

Installing Phoenix, Elixir and PostgreSQL on FreeBSD

Brendan Sechter

2016-03-20

This post covers installing Phoenix, Elixir and PostgreSQL on a clean installation of FreeBSD. The post assumes FreeBSD is installed and you have created an unprivileged user. FreeBSD installation instructions can be found [here](#).

Software Versions

```
$ date -u "+%Y-%m-%d %H:%M:%S +0000"
2016-03-19 07:35:02 +0000
$ uname -vm
FreeBSD 11.0-CURRENT #0 r296925: Wed Mar 16 20:53:04 JST 2016      root@mirage.sennue.com:/usr/o
$ psql --version
psql (PostgreSQL) 9.4.6
$ mix hex.info
Hex:      0.11.3
Elixir:   1.2.3
OTP:      18.2.4
* snip *
$ mix phoenix.new -v
Phoenix v1.1.4
```

Bootstrapping the system

Login as root and install **portmaster**. If you are using packages instead of ports, you will need to install packages instead of using **portmaster**.

```
portsnap fetch extract
(cd /usr/ports/ports-mgmt/portmaster && make config-recursive && make config-recursive && make
portmaster security/sudo
```

The following utilities may also be useful. Note that **ftp/curl** is used at the end of the post.

```
portmaster ports-mgmt/portupgrade devel/git shells/bash editors/vim ftp/curl ftp/wget
```

Installing PostgreSQL

Install **databases/postgresql94-server** as root. If you would rather install a different version, do that instead.

```
portmaster databases/postgresql94-server
```

Initialize the database.

```
service postgresql initdb
```

If you want to enable remote connections, add the following line to **/usr/local/pgsql/data/postgresql.conf**.

```
listen_addresses = '*'
```

If you want to change the port PostgreSQL listens on, add the following line to **/usr/local/pgsql/data/postgresql.conf**. Replace 5432 with the port you want to use.

```
port = 5432                                # (change requires restart)
```

To use password hash authentication, add the following line to **/usr/local/pgsql/data/pg_hba.conf**. Replace 192.168.0.0/16 with your own network. Consider changing “trust” to “md5” for local and loopback IP address connections.

```
host    all             all             192.168.0.0/16          md5
```

Enable PostgreSQL in **/etc/rc.conf**.

```
echo 'postgresql_enable="YES"' >> /etc/rc.conf
```

Start PostgreSQL.

```
service postgresql start
```

Add a PostgreSQL super-user with database and role creation privileges. Replace username with your unprivileged FreeBSD user login.

```
su postgres
createuser -sdrP username
```

You should now be able to start **psql** as your non-root user.

```
psql
```

Installing Elixir and Phoenix

Install Elixir and Phoenix.

```
sudo portmaster lang/elixir
mix local.hex
mix archive.install https://github.com/phoenixframework/archives/raw/master/phoenix_new.ez
```

Optionally, install **npm** and **brunch**.

```
sudo portmaster www/node www/npm
sudo npm install -g brunch
```

Creating a Sample Project

Create a new Phoenix project.

```
mix phoenix.new phoenix_service --no-brunch
cd phoenix_service
mix ecto.create
```

Create a simple JSON memo service.

```
mix phoenix.gen.json Memo memos title:string body:string
```

Revise the **web/router.ex** file. The “/api” scope needs to be uncommented and the “/memos” route needs to be added.

web/router.ex file

```

defmodule PhoenixService.Router do
  use PhoenixService.Web, :router

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

  pipeline :api do
    plug :accepts, ["json"]
  end

  scope "/", PhoenixService do
    pipe_through :browser # Use the default browser stack

    get "/", PageController, :index
  end

  scope "/api", PhoenixService do
    pipe_through :api

    resources "/memos", MemoController, except: [:new, :edit]
  end
end

```

Run the migration.

```
mix ecto.migrate
```

Tests should pass.

```
mix test
```

Running the Sample Project

Start the server.

```
mix phoenix.server
```

The following **curl** commands can be used to interact with the running JSON API. My [previous post](#) covers a shell script for interacting with this sample app.

```

# POST new
curl -H 'Content-Type: application/json' -X POST -d '{"memo": {"title": "New Title", "body": "T
# PATCH id 1
curl -H 'Content-Type: application/json' -X PATCH -d '{"memo": {"title": "Patched Title"}}' htt
curl -H 'Content-Type: application/json' -X PATCH -d '{"memo": {"body": "Patched memo body."}}'
# PUT id 1
curl -H 'Content-Type: application/json' -X PUT -d '{"memo": {"title": "Updated Title", "body":
# GET all
curl -H 'Content-Type: application/json' http://localhost:4000/api/memos
# GET id 1
curl -H 'Content-Type: application/json' http://localhost:4000/api/memos/1

```

```
# DELETE id 1
curl -H 'Content-Type: application/json' -X DELETE http://localhost:4000/api/memos/1
```

References:

- [Phoenix, Deployment](#)
- [Phoenix, Building a JSON API With Phoenix](#)
- [Phoenix, Building a versioned REST API with Phoenix Framework](#)
- [Phoenix, A Shell Script for Working with Phoenix JSON APIs](#)
- [Elixir as a Service on FreeBSD](#)
- [PostgreSQL, Installing PostgreSQL on FreeBSD](#)
- [PostgreSQL, Installing and Using Postgresql on FreeBSD](#)
- [PostgreSQL, installation and configuration](#)
- [Sh - the Bourne Shell](#)
- [FreeBSD, Installing FreeBSD](#)