



**Build a Chat Service
with
Elm & Elixir “Phoenix Framework”**

About us

Helder Pinto

Web developer at Yarilabs

github:

@helderjnpinto, @hjnp

Xavier Francisco

Developer at Yarilabs

github:

@qu4tro

About this project

Main goals:

- Create a bigger community on *LambdaPT*.
- To experiment and learn with powerful and new technologies (Elm & Elixir).



<https://lambdapt-autoinvite.herokuapp.com/>

Powered by:



Project concept



Components

- Widget web chat inside of “Host web page”.



- Elixir with phoenix framework “ Server ”



Phoenix Framework

- Slack bot “Connect widgets with hosts”:



- Database “Store some data”:



Components

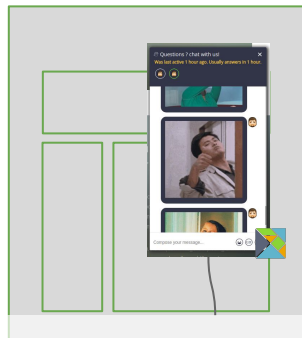


Hosts



Slack bot (or any other integration)

Host web page



Websockets /
Phoenix
Channel



Database



PostgreSQL

Visitor(s)



Elm architecture

Basic Html elm program:

- Model — the state of your application
- Update — a way to update your state
- View — a way to view your state as HTML



The Elm Architecture |> Effects

- Model — the state of your application
- Update — a way to update your state
- View — a way to view your state as HTML

Instead of just producing `Html`, we will now be producing commands and subscriptions:

- **Commands** — A `Cmd` lets you *do* stuff: generate a random number, send an HTTP request, etc.
- **Subscriptions** — A `Sub` lets you register that you are interested in something: tell me about location changes, listen for web socket messages, etc.



The Elm Architecture |> Example

```
main =  
  Html.beginnerProgram { model = model, view = view, update = update }
```

-- MODEL

```
type alias Model =  
  Int
```

```
model : Model  
model =  
  0
```

-- UPDATE

```
type Msg = Increment | Decrement
```

```
update : Msg -> Model -> Model
```

```
update msg model =
```

```
  case msg of  
    Increment ->  
      model + 1
```

```
    Decrement ->  
      model - 1
```

-- VIEW

```
view : Model -> Html Msg
```

```
view model =
```

```
  div []  
    [ button [ onClick  
      Decrement ] [ text "-" ]  
      , div [] [ text (toString  
        model) ]  
      , button [ onClick  
        Increment ] [ text "+" ]  
    ]
```

Phoenix Framework

Basic introduction:

- Phoenix is a web development framework written in Elixir which implements the server-side MVC “Model-View-Controller” pattern.
- In this case we’re using **Channels** to handle messages over the websocket connection.
- Using websockets, it’s really easy to implement real-time communication.

Phoenix Framework |> Module |> Channels

- Over the **Channels** layer, Phoenix allows for real-time two-way communication between client-server.
- **Channels** are based on a simple idea - **sending** and **receiving** messages or **broadcast** messages on topics.
- **Senders** can publish messages to those topics and **Receivers** can subscribe to topics and receive any message send to those.

Demo

<https://mercurio-chat.netlify.com>

Open source project github repos:

- Elm (<https://github.com/realyarilabs/chat-elm-client>)
- Elixir (<https://github.com/realyarilabs/chat-server>)

QUESTIONS ?