

HolaMundo con VueJS

Rodriguez Perez, Juan Manuel



Contenidos

1. Introducción	2
2. Que necesitamos	3
3. Creando nuestro HolaMundo	4
4. Ya creado el proyecto	6
4.1 Ejemplo de los archivos mencionados.....	7
5. Ejecutando la app	11
6. Bibliografía	12
7. Control de versiones	13

1. Introducción

Documentación de la creación de un HolaMundo usando el framework VueJS.

2. Que necesitamos

Antes de empezar recordar que necesitamos tener instalado para crear nuestra aplicación.

- a. Tener instalado Node (versión ≥ 6) y NPM 3+
- b. Recomendable usar la extensión **vue-devtools** en tu navegador.
- c. Instalar Vue mediante el comando **npm -g install vue** desde la consola de Node
- d. Instalar Vue-cli: **npm -g install vue-cli**
- e. Un editor de texto (en mi caso usare el VSCode que integra también la opción de consolas de comando desde el propio editor).

3. Creando nuestro HolaMundo

Para crear nuestra aplicación una vez lo tengamos todo instalado es muy sencillo sirve con ejecutar el siguiente comando:

vue init webpack my-app

Una vez ejecutado comenzara la instalación, con diferencia con otros frameworks como puede ser Angular durante la instalación te preguntara ciertas opciones de instalación.

a. Nombre del proyecto:

```
? Project name (Hello-World-Vue) |
```

Si no introducimos un nombre por defecto dejara el marcado entre paréntesis que será el que coincida con el indicado en el comando usado para crear la app con vue-cli.

Por ejemplo, en mi caso, debo introducir otro nombre al usar mayúsculas, como bien podemos observar en la notificación que obtenemos por consola.

```
? Project name (Hello-World-Vue) |
>> Sorry, name can no longer contain capital letters.
```

b. Descripción del proyecto.

Si en el paso anterior va todo correcto nos pedirá que le demos una descripción al proyecto.

```
? Project name hello-world-vue
? Project description (A Vue.js project) Proyecto para la creación de un hola mundo y comparación con otros frameworks.
```

c. Indicar el nombre del autor:

```
? Author (Juan Manuel Rodriguez Perez <juanmanuelrodriguez@afaya.es>) Juan Manuel Rodriguez Perez Afaya
```

d. Opción de compilación:

En este punto nos dará a elegir entre dos opciones a elegir con las flechas del teclado.

```
? Vue build
> Runtime + Compiler: recommended for most users
  Runtime-only: about 6KB lighter min+gzip, but templates (or any Vue-specific HTML) are ONLY allowed in .vue files - render functions are required elsewhere
```

Yo he elegido la que recomienda para la mayoría de los usuarios.

e. Opción de instalar router:

En esta opción le podemos indicar si queremos que nos instale el gestor de routing de vue.

```
? Install vue-router? (Y/n) y
```

f. Opción instalar ESLint:

ESLint, para quien no lo sepa es un helper de EcmaScript para ayudar que nuestro código sea mucho más limpio ayudándonos con notificaciones mientras desarrollamos.

```
? Use ESLint to lint your code? (Y/n) y
```

g. Elección del tipo de ESLint a usar:

En esta pregunta podemos elegir entre varias opciones de configuración del ESLint.

```
? Pick an ESLint preset (Use arrow keys)
> Standard (https://github.com/standard/standard)
  Airbnb (https://github.com/airbnb/javascript)
  none (configure it yourself)
```

h. Preparación de Unit Test

Nos da la opción de preparar la app para el uso de Unit Test

```
? Set up unit tests (Y/n) y
```

i. Selección de herramienta de ejecución de los test.

```
? Pick a test runner (Use arrow keys)
> Jest
  Karma and Mocha
  none (configure it yourself)
```

j. Preparar test e2e

```
? Setup e2e tests with Nightwatch? (Y/n) y
```

k. Opción de ejecutar automáticamente npm install

En este paso nos hace una pregunta interesante, ya que nos pregunta si después de la instalación queremos que nos ejecute automáticamente el comando npm install de node por el cual instalara todas las librerías necesarias teniendo en cuenta también las opciones elegidas anteriormente.

```
? Should we run `npm install` for you after the project has been created? (recommended) (Use arrow keys)
> Yes, use NPM
  Yes, use Yarn
  No, I will handle that myself
```

