Web Application Development

Produced by

David Drohan (ddrohan@wit.ie)

Department of Computing & Mathematics
Waterford Institute of Technology

http://www.wit.ie

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Vue.js RELATED

DONATIONVUE & WEBPACK

DonationVue & Webpack

A JAVASCRIPT MODULE BUNDLER

What is Webpack?

“At its core, **webpack** is a *static module bundler* for modern JavaScript applications. When webpack processes your application, it internally builds a *dependency graph* which maps every module your project needs and generates one or more *bundles*.”

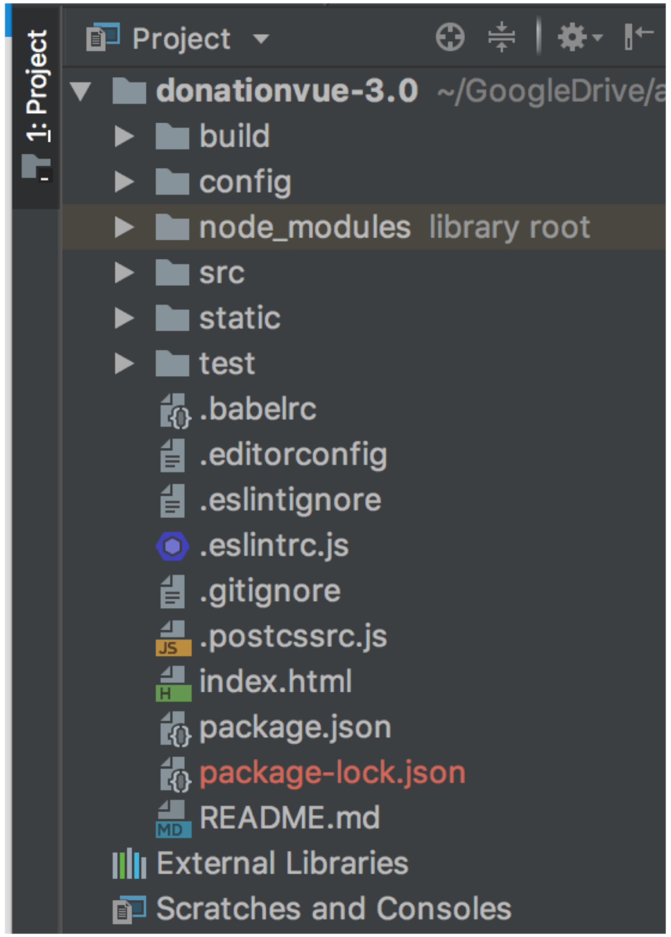
Since version 4.0.0, **webpack does not require a configuration file** to bundle your project, nevertheless it is incredibly configurable to better fit your needs.

Project Structure & Webpack

In WebStorm specifically, this is what you get when you choose to build your Vue app with **Webpack**.

In the project root folder you can find files and folders. Let's examine the most important ones. The *package.json* file contains all the dependencies of your project.

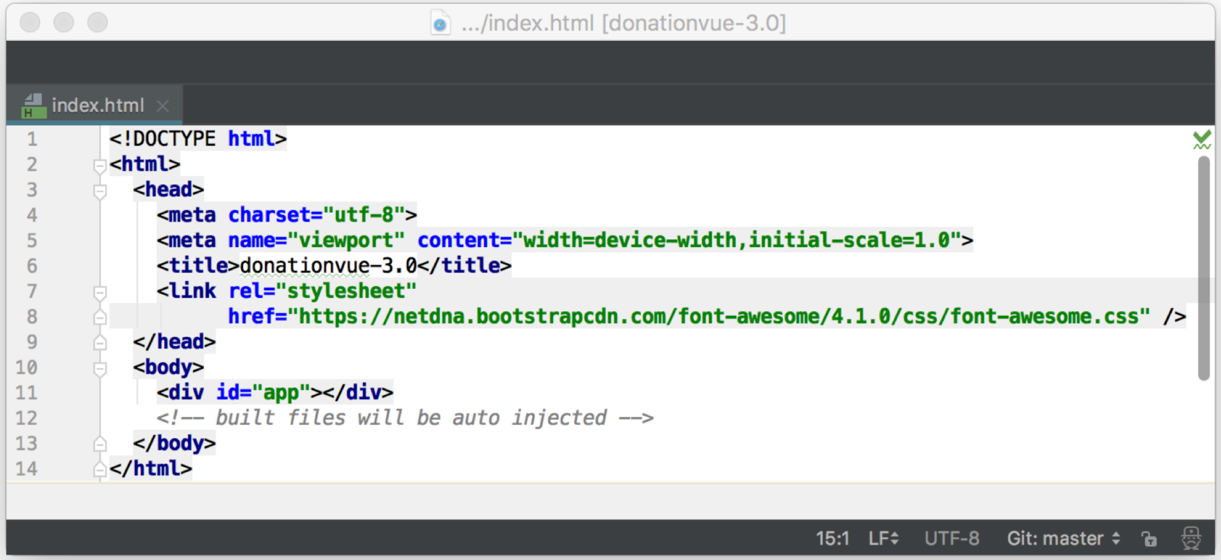
By using the command **npm install**, we make sure that the dependencies listed in *package.json* are installed into the ***node_modules*** folder of the project.



```
Project
▼ donationvue-3.0 ~/GoogleDrive/a
  ▶ build
  ▶ config
  ▶ node_modules library root
  ▶ src
  ▶ static
  ▶ test
    .babelrc
    .editorconfig
    .eslintignore
    .eslintrc.js
    .gitignore
    .postcssrc.js
    index.html
    package.json
    package-lock.json
    README.md
  External Libraries
  Scratches and Consoles
```

