

Elixir and Phoenix Framework

Adrian Kosmaczewski

2022-02-25

I've been learning a bit about the Elixir programming language lately, and for that I created a small app using the Phoenix Framework.

I had never used Elixir before, so this post contains some information about how I got started. You can see the results of this experiment on [GitLab](#).

Install Erlang, Elixir, and Phoenix

Elixir is a language that compiles to BEAM bytecode, running in the Erlang virtual machine. Getting Elixir installed is explained in detail in the [Elixir documentation](#). Here's what I had to do on Ubuntu 20.04:

```
$ wget https://packages.erlang-solutions.com/erlang-solutions_2.0_all.deb
$ sudo dpkg -i erlang-solutions_2.0_all.deb
$ sudo apt update
$ sudo apt-get install esl-erlang
$ sudo apt install elixir
```

The Elixir language itself feels very Ruby-like at first approach, and Phoenix is quite a bit like Rails, but of course with all the power of the Erlang VM behind. All of this is very intentional, of course.

```
defmodule FortuneElixirWeb.PageController do
  use FortuneElixirWeb, :controller

  def index(conn, _params) do
    number = :rand.uniform(1000)
    message = elem(System.cmd("fortune", []), 0)
    hostname = elem(System.cmd("hostname", []), 0)
    version = "1.2-elixir"
    fortune = %{ "number" => number, "message" => message, "hostname" => hostname, "version"
    accept = case get_req_header(conn, "accept") do
      [version] -> version
      _ -> "text/html"
    end
    case accept do
```

```
    "application/json" -> json(conn, fortune)
    "text/plain" -> text(conn, "Fortune #{version} cookie of the day ##{number}:\n\n#{message}")
    _ -> render(conn, "index.html", fortune: fortune)
  end
end
end
```

(Yes, it's the same app I made with Rust and Python a while ago.)

Speaking about the framework, here it goes:

```
$ mix archive.install hex phx_new
```

Create and Run a Project

And once you have the framework installed, creating a new project is very simple:

```
$ mix phx.new fortune_elixir --no-ecto
```

The `--no-ecto` option removes the boilerplate for PostgreSQL connectivity, which is otherwise installed by default. I wanted to create the simplest possible project.

Finally, start the project and open it in your browser:

```
$ cd fortune_elixir
$ mix phx.server
```

Connect to <http://localhost:4000> to see the application in action.