



**Writing real-time web  
applications without the  
need of writing javascript.**

Ruben Amortegui  
@ramortegui  
<https://github.com/ramortegui>



# Agenda

- News
- Hands on! Phoenix.LiveView 101  
Simple counter with Phoenix.LiveView
- Case: Phoenix Trivia

# News

- Elixir 1.9 released (June 24th)
  - Releases

# Old News

<https://elixirconf.com>