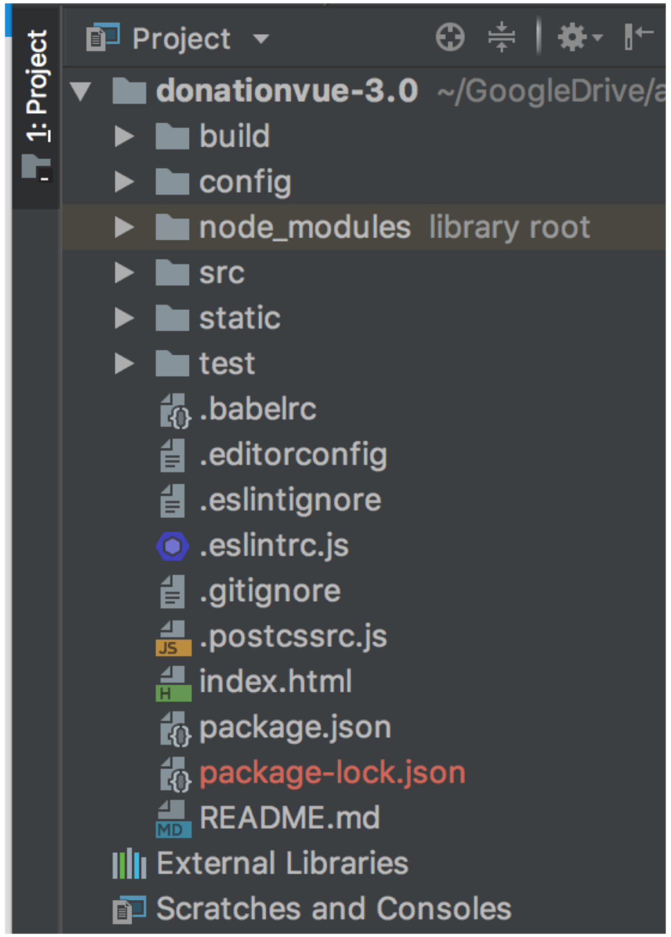
Project Structure & Webpack

The file *index.html* contains the following:


.../index.html [donationvue-3.0]
index.html
15:1 LF÷ UTF-8 Git: master ÷

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <title>donationvue-3.0</title>
    <link rel="stylesheet"
          href="https://netdna.bootstrapcdn.com/font-awesome/4.1.0/css/font-awesome.css" />
  </head>
  <body>
    <div id="app"></div>
    <!-- built files will be auto injected -->
  </body>
</html>
```

This file is the starting point of your application. Note that within the *body* section a **<div>** element is available which has the *id* property set to string *app*. This element is used as a placeholder for the output which is generated by Vue.js.

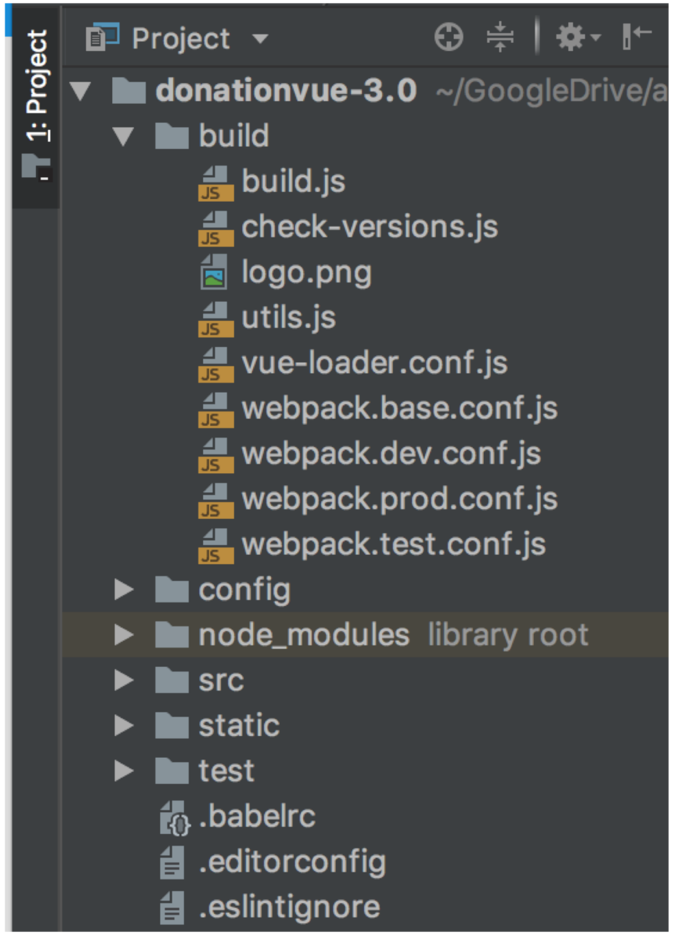


