

Vue CLI

[NEXT: COMPONENTS AND THE VUE INSTANCE](#)

An insignificant but time-consuming phase of our project's lifecycle is **setting up and tooling**. There's a lot of terms to understand and digest -- webpack, SystemJS, bundling, linting, uglifying, etc. A CLI tool helps you automate the process of setting up your project environment from development, to testing and even as far as production.

Therefore, you don't have to care about how to start anymore, just run a few commands and you are good to go.

As a side note, the CLI tool is not required for you to build a Vue application. In the previous [Hello World](#) example, we had a typical Codepen project running without any CLI help. If for any reason, you feel like the CLI tool is overkill (which I don't think it is), then simply use the CDN or download the Vue scripts and include them manually in your project.

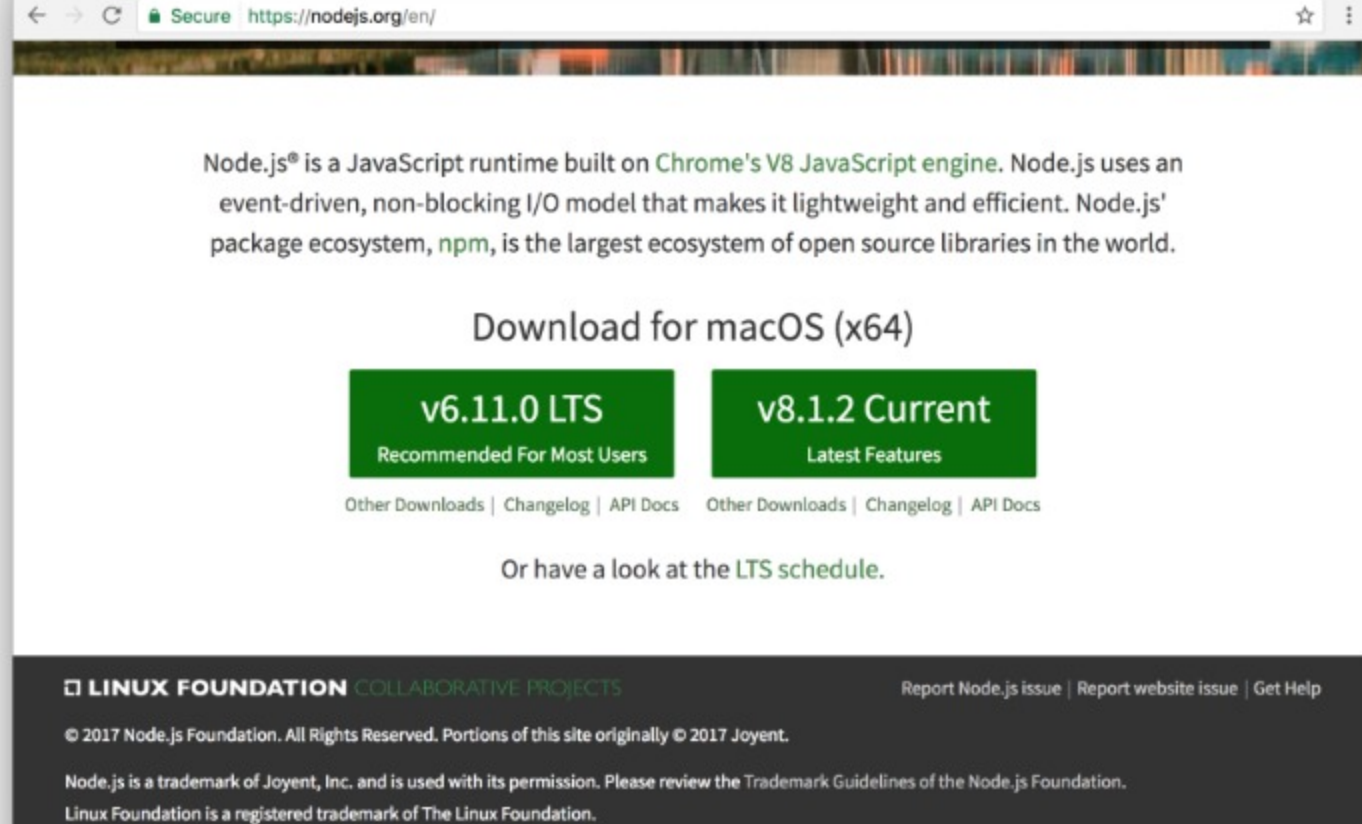
Table of Contents

- 1 Installation
- 2 Vue CLI Templates
- 3 Creating a Project

As a matter of fact, there is a Vue CLI template that works that way. We will discuss the templates in upcoming lessons.

Installation

The CLI tool requires [Node](#) and [npm](#). Node in a nutshell is JavaScript on the server. It comes with a package manager known as Node Package Manager (npm) after installation. The main reason we'll be needing Node is for npm.



npm is the largest ecosystem of open source libraries in the world. It allows you to install and manage JavaScript libraries and tools in a very efficient manner. We will use npm to install the Vue CLI tool:

```
npm install -g vue-cli
```

The `-g` flag is known as the global flag and tells npm that this installation is not specific to a given project but to our entire machine. This makes sense because we will be using the CLI tool for different types of project from anywhere in our computer.

Vue CLI Templates

Vue CLI is flexible enough that it allows you to use whatever template suits you as the boilerplate while generating a new project using the CLI's `new` command. You can [create your template](#) or use [existing official templates](#).

The existing templates will always be able to provide everything you need for your project. Only extreme cases will demand that you create your template. Let's have a look at the existing official templates as seen on the [Vue CLI Repo](#):

1. **webpack**: A full-featured [webpack](#) + vue-loader setup with hot reload, linting, testing & CSS extraction.
2. **webpack-simple**: A simple webpack + vue-loader setup for quick prototyping.
3. **browserify**: A full-featured [Browserify](#) + vueify setup with hot-reload, linting & unit testing.
4. **browserify-simple**: A simple Browserify + vueify setup for quick prototyping.
5. **simple**: The simplest possible Vue setup in a single HTML file.

