Intro to Vite

In this lesson Evan You will cover what is Vite, and why it's the next generation of online tooling. We'll also look at some of the features you get out of the box.

Why Vite?

Vite is the french word for fast, pronounced Veet. It's a build tool that comes with a dev server, and it bundles your code for production. It's similar to Vue CLI but leaner and faster. One problem you might run into as a developer is the amount of time it take for your application to load up when you make changes. Vite makes the feedback loop speed during development super fast.

How does it do this?

Vite authors our code as native ES Modules, which modern browsers support as a way to load your JavaScript. Plus, if you have large dependencies, Vite will pre-bundle them to reduce the number of requests that the browser must do to the web server.

To get started with Vite

First you need Node.js and NPM installed on your system. To start a project with Vite you simply type:
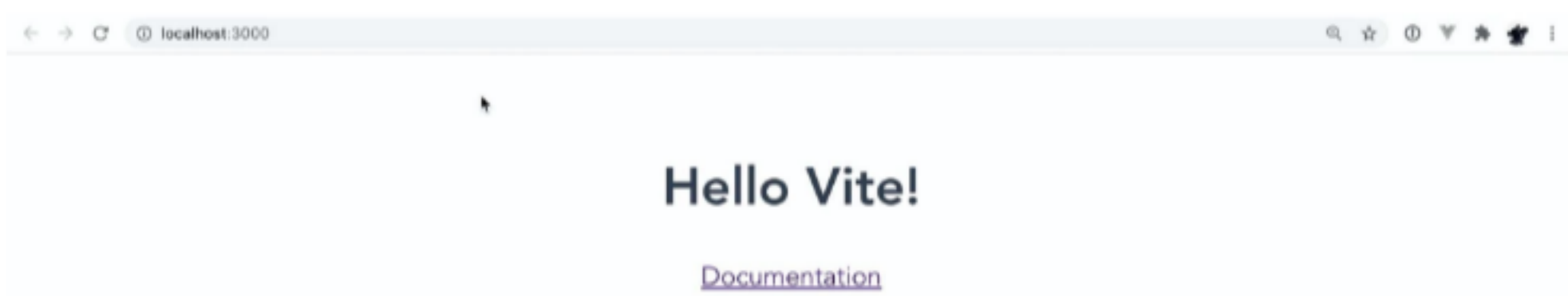
```
$ npm init @vitejs/app
```

This will ask for a project name, select a framework & variant. To just get a basic app running, you can select *vanilla*. This will create the basic files you need to start building a webpage, with an index.html, main.js (for our JavaScript code), package.json (with Vite as the only dependency), and style.css. This simple webpage could be run on any web server.

To run it with Vite, however, you'll want to go into the directory, install packages, and run the script to startup the dev server with the following commands:

```
$ cd /project
$ npm install
$ npm run dev
```

Now we have a local server running. It'll look something like this:

There's not much different from a http server, but if we change things around such as renaming files, Vite reloads everything automatically. We can even include TypeScript files in our html like so:

index.html

```
...
<script type="module" src="/main.ts"></script>
...
```

Vite will transpiled TypeScript automatically. Vite doesn't do any type-checking for you, which you should know. Vite uses es-build to transpile the files, which is about 30 times faster for transpiling than TypeScript itself. If you care about type-checking, you'll need to import it using:

main.ts

```
import type { } './'
```

However, if you're using a modern editor, you probably get type-checking already in your IDE. Vite can leave that for your IDE, to get the speed boost.

Vite has "hot updating," which you can easily see if you make any changes to the `style.css` file which is imported in our `main.js` file.

Handling NPM Dependencies

Vite also handles NPM dependencies for you. We can see this by importing a library like so:

main.ts

```
import { debounce } from 'lodash-es'
```

All then we have to do is install the dependency:

```
$ npm install lodash-es
```

Then we restart our dev server, and if we look into the browser, we can see that the new dependency has been loaded. All of this happened without reloading the browser.

CSS & Debugging

Learn how Vite makes it easy to use CSS plugins, and how to debug

Create a free account to unlock it.

🔓 Unlock Content

Working with Assets

Vite makes it easy to work with both imported assets and public assets

Unlock this lesson by subscribing to a plan.

🔓 Unlock Content

Getting Vite working with Vue

Moving to Vite

There are a few things you should keep in mind when moving over an existing

Unlock this lesson by subscribing to a plan.

🔓 Unlock Content