```
donationvue-3.0 ~/GoogleDrive/a
  build
  config
  node_modules library root
  src
  static
  test
  .babelrc
  .editorconfig
  .eslintignore
  .eslintrc.js
  .gitignore
  .postcssrc.js
  index.html
  package.json
  package-lock.json
  README.md
External Libraries
Scratches and Consoles
```

Project Structure & Webpack

The ***build*** folder contains all the files relating to how the project is built, and config settings for both development and production ‘builds’ and also testing.

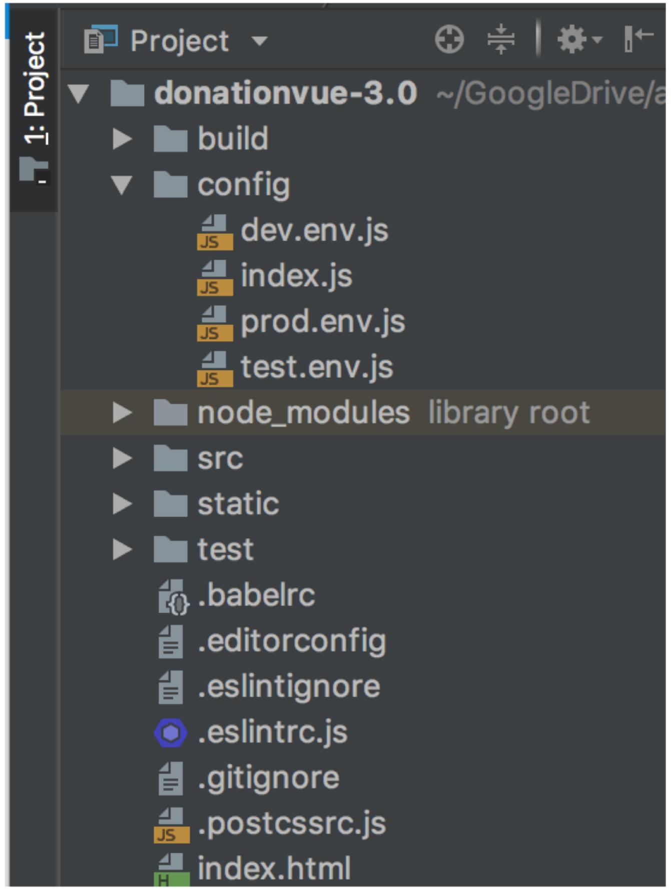
You generally don’t need to modify/changes these files, but if you do, make sure you know what you’re doing!



Project Structure & Webpack

The ***config*** folder contains all the files relating to actual configuration settings for the different builds in the ***build*** folder.

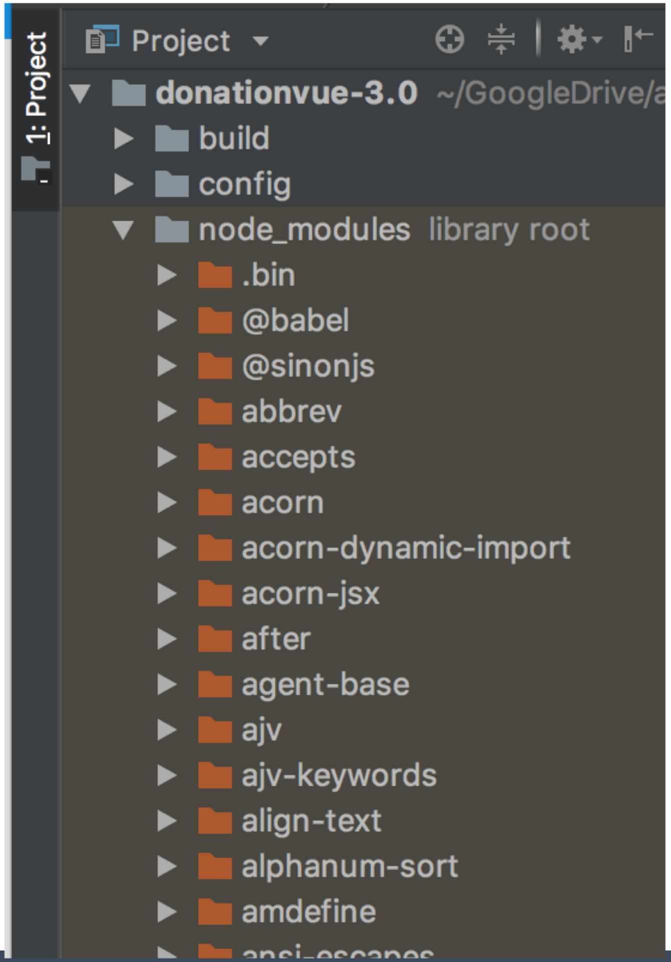
Again, you generally don't need to modify/changes these files, but you can if you want/need to do so.



```
1: Project
Project
▼ donationvue-3.0  ~/GoogleDrive/a
  ▶ build
  ▼ config
      dev.env.js
      index.js
      prod.env.js
      test.env.js
  ▶ node_modules  library root
  ▶ src
  ▶ static
  ▶ test
    .babelrc
    .editorconfig
    .eslintignore
    .eslintrc.js
    .gitignore
    .postcssrc.js
    index.html
```

Project Structure & Webpack

The ***node_modules*** folder - Fairly self explanatory ☺

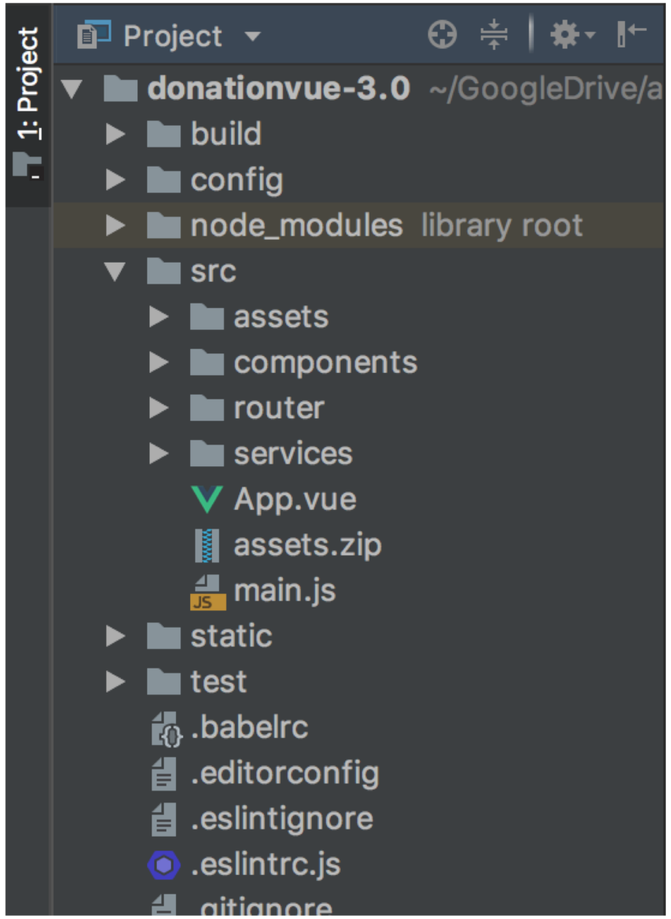