Creating a Project

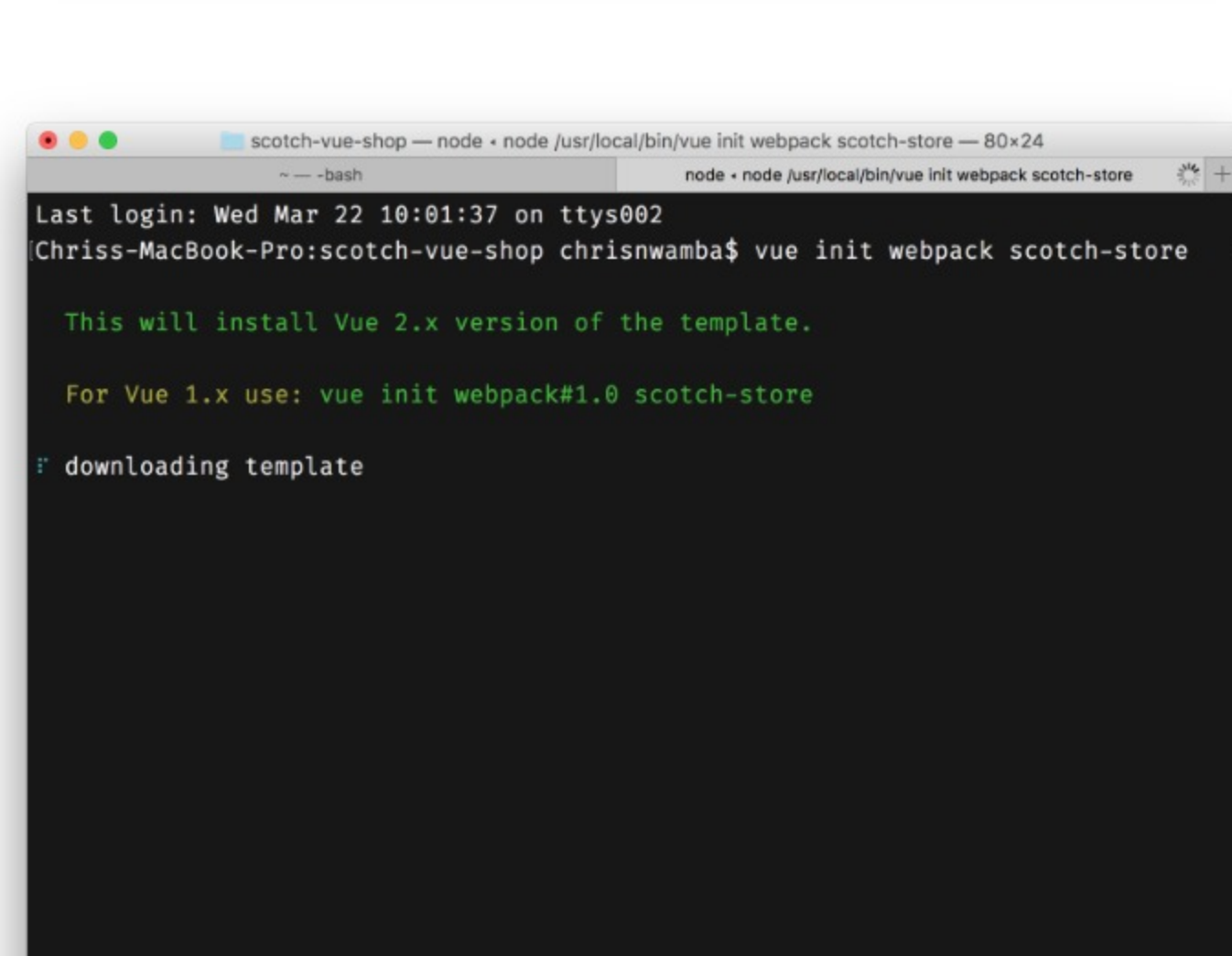
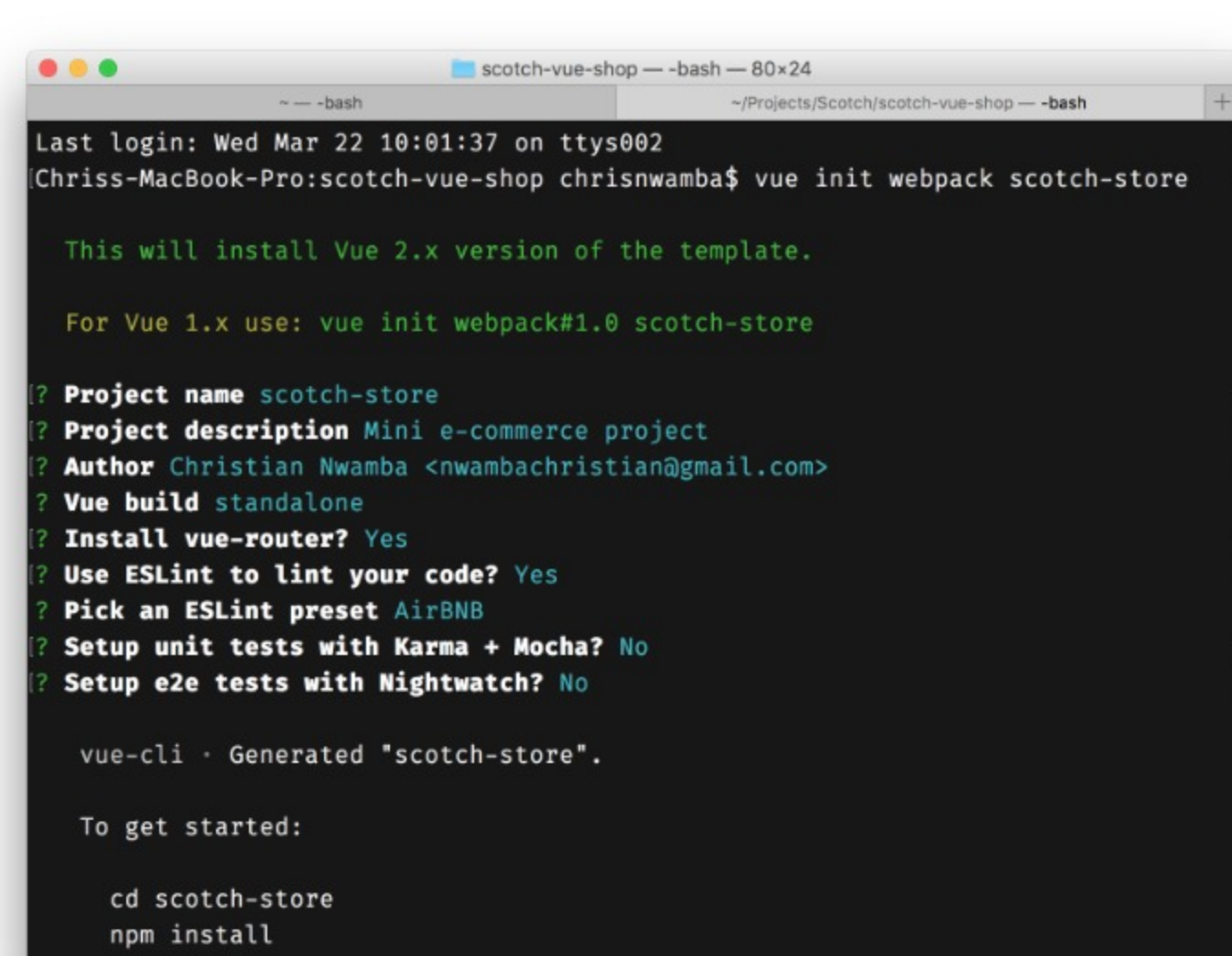
Now that you know what the CLI tool is, we have it installed, and we know the different templates, let's see how we can create a new project using any of the pre-made templates.

