



Alexander McRae

Posted on 28 avr. 2020

# Setting up Vue and Phoenix 1.5 with vue-cli

#elixir #vue #javascript #phoenix

Setting up Vue with the Phoenix framework is oddly difficult, in this guide I walk through how I have done it.

At the end of this guide you will have a Phoenix application which serves the Vue application with hot code reloading using webpack.

## Why?

Everyone will have a different reasons but I want to use Vue for the Progressive Web App support.

## Setup Phoenix

The first thing we are going to do it setup Phoenix. If you want you can use the `--no-webpack` option. I will include it for not since I want admin pages which use Phoenix's template system.

```
mix phx.new vue_phx
cd vue_phx
```

## Setup Vue

Now we create the Vue application with the vue-cli. I have choosen to call mine app but name it whatever you want.

```
vue create app
```

Go through and select the features you want. Then we create and edit the `vue.config.js` file in the root of the new vue application.

```
// vue_phx/app/vue.config.js
const path = require("path");

module.exports = {
  outputDir: path.resolve(__dirname, "../priv/app"),
};
```

This will change where you Vue app is outputted. If you choose the no webpack option then you can change it to `../priv/static` but agian for my admin pages I keep them seperate.

One last thing before we move on is to install the webpack-cli

```
cd app
npm install -D webpack-cli
```

## Making Phoenix start the webpack watcher

Now in the dev config of the phoenix application we will add another watcher for the vuejs application.

```
# vue_phx/config/dev.ex
...

config :village, VillageWeb.Endpoint,
  http: [port: 4000],
  debug_errors: true,
  code_reloader: true,
  check_origin: false,
  watchers: [
    node: [
      "node_modules/webpack/bin/webpack.js",
      "--mode",
      "development",
      "--watch-stdin",
      cd: Path.expand("../assets", __DIR__)
    ],
    node: [
      "node_modules/webpack/bin/webpack.js",
      "--mode",
      "development",
      "--watch-stdin",
      "--config",
      "node_modules/@vue/cli-service/webpack.config.js",
      cd: Path.expand("../app", __DIR__)
    ]
  ]
...

```

The first watcher won't be there if you choose the no-webpack option. The second watcher tells phoenix to start the webpack cli and passes in the [generated config](#) as an option.

Note, this means we will not start the frontend using

`npm run serve` as Phoenix will serve the static files and webpack will do the hot reloading for us.

## Getting Phoenix to serve the frontend

Now we are going to get phoenix to serve the application at `localhost:4000/`.

In `lib/vue_phx_web/endpoint.ex` there is a static file server using

`Plug.Static`. We are going to add another static file server right below it.

I change the original aswell to serve `at: "/admin"`.

```
# vue_phx/lib/vue_phx_web/endpoint.ex

...

plug Plug.Static,
  at: "/",
  from: {:vue_phx, "priv/app"},
  gzip: false,
  only: ~w(index.html manifest.json service-worker.js css fonts img js favicon.ico)
  only_matching: ["precache-manifest"]

...

```

Now if you go to `localhost:4000/index.html` you should see your Vue app.

The issue with this is that `localhost:4000/` doesn't serve it correctly.

We can fix that by creating a page controller.

```
# vue_phx/lib/vue_phx_web/controllers/page_controller.ex

defmodule VuePhxWeb.PageController do
  use VuePhxWeb, :controller

  def index(conn, _params) do
    conn
    |> put_resp_header("content-type", "text/html; charset=utf-8")
    |> Plug.Conn.send_file(200, "priv/app/index.html")
    |> halt()
  end
end
```

This will serve the correct file. Now we just add it to the `router.ex`

```
# vue_phx/lib/vue_phx_web/router.ex

defmodule VuePhxWeb.Router do
  use VuePhxWeb, :router

  pipeline :browser do
    plug :accepts, ["html"]
    plug :fetch_session
    plug :fetch_flash
    plug :protect_from_forgery
    plug :put_secure_browser_headers
  end

  scope "/", VuePhxWeb do
    pipe_through :browser

    get "/*path", PageController, :index
  end
end
```

Now `localhost:4000/` should serve your Vue application. Let me know if you have any issues! Shoot me an email at [mail@alexandermcrae.com](mailto:mail@alexandermcrae.com)

## Discussion (6)



Josh • Jun 17 '20

...



Hey this is a pretty cool setup, I love how you have it all setup in one application. Man, looking at how I had done separate admin/public frontends is a nightmare compared to this! Thanks again for sharing this!



Alexander McRae • Jun 25 '20

...



Thanks so much for the comment!



cyberlion85 • Dec 21 '21

...



Hello, do you have a git repository for this project?



Chris Owen • Aug 3 '21

...