```
1: Project
Project
▼ donationvue-3.0 ~/GoogleDrive/a
  ▶ build
  ▶ config
  ▼ node_modules library root
    ▶ .bin
    ▶ @babel
    ▶ @sinonjs
    ▶ abbrev
    ▶ accepts
    ▶ acorn
    ▶ acorn-dynamic-import
    ▶ acorn-jsx
    ▶ after
    ▶ agent-base
    ▶ ajv
    ▶ ajv-keywords
    ▶ align-text
    ▶ alphanum-sort
    ▶ amdefine
    ▶ ansi-escapes
```

Project Structure & Webpack

The ***src*** folder is where you'll spend most of your development time. This is where all your source code goes, including, your assets, javascript, components and any other source files relating to your project.

Let's take a closer look at file *main.js* in folder ***src***. That's the place where the Vue application is initialized:

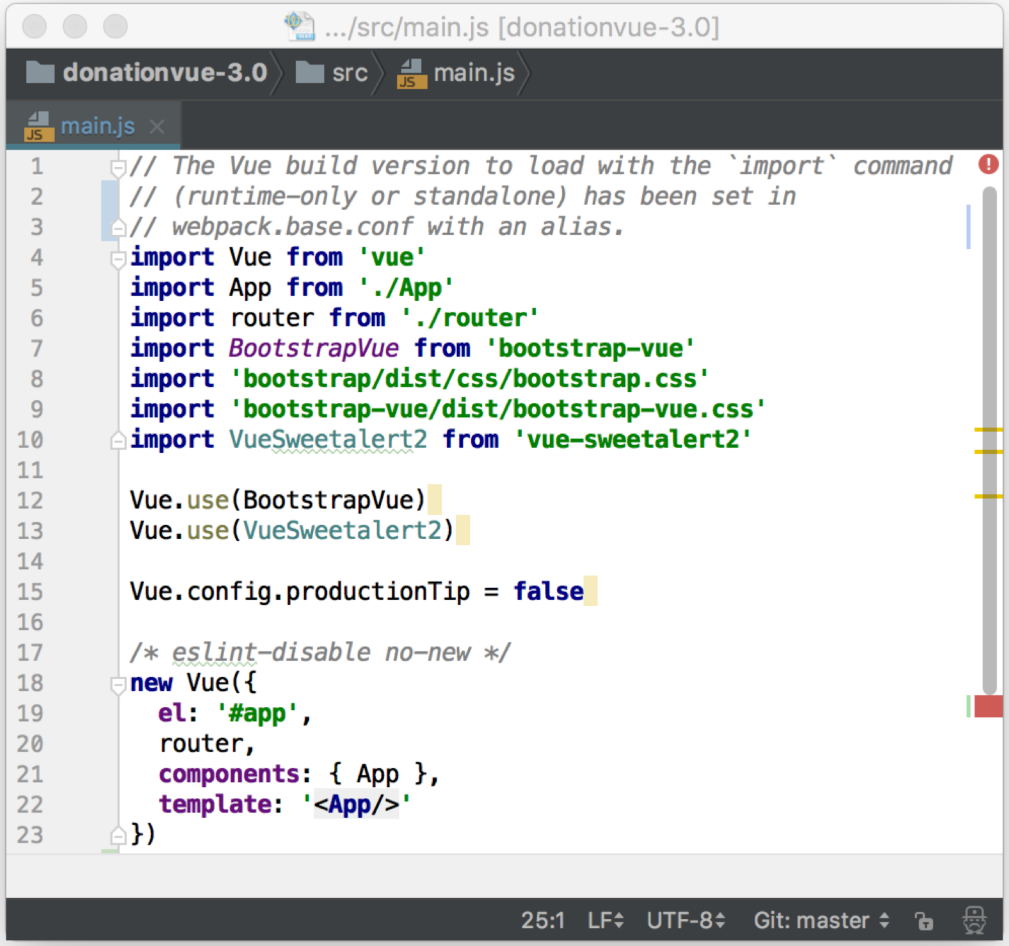