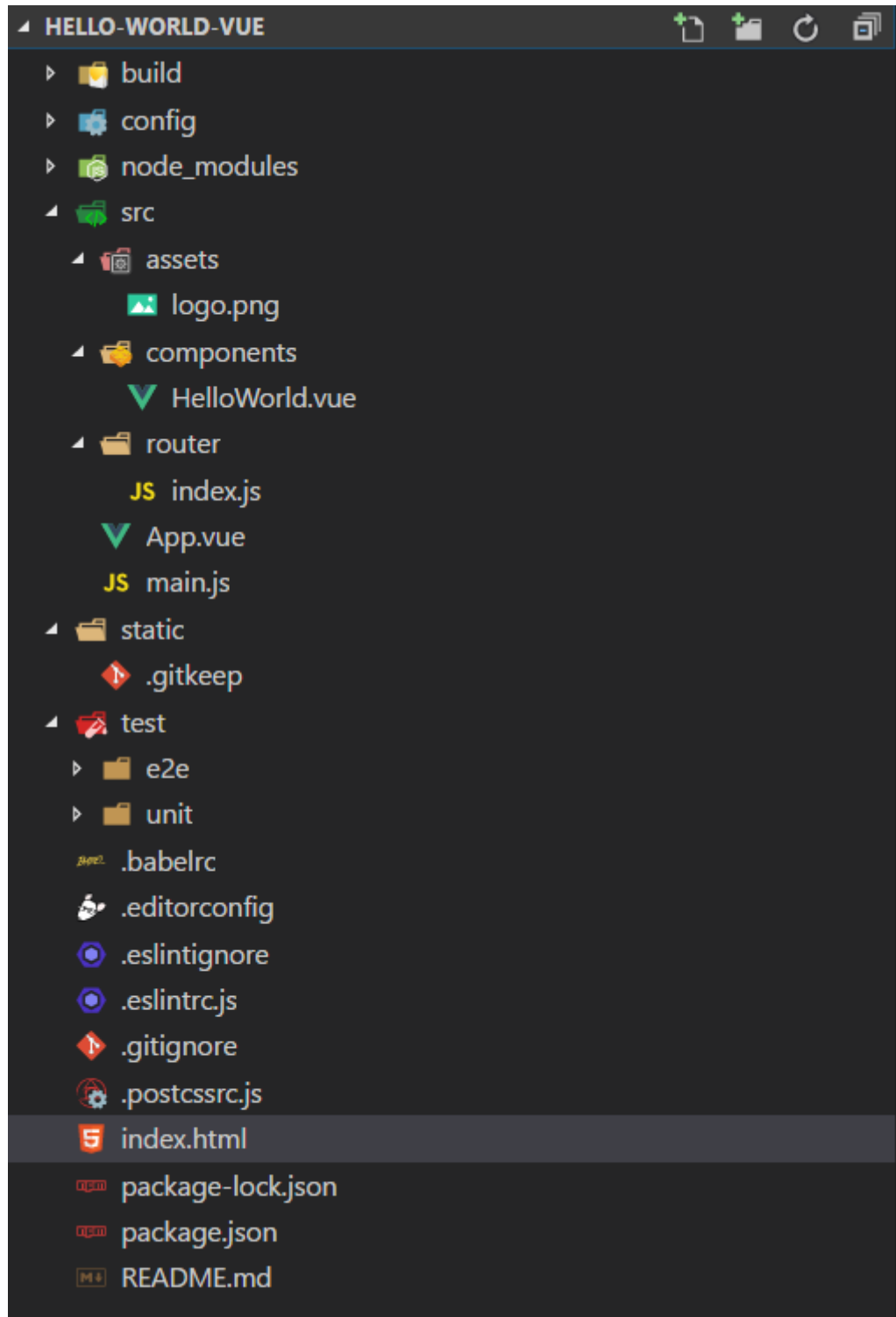
Una vez esto iniciara la creación del proyecto:

```
vue-cli - Generated "Hello-World-Vue".

# Installing project dependencies ...
# =====
[ ] ..... | fetchMetadata: sill pacote range manifest for eslint-friendly-formatter@^3.0.0 fetched in 84ms
```

4. Ya creado el proyecto

Una vez finalizado los pasos del punto 3, vue cli nos habrá creado el proyecto con la siguiente estructura (puede variar según las opciones seleccionadas).



En la carpeta **src** será donde vayamos realizando nuestro desarrollo de la aplicación, en ella podemos destacar las siguientes carpetas y archivos.

