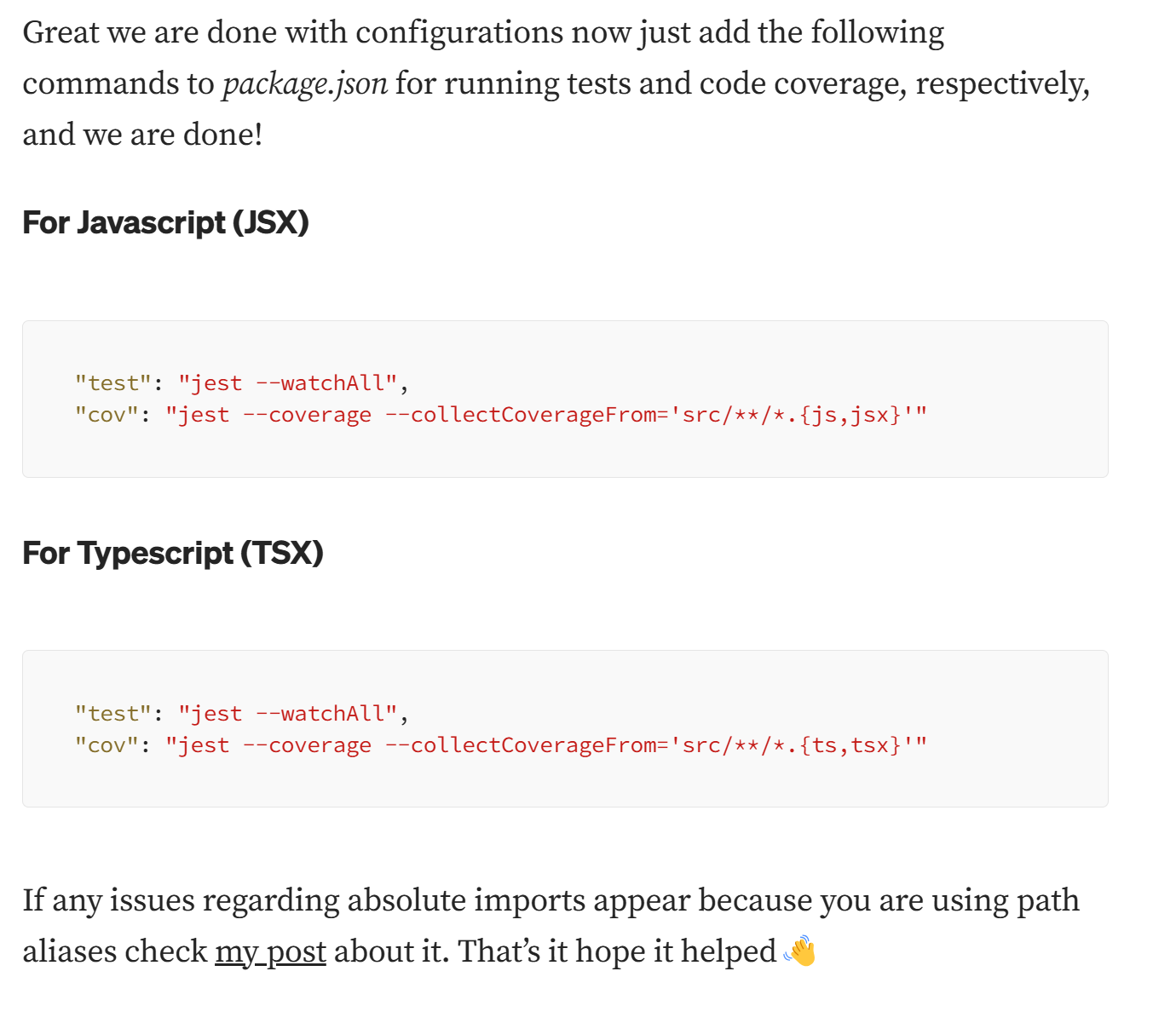


export default {  
 testEnvironment: "jest-environment-jsdom", // Same name of the lib you installed  
 setupFilesAfterEnv: ["<rootDir>/jest.setup.ts"], // The file you created to extend jest config and "implement" the jest-dom environment in the jest globals  
 moduleNameMapper: {  
 "\\.(gif|ttf|eot|svg|png)$": "<rootDir>/test/\_\_mocks\_\_/fileMock.js", // The global stub for weird files  
 "\\.(css|less|sass|scss)$": "identity-obj-proxy", // The mock for style related files  
 "^@/(.\*)$": "<rootDir>/src/$1", // [optional] Are you using aliases?  
 },  
};





{  
 "presets": [  
 ["@babel/preset-env", { "targets": { "esmodules": true } }], // It's the name of the lib you installed  
 ["@babel/preset-react", { "runtime": "automatic" }], // It's the name of the lib you installed  
 "@babel/preset-typescript" // It's the name of the lib you installed  
 ]  
}



"test": "jest --watchAll",  
"cov": "jest --coverage --collectCoverageFrom='src/\*\*/\*.{ts,tsx}'"