

Setting up a TypeScript project with Vite

Summary: in this tutorial, you will learn how to set up a web project that uses TypeScript and Vite to compile TypeScript to JavaScript.

[Vite](#) is a front-end build tool that offers a faster and leaner development experience for modern web projects. The name “Vite” means “fast” in French, which reflects its goal.

Setting up a TypeScript project

We'll show you step-by-step how to set up a TypeScript project using Vite:

Step 1. Install Node.js

Open the command prompt on Windows or the Terminal on macOS and check if Node.js is installed on your computer:

```
node -v
```

If you get an error, it means Node.js is not available. You need to [download and install Node.js](#) from the [official website](#).

Step 2. Create a Vite project

Open the terminal and use the following command to create a new Vite project:

```
npm create vite@latest
```

If Vite is not installed on your computer, it'll prompt you to install it:

```
Need to install the following packages:  
create-vite@5.5.2  
Ok to proceed? (y) y
```

Enter `y` to proceed with the installation. Note that you likely see a higher version above.

Once you install the `create-vite` package, you'll be prompted to name your project, choose a framework, and select a variant:

```
✓ Project name: ... myapp
✓ Select a framework: » Vanilla
✓ Select a variant: » TypeScript
```

In this example, we use `myapp` as the name of the project, `Vanilla` as a framework, and `TypeScript` as a variant.

The command will scaffold the project in a directory whose name is the same as the project name:

```
Scaffolding project in D:\myapp...

Done. Now run:

  cd myapp
  npm install
  npm run dev
```

Step 3. Navigate to the project directory

Navigate to the project directory:

```
cd myapp
```

Step 4. Install dependencies

Install the necessary dependencies by running the `npm install` command:

```
npm install
```

Step 5. Start development server

Run the following command to start the development server:

```
npm run dev
```

It'll return the following output:

```
> myapp@0.0.0 dev
> vite

VITE v5.4.1 ready in 275 ms

→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

Open the `http://localhost:5173/` on the web browser, you see the app.

To stop the development server, you can press the `Ctrl+C` .

Examine the project structure

Here's the project structure:

```
├─ index.html
├─ node_modules
├─ package-lock.json
├─ package.json
├─ public
│   └─ vite.svg
├─ src
│   └─ counter.ts
│   └─ main.ts
│   └─ style.css
│   └─ typescript.svg
│   └─ vite-env.d.ts
└─ tsconfig.json
```

Here are the files and directories in the project (we'll focus on the main ones):

index.html

This is the entry point for the app. The `index.html` include the `src/main.ts` file which is the main TypeScript file.

During the development, Vite will compile this `src/main.ts` file into a JavaScript bundle and inject it into the `index.html` file.

src/main.ts

This is the entry point for your application. In this `main.ts` file.

style.css

This file stores the CSS code of the app. You can use it in your app using the import statement:

```
import './style.css';
```

Note that you can import a CSS file like a JavaScript file because of the way Vite handles asset bundling and processing. So you don't have to manually link stylesheets in HTML.

package.json

This `package.json` file is used to manage dependencies, scripts, and configurations.

node_modules/

This directory contains all the installed npm packages and their dependencies. When you execute the `npm install` command, it'll install all packages listed on the `package.json` file in this folder.

tsconfig.json

This is the TypeScript configuration file that includes compiler options, file inclusions/exclusions, and other settings. It defines how TypeScript should behave.

Customize the project

Step 1. Delete the `counter.ts` file from the `src` directory. This step is optional.

Step 2. Replace all the code in the `src/main.ts` file with the following:

```
import './style.css';

const app = document.querySelector<HTMLDivElement>('#app');
app!.innerHTML = '<h1>Hello, TypeScript!</h1>';
```

How it works.

First, import `style.css` into the `main.ts` file:

```
import './style.css';
```

Second, select the `app` element and change its `innerHTML` to an HTML fragment:

```
const app = document.querySelector<HTMLDivElement>('#app');
app!.innerHTML = '<h1>Hello, TypeScript!</h1>';
```

Third, run the development server, you'll see the text `Hello, TypeScript!`

Build the project for production

Run the following command to build the project for production:

```
npm run build
```

It'll create an optimized production build in the `dist` directory with the following structure:

```
├─ assets
│   ├─ index-Cz4zGhbH.css
│   └─ index-NXap3Nzt.js
├─ index.html
└─ vite.svg
```

To preview the production build locally, you can run the following command:

```
npm run preview
```

This command will start a local server to view the product build at <http://localhost:4173/>

```
> myapp@0.0.0 preview
> vite preview

→ Local:   http://localhost:4173/
→ Network: use --host to expose
→ press h + enter to show help
```

Now, you can start developing your TypeScript application using Vite. Happy programming!