- Assets: Será la carpeta para alojar recursos como imágenes, etc.
- Components: Carpeta para ir albergando los diferentes componentes vue desarrollados.
- Router: Tendrá el archivo js con el enrutamiento que queramos para nuestra app.
- App.vue: Será el componente principal de la aplicación y el padre del resto de componentes será el que renderice vue y será el llamado desde el index.html
- Main.js: Archivo js principal de configuración de la aplicación donde por ejemplo declararemos los componentes que vayamos creando entre otros.
- Index.html: Archivo index de la aplicación punto de entrada de la aplicación y archivo principal usado por los navegadores.

4.1 Ejemplo de los archivos mencionados

a. Index.html:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <title>hello-world-vue</title>
  </head>
  <body>
    <div id="app"></div>
    <!-- built files will be auto injected -->
  </body>
</html>
```

Es un archivo html común, donde nos debemos de fijar en el div con el identificador **id="app"** que será usado por el archivo main.js para montar el componente principal.

b. App.vue:

Componente principal de la app, es un archivo vue, donde podemos observar y ver que llama la atención de como Vue nos aconseja que queda componente este en un único fichero conteniendo tanto el código html, javascript y css.

El código html como veis se encierra entre las etiquetas de <template>

```
<template>
  <div id="app">
    
    <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;
  margin-top: 60px;
}
</style>
```

Al ser el componente principal tendrá la etiqueta :

```
<router-view/>
```

Donde se irán cargando el resto de los componentes según naveguemos por la app.

- c. Componente secundario, HelloWorld.vue:
Este componente será el que contenga el contenido, en este caso el saludo típico de Hola Mundo.

```
1  <template>
2    <div class="hello">
3      <h1>{{ msg }}</h1>
4    </div>
5  </template>
6
7  <script>
8    export default {
9      name: 'HelloWorld',
10     data () {
11       return {
12         msg: 'Hola Mundo usando Vue.'
13       }
14     }
15   }
16 </script>
17
18 <!-- Add "scoped" attribute to limit CSS to this component only -->
19 <style scoped>
20   h1, h2 {
21     font-weight: normal;
22   }
23   ul {
24     list-style-type: none;
25     padding: 0;
26   }
27   li {
28     display: inline-block;
29     margin: 0 10px;
30   }
31   a {
32     color: #42b983;
33   }
34 </style>
35
```

d. Archivo index.js del router:

Este archivo contendrá nuestras rutas de la aplicación y en nuestro caso será el que indique que inicialmente dentro de la etiqueta router-view que esta en el App.vue se cargue el anterior componente al indicarlo en la especificación como lo vemos en la siguiente imagen.

```
1  import Vue from 'vue'
2  import Router from 'vue-router'
3  import HelloWorld from '@/components/HelloWorld'
4
5  Vue.use(Router)
6
7  export default new Router({
8    routes: [
9      {
10       path: '/',
11       name: 'HelloWorld',
12       component: HelloWorld
13     }
14   ]
15 })
16
```

5. Ejecutando la app

Para ejecutar la aplicación desde la carpeta de la misma usamos el comando:

- **npm run dev**

```
> hello-world-vue@1.0.0 dev [redacted] Hello-World-Vue
> webpack-dev-server --inline --progress --config build/webpack.dev.conf.js

95% emitting

DONE Compiled successfully in 4405ms

I Your application is running here: http://localhost:8080
```

Abrimos el navegador y escribimos la url que nos indica en la consola de comandos.



Y podemos observar como nos carga nuestra app.

6. Bibliografía

Enlaces interesantes a tener en cuenta o donde se ha obtenido información.

- Node: <https://nodejs.org/es/>
- VueJS: <https://vuejs.org/>
- vue-devtools: <https://github.com/vuejs/vue-devtools#vue-devtools>
- Vue.JS school:

https://vueschool.io/?utm_source=Vuejs.org&utm_medium=Banner&utm_campaign=Sponsored%20Banner&utm_content=V1

- Presentación de jdonsan “charla-aprendiendo-vuejs”:
<https://jdonsan.github.io/charla-aprendiendo-vuejs/slides/#slide=1>
- ESLint: <https://eslint.org/>

7. Control de versiones

Versión	Descripción	Fecha
1.0	Versión Inicial	10/06/2018
2.0	Versión Revisada	08/07/2018