```
1: Project
Project
▼ donationvue-3.0  ~/GoogleDrive/a
    ▶ build
    ▶ config
    ▶ node_modules  library root
    ▼ src
        ▶ assets
        ▶ components
        ▶ router
        ▶ services
          App.vue
          assets.zip
          main.js
    ▶ static
    ▶ test
      .babelrc
      .editorconfig
      .eslintignore
      .eslintrc.js
      .gitignore
```

Project Structure & Webpack

On top of the file you can find a number of import statements and by using the *new* keyword a new instance of the main framework class *Vue* is created. The constructor takes an object as a parameter which contains three properties:

- **el**: By assigning the string ***#app*** to this property we're defining that the output of the Vue application should be rendered to the **<div id="app"></div>** element in *index.html*.
- **template**: The template contains the HTML code which is used to generate the output of the Vue.js application.
- **components**: List of Vue.js components which are used in the template.

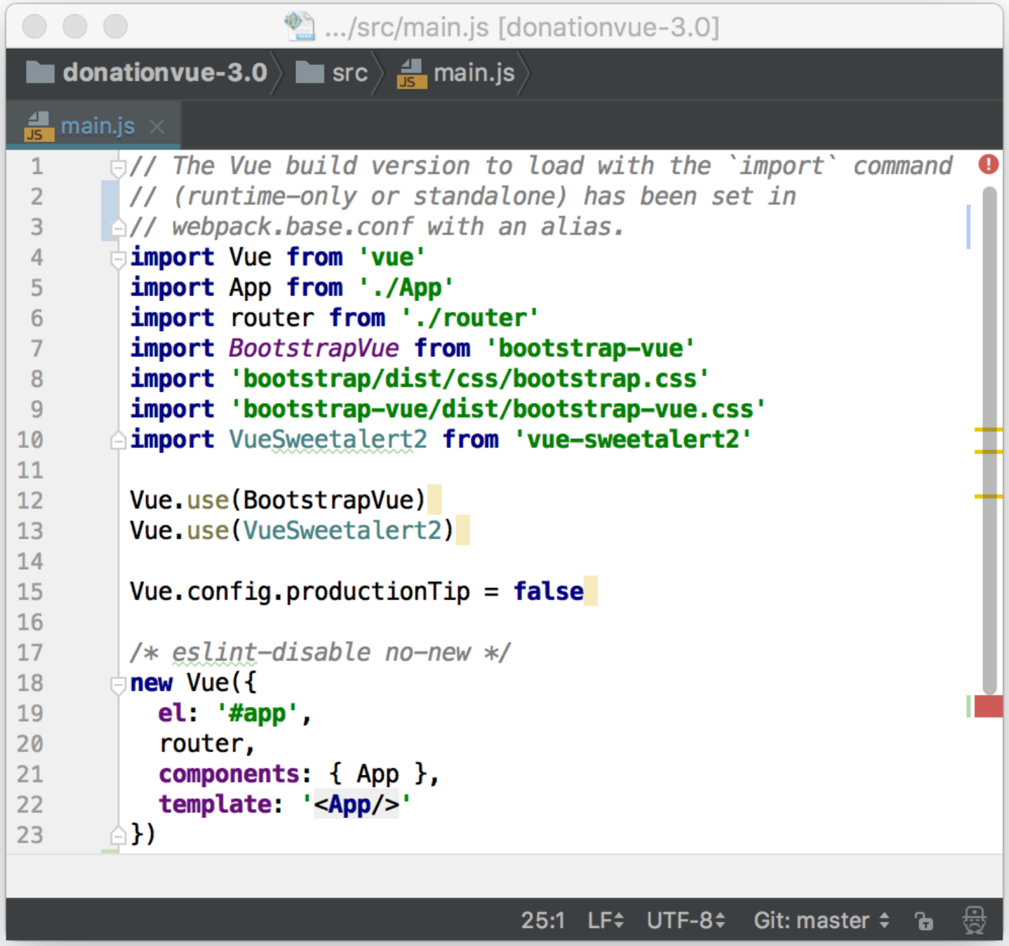


```
// The Vue build version to load with the `import` command
// (runtime-only or standalone) has been set in
// webpack.base.conf with an alias.
import Vue from 'vue'
import App from './App'
import router from './router'
import BootstrapVue from 'bootstrap-vue'
import 'bootstrap/dist/css/bootstrap.css'
import 'bootstrap-vue/dist/bootstrap-vue.css'
import VueSweetalert2 from 'vue-sweetalert2'

Vue.use(BootstrapVue)
Vue.use(VueSweetalert2)