We are building a *real* application so we would need all the features the CLI can offer. This means we are not going to go for `simple` or the `-simple` template.

webpack is a more evolved and popular tool, so let's ditch Browserify for the cooler kid:

```
# Run the following anywhere
# on your machine that you want
# your project to be located
vue init webpack scotch-store
```

Answer the brief questions using the answers shown in the image below:

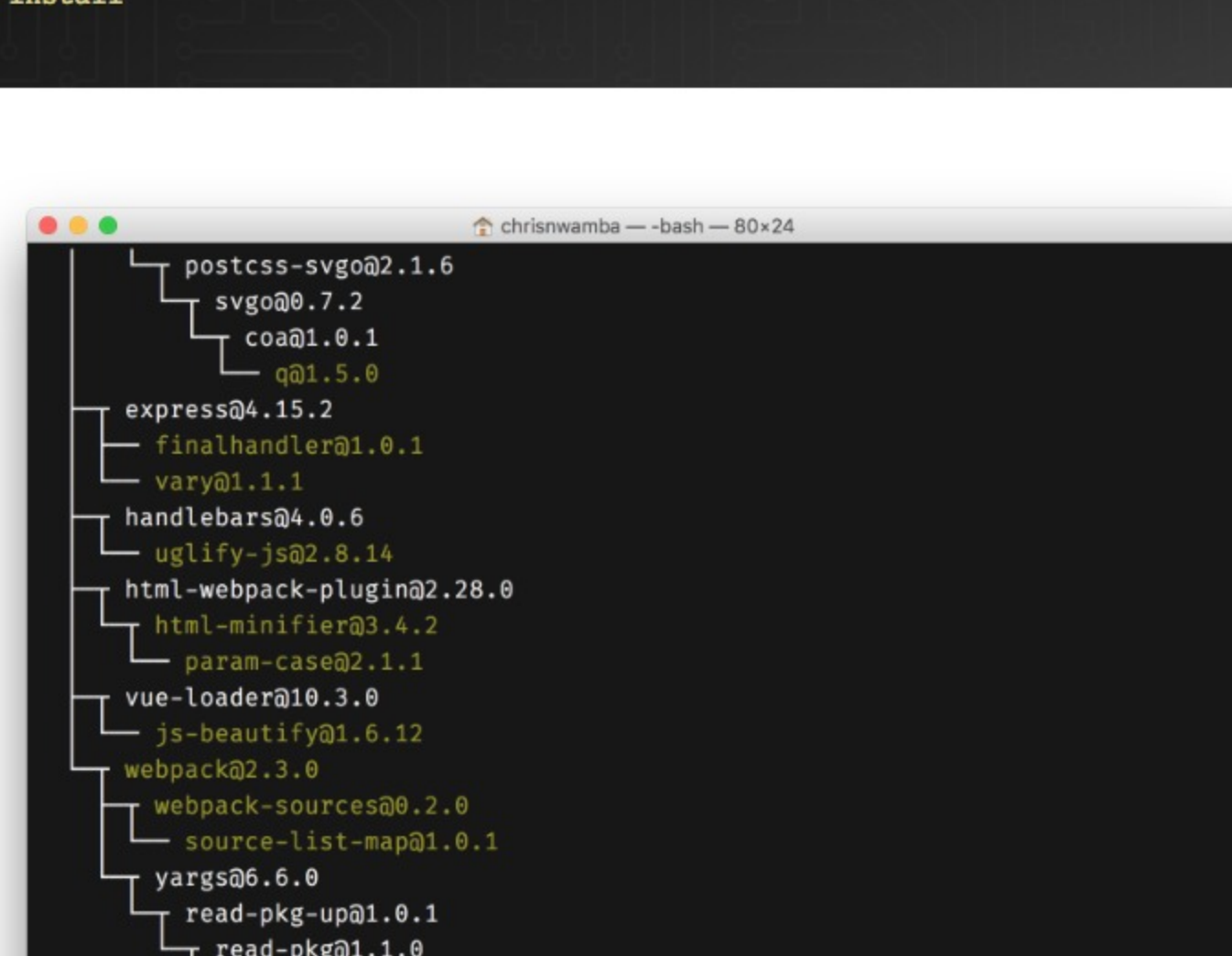


The syntax is `vue init [template] [project-name]`.

