

# **Generating Your Own React Project Template**

# Getting started - SHORTCUT MODE

To skip the rest of this setup and simply *duplicate* what SDG already setup:

```
cd ~/sdg
degit suncoast-devs/react-project-template react-projet-template
cd react-project-template
git init
git add .
git commit -m "Initial commit"
hub create
git push origin HEAD
```

And you will have your own copy of SDG's project template.

# Getting started

In the same directory where you create projects:

```
npm init vite -- react-project-template
```

- Choose **react** as the *framework*
- Choose **react-ts** as the *variant*

# Check out the template it created

```
cd react-project-template
```

```
npm install
```

```
git init
```

```
git add .
```

```
git commit -m "Initial vite template"
```

```
code .
```

**Customize!**

# **Add sass for SASS/SCSS awesomeness**

- This will let us use the awesome sass/scss tooling

```
npm install --save-dev sass
```

# index.html

- Change the default `<title>`

# favicon.svg

Suggest SDG's badge graphic available [here](#)



# src/main.tsx

Change

```
import './index.scss'
```

**to**

```
import `./index.scss`
```

*and*

```
import App from './App'
```

**to**

```
import { App } from './App'
```

**src/App.css**

Delete this file

**src/index.css**

Rename the file to `src/index.scss`

# src/index.scss

Use the following starter CSS:

```
:root {  
  font: 16px / 1 sans-serif;  
}
```

```
html {  
  height: 100%;  
}
```

```
body {  
  margin: 0;  
  min-height: 100%;  
}
```

`package.json`

# Individual changes for scripts

Update the scripts section to include:

```
"start": "vite",  
"predeploy": "npm run build",  
"deploy": "netlify deploy --prod --dir=dist",
```

To view your app on the local network, use:

```
"start": "vite --host",
```

# package.json

## Configure the code formatting tool

Add this section:

```
"prettier": {  
  "trailingComma": "es5",  
  "tabWidth": 2,  
  "semi": false,  
  "singleQuote": true,  
  "useTabs": false  
}
```

# package.json

```
{
  "version": "0.0.0",
  "scripts": {
    "dev": "vite",
    "build": "tsc && eslint --ext .tsx,.ts . && vite build",
    "serve": "vite preview",
    "start": "vite",
    "predeploy": "npm run build",
    "deploy": "netlify deploy --prod --dir=dist"
  },
  "prettier": {
    "trailingComma": "es5",
    "tabWidth": 2,
    "semi": false,
    "singleQuote": true,
    "useTabs": false
  },
  "dependencies": {
    "react": "^17.0.0",
    "react-dom": "^17.0.0"
  },
  "devDependencies": {
    "@types/react": "^17.0.0",
    "@types/react-dom": "^17.0.0",
    "@vitejs/plugin-react-refresh": "^1.3.1",
    "sass": "^1.35.2",
    "typescript": "^4.3.2",
    "vite": "^2.4.2"
  }
}
```

# src/App.tsx

Change the content of this file to:

```
import React from 'react'  
  
export function App() {  
  return <div>Hello, World</div>  
}
```

# **Ensure the latest version of vite and typescript**

```
npm install --save-dev vite typescript
```

# Turn on TypeScript checking in our terminal

```
npm install --save-dev vite-plugin-checker
```

Open `vite.config.ts` and add the following to the top:

```
import checker from 'vite-plugin-checker'
```

To the `plugins` section add:

```
checker({ typescript: true })
```

# src/vite.config.js

Your should look similar to:

```
import { defineConfig } from 'vite'
import reactRefresh from '@vitejs/plugin-react-refresh'
import checker from 'vite-plugin-checker'

// https://vitejs.dev/config/
export default defineConfig({
  plugins: [reactRefresh(), checker({ typescript: true })],
})
```

# Turn on additional code checks (eslint)

```
npm install --save-dev "prettier"  
npm install --save-dev "vite-plugin-linter"  
npm install --save-dev "@typescript-eslint/parser"  
npm install --save-dev "eslint-plugin-react-hooks"  
npm install --save-dev "eslint"  
npm install --save-dev "eslint-config-prettier"  
npm install --save-dev "eslint-plugin-prettier@3.4.1"  
npm install --save-dev "eslint-plugin-react"  
npm install --save-dev "eslint-config-react-app"  
npm install --save-dev "eslint-plugin-jsx-a11y"
```

# Turn on additional code checks (eslint)

## Update vite.config.ts

```
import { defineConfig } from 'vite'
import reactRefresh from '@vitejs/plugin-react-refresh'
import checker from 'vite-plugin-checker'
import { ESLinter, linterPlugin, TypeScriptLinter } from 'vite-plugin-linter'

// https://vitejs.dev/config/
export default defineConfig(configEnv => ({
  plugins: [
    reactRefresh(),
    checker({ typescript: true }),
    linterPlugin({
      disableForBuild: true,
      include: ['./src/**/*.ts', './src/**/*.tsx'],
      linters: [
        new ESLinter({
          configEnv: configEnv,
          serveOptions: { cache: false, formatter: 'visualstudio' },
        }),
        new TypeScriptLinter(),
      ],
    }),
  ],
}))
```

# Turn on additional code checks (eslint)

Add an `.eslintrc.js` to the root of the project

```
module.exports = {
  root: true,
  parser: '@typescript-eslint/parser',
  parserOptions: {
    ecmaVersion: 2020,
    sourceType: 'module',
    ecmaFeatures: {
      jsx: true,
    },
  },
  settings: {
    react: {
      version: 'detect',
    },
  },
  env: {
    browser: true,
    amd: true,
    node: true,
  },
  extends: [
    'eslint:recommended',
    'plugin:react/recommended',
    'plugin:react-hooks/recommended',
    // Uncomment this next line if you want to check your code for accessibility issues!
    // 'plugin:jsx-a11y/recommended',
    'prettier', // Make sure this is always the last element in the array.
  ],
  plugins: [],
  rules: {
    'react/react-in-jsx-scope': 'off',
    'jsx-a11y/accessible-emoji': 'off',
    'react/prop-types': 'off',
    '@typescript-eslint/explicit-function-return-type': 'off',
    'jsx-a11y/anchor-is-valid': 'off',
  },
}
```

# **Add a file to configure paths and files not to check**

Add an `.eslintignore` to the root of the project

```
dist/
```

```
node_modules/
```

# Add a file to setup Netlify to work with React Router

When we get to React Router we will need to configure Netlify to serve our projects `index.html` for any URL the user refreshes their browser on.

To this:

- Create a directory named `public` at the root of your project
- **INSIDE** that directory, place a file named `_redirects` with the following content:

```
/* /index.html 200
```

# Create a github repository for the project

```
hub create
```

```
git push origin HEAD
```

# Congratulations

You have just created your own initial starter project for the React projects we'll be doing in this course.

You can return to this repository and make any changes you'd like to become your new default template.

Perhaps you'll develop a color palette you like to use in each project. Come back to this template and add that CSS. All **new** projects you create will have that CSS.

# More templates!

When we start with React Router, and other third party libraries, we will use the same template. At that time, you can add those libraries to your default template if you wish.