Vue.config.productionTip = false

/* eslint-disable no-new */
new Vue({
  el: '#app',
  router,
  components: { App },
  template: '<App/>'
})
```

Project Structure & Webpack

The template only consists of one element: **<App/>.** Of course this is not a standard HTML element. This is the element which is assigned to App component. In order to be able to use <App/> in the template the App component is also listed in the object which is assigned to the *components* property.

So let's see what's inside the App component implementation in file ***App.vue***:



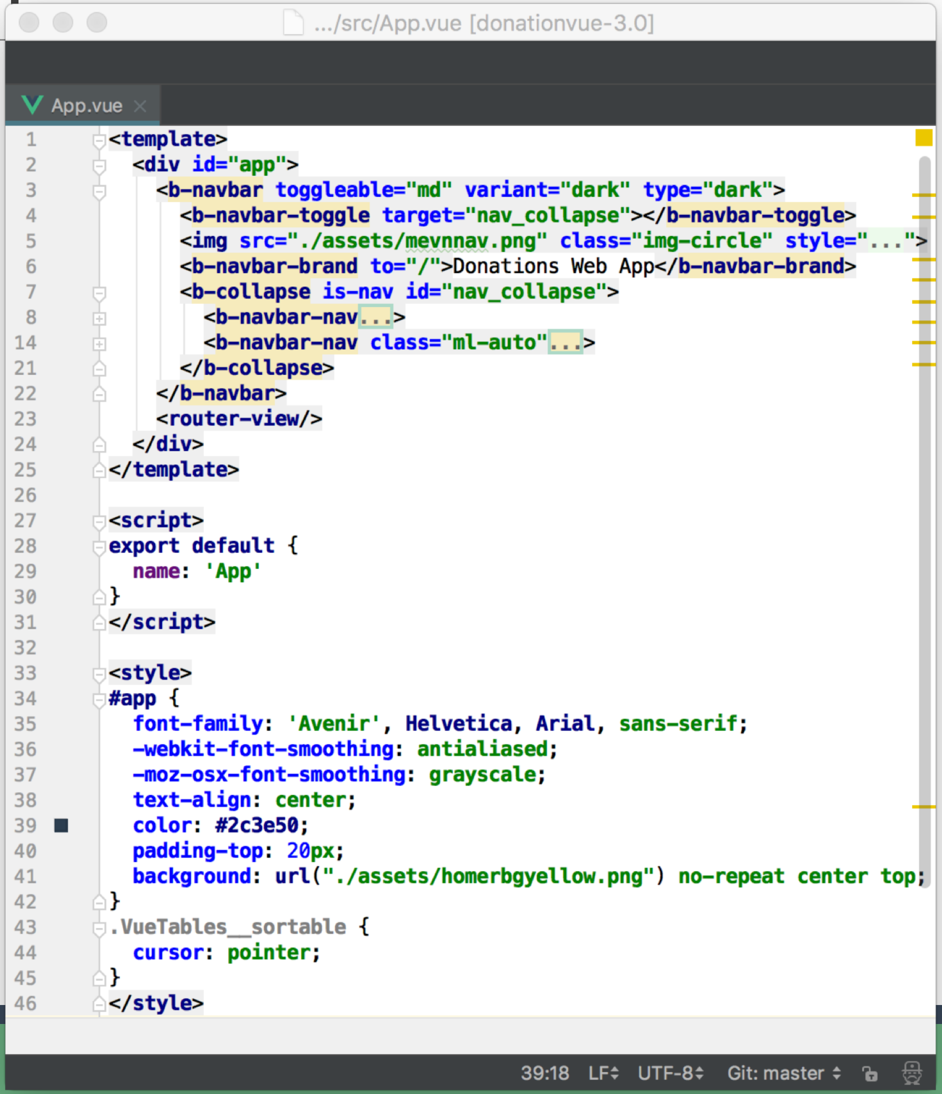
```
// The Vue build version to load with the `import` command
// (runtime-only or standalone) has been set in
// webpack.base.conf with an alias.
import Vue from 'vue'
import App from './App'
import router from './router'
import BootstrapVue from 'bootstrap-vue'
import 'bootstrap/dist/css/bootstrap.css'
import 'bootstrap-vue/dist/bootstrap-vue.css'
import VueSweetalert2 from 'vue-sweetalert2'

Vue.use(BootstrapVue)
Vue.use(VueSweetalert2)

Vue.config.productionTip = false

/* eslint-disable no-new */
new Vue({
  el: '#app',
  router,
  components: { App },
  template: '<App/>'
})
```

Project Structure & Webpack

The <template> section is essentially specifying our home page, which includes the <navbar> - this is injected into our *index.html* on loadup via *main.js*.

The <script> section is making a default export of an object declaring the component named *app*.

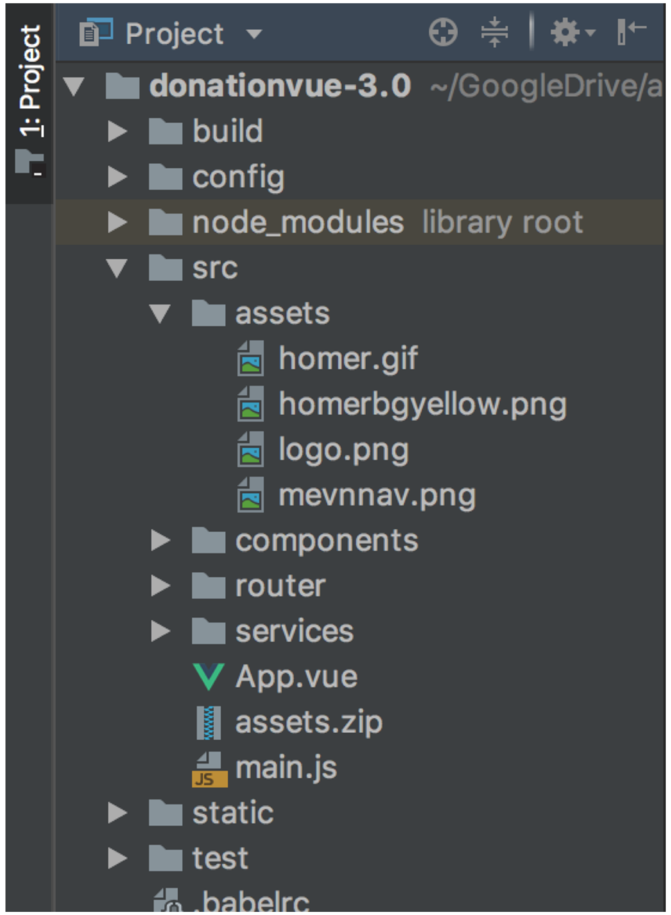
The <style> section specifies global styling (not ‘scoped’) for our app, including our background.



```
<template>
  <div id="app">
    <b-navbar toggleable="md" variant="dark" type="dark">
      <b-navbar-toggle target="nav_collapse"></b-navbar-toggle>
      <img src="./assets/mevnnav.png" class="img-circle" style="...">
      <b-navbar-brand to="/">Donations Web App</b-navbar-brand>
      <b-collapse is-nav id="nav_collapse">
        <b-navbar-nav...>
        <b-navbar-nav class="ml-auto"...>
      </b-collapse>
    </b-navbar>
    <router-view/>
  </div>
</template>

