# Step-by-Step Instructions for Setting Up a React App with Vite and Testing with TypeScript

1. Create a new React App using Vite with Typescript

npm create vite@latest my-react-app --template react-ts

npm create vite@latest my-react-app --template react-ts

1. Navigate to the project directory

cd my-react-app

cd my-react-app

1. Install dependencies

npm install

1. Install testing dependencies

npm install --save-dev @testing-library/react @testing-library/jest-dom @types/jest ts-jest jest jest-environment-jsdom

1. Create a jest.config.**mjs** with the following inside:

export default {

testEnvironment: 'jest-environment-jsdom',

setupFilesAfterEnv: ['<rootDir>/src/setupTests.ts'],

moduleNameMapper: {

'\\.(css|less|scss|sass)$': 'identity-obj-proxy',

},

transform: {

'^.+\\.tsx?$': 'ts-jest',

},

};

1. Install babel dependencies

npm install --save-dev @babel/core @babel/preset-env @babel/preset-react @babel/preset-typescript

1. Create a babel.config.**cjs** file in the root directory

module.exports = {

presets: [

'@babel/preset-env',

'@babel/preset-react',

'@babel/preset-typescript',

],

};

1. Create a file called setupTests.ts in the src folder. Write this code in it.

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

1. Update package.json

  "scripts": {

    "dev": "vite",

    "build": "tsc -b && vite build",

    "lint": "eslint . --ext ts,tsx --report-unused-disable-directives --max-warnings 0",

    "preview": "vite preview",

    "test" : "jest"

  }

1. Change the tsconfig.json file. Replace the whole file.

{

"compilerOptions": {

"target": "ESNext",

"module": "ESNext",

"lib": ["DOM", "DOM.Iterable", "ESNext"],

"allowJs": true,

"skipLibCheck": true,

"esModuleInterop": true,

"allowSyntheticDefaultImports": true,

"strict": true,

"forceConsistentCasingInFileNames": true,

"moduleResolution": "node",

"resolveJsonModule": true,

"isolatedModules": true,

"noEmit": true,

"jsx": "react-jsx"

},

"include": ["src"],

"exclude": ["node\_modules"]

}

1. Create your component

import React from 'react';

const HelloWorld: React.FC = () => {

return <h1>Hello, world!</h1>;

};

export default HelloWorld;

1. Create the test for your component (HelloWorld.test.tsx). You MUST import React

import React from 'react';

import { render, screen } from '@testing-library/react';

import HelloWorld from '../HelloWorld';

test('renders Hello, world!', () => {

render(<HelloWorld />);

const element = screen.getByText(/Hello, world!/i);

expect(element).toBeInTheDocument();

});

1. Run the development server in one console

npm run dev

1. Run your tests in another console

npm test