You need to install the dependencies generated in the `package.json`. To do that, enter the `scotch-store` folder and run the install command:

```
# Enter the scotch-store folder
cd scotch-store

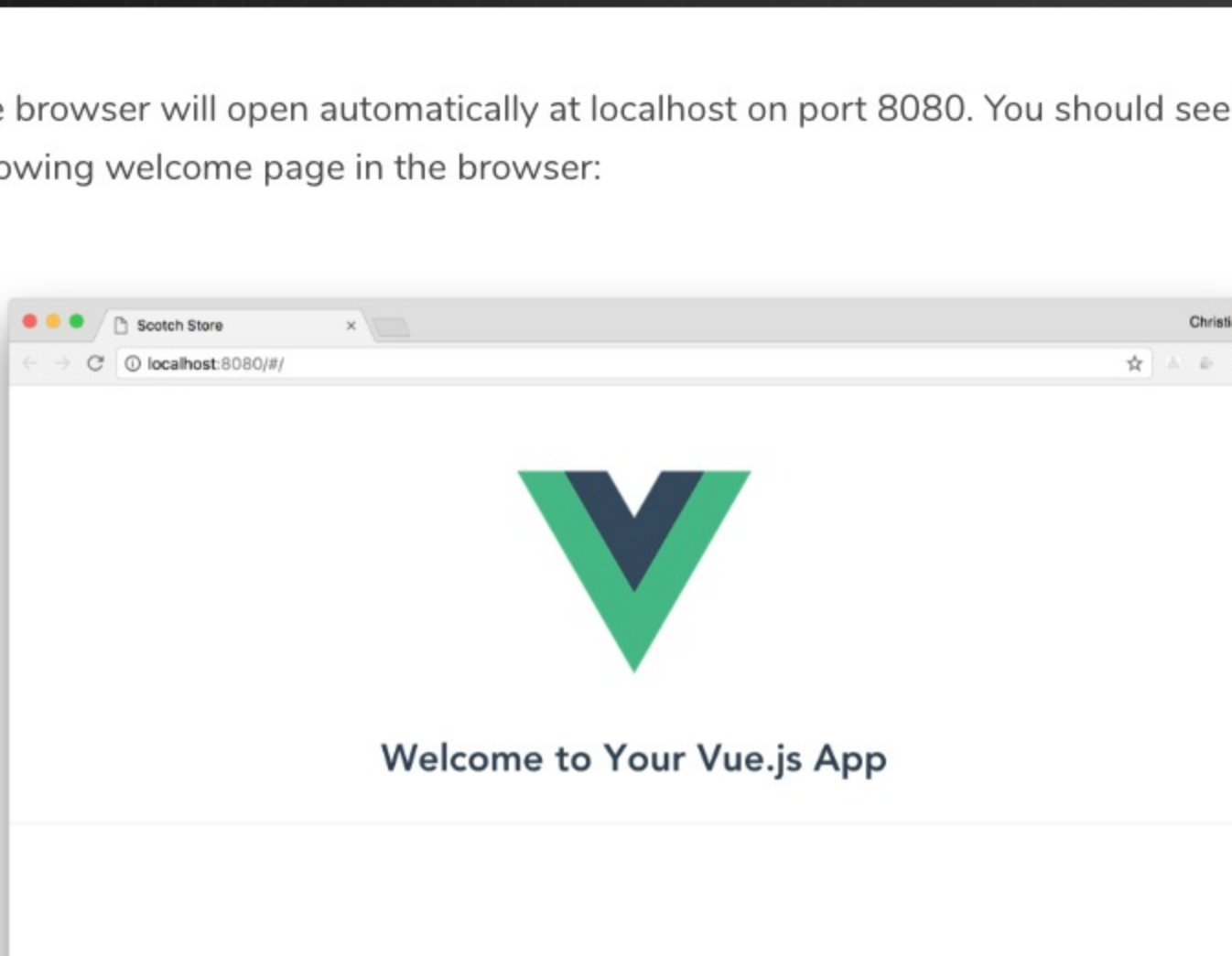
# Run install command
npm install
```



You can run the app now to see what was generated by running the following in the same terminal you used to install the dependencies:

```
# Run the Vue app
npm run dev
```

The browser will open automatically at localhost on port 8080. You should see the following welcome page in the browser:



Getting Started

- 1 Introduction [Free](#)
- 2 Tools [Free](#)
- 3 Our Task [Free](#)
- 4 Hello World [Free](#)

Vue Basics

- 5 **Vue CLI** [Free](#)
- 6 Components and the Vue Instance [Free](#)
- 7 Template Syntax [Free](#)
- 8 Conditional & List Rendering [Free](#)
- 9 Styles & Classes [Free](#)

Routing

- 10 Single Page Apps [Free](#)
- 11 Creating Routes [Free](#)
- 12 Route Outlets & Links [Free](#)
- 13 Nested & Dynamic Routes [Free](#)

Forms

- 14 Two Way Binding (Reactivity) [Free](#)
- 15 Form Validation [Free](#)
- 16 Handling Form Submissions [Free](#)

API Backend

- 17 Prelude [Free](#)
- 18 Setup [Free](#)
- 19 Provisioning a MongoDB Database [Free](#)
- 20 Enabling CORS [Free](#)
- 21 Schemas & Models [Free](#)
- 22 Routes and Controllers [Free](#)
- 23 Testing with POSTman [Free](#)

Vuex

- 24 The Current Problem [Free](#)
- 25 State Management [Free](#)
- 26 Getters [Free](#)
- 27 Mutations [Free](#)
- 28 Actions [Free](#)

Using Store In Components

- 29 State and Actions on Components [Free](#)
- 30 LAB: Product List [Free](#)
- 31 LAB: Product Details [Free](#)
- 32 LAB: Admin Features [Free](#)
- 33 LAB: Spinner, Cart & Store Subscription [Free](#)