<script>
export default {
  name: 'App'
}
</script>

<style>
#app {
  font-family: 'Avenir', Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  padding-top: 20px;
  background: url("./assets/homerbgyellow.png") no-repeat center top;
}
.VueTables__sortable {
  cursor: pointer;
}
</style>
```

Project Structure & Webpack

The ***assets*** folder – Again, fairly self explanatory ☺

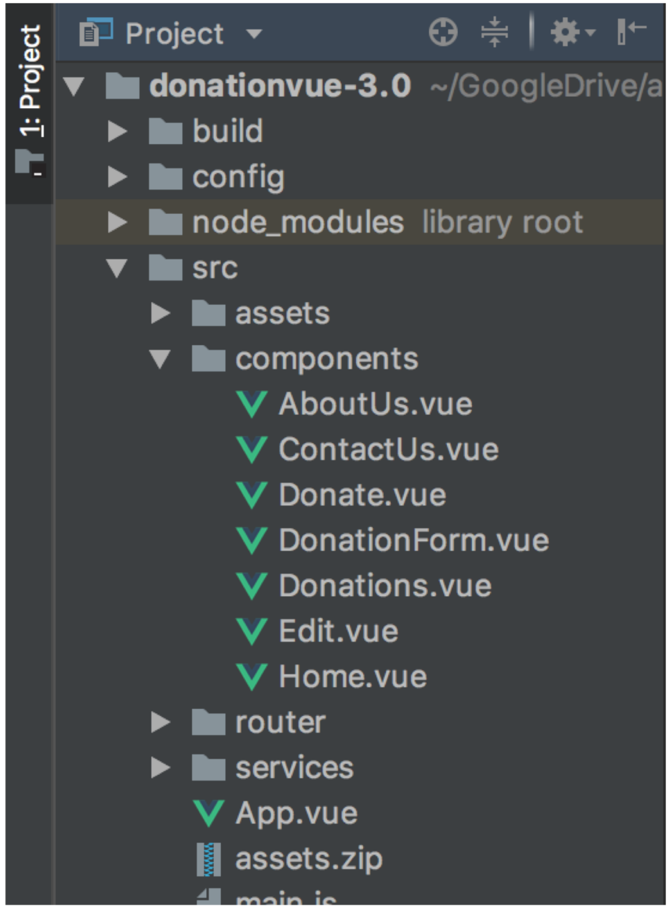


```
▼ donationvue-3.0  ~/GoogleDrive/a
  ▶ build
  ▶ config
  ▶ node_modules  library root
  ▼ src
    ▼ assets
        homer.gif
        homerbgyellow.png
        logo.png
        mevnnav.png
    ▶ components
    ▶ router
    ▶ services
      App.vue
      assets.zip
      main.js
  ▶ static
  ▶ test
    .babelrc
```

Project Structure & Webpack

The ***components*** folder is where you store all your Single-File Components (surprise, surprise!!)

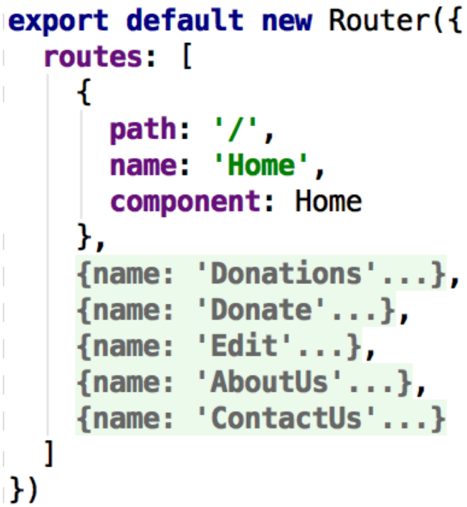
Here we have all our components in a single folder but it's common practice to create subfolders for all your related components (buttons, nav-drawers, links etc.) as the project grows.



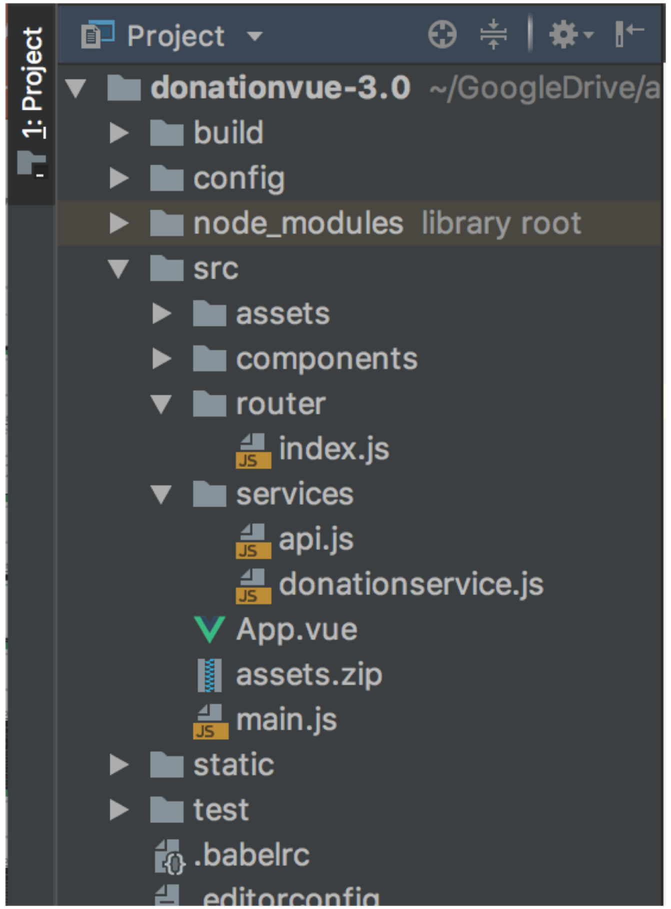
Project Structure & Webpack

The ***router*** folder is where (you've guessed it) our routes for our web app goes.

When apps grow, the number of routes grows, with **Before Enter** hooks etc, so it's a good idea to put your router in a separate file, or files. With a subdirectory you can keep this all in one place without bloating one src/router.js file.



```
export default new Router({
  routes: [
    {
      path: '/',
      name: 'Home',
      component: Home
    },
    {name: 'Donations'...},
    {name: 'Donate'...},
    {name: 'Edit'...},
    {name: 'AboutUs'...},
    {name: 'ContactUs'...}
  ]
})
```

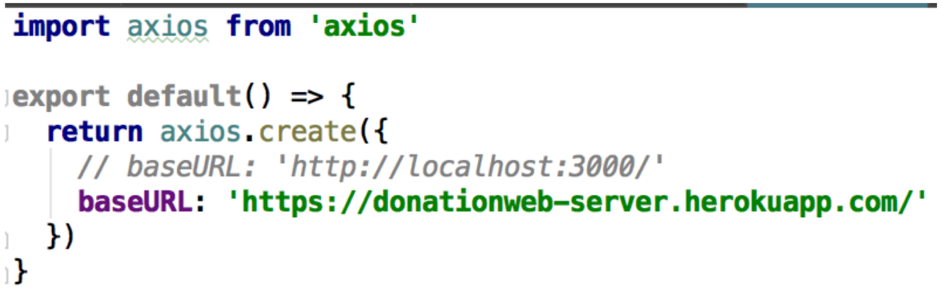


```
▼ donationvue-3.0 ~/GoogleDrive/a
  ▶ build
  ▶ config
  ▶ node_modules library root
  ▼ src
    ▶ assets
    ▶ components
    ▼ router
        index.js
    ▼ services
        api.js
        donationservice.js
      App.vue
      assets.zip
      main.js
  ▶ static
  ▶ test
    .babelrc
    .editorconfig
```

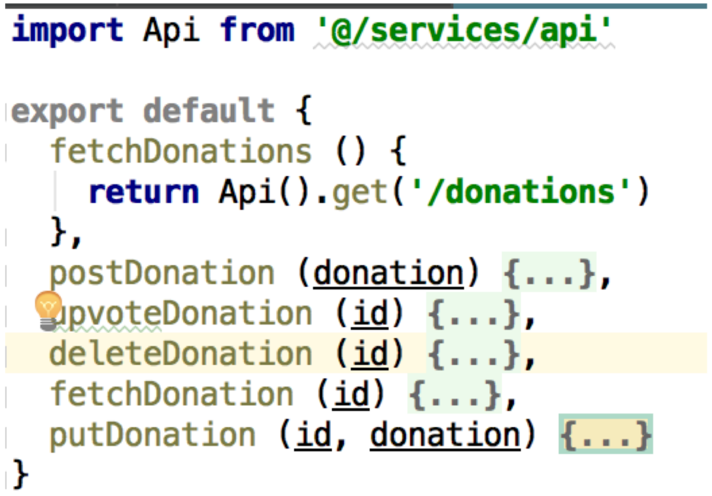
Project Structure & Webpack

The ***services*** folder is our own custom folder for our RESTful Api.

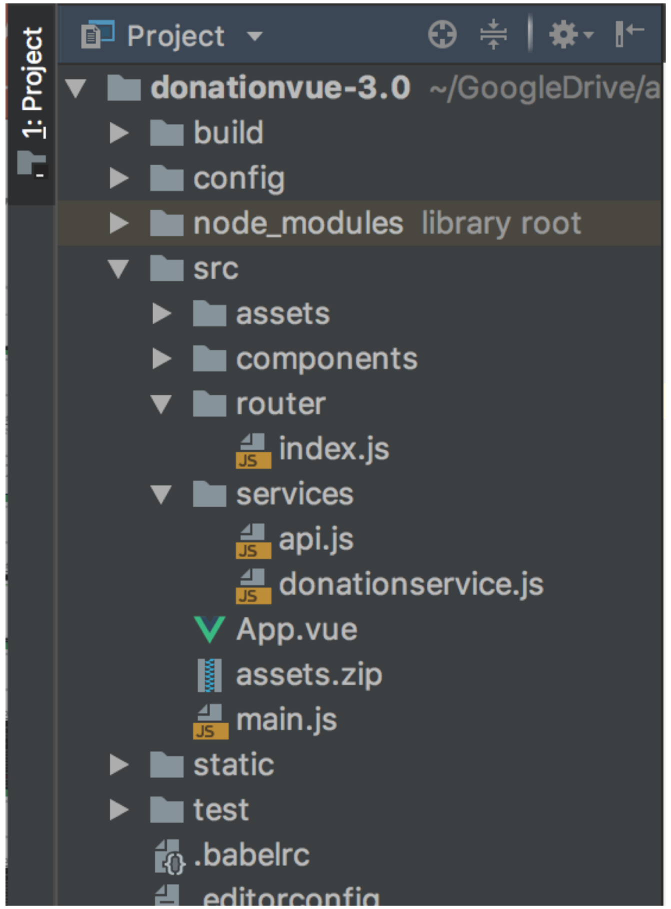
Again, it makes sense to keep this in separate files in a separate subfolder for maintainability and usability purposes.



```
import axios from 'axios'

export default() => {
  return axios.create({
    // baseURL: 'http://localhost:3000/'
    baseURL: 'https://donationweb-server.herokuapp.com/'
  })
}
```



```
import Api from '@/services/api'

export default {
  fetchDonations () {
    return Api().get('/donations')
  },
  postDonation (donation) {...},
  upvoteDonation (id) {...},
  deleteDonation (id) {...},
  fetchDonation (id) {...},
  putDonation (id, donation) {...}
}
```



References

- ❑ https://vuejs.org
- ❑ https://webpack.js.org/concepts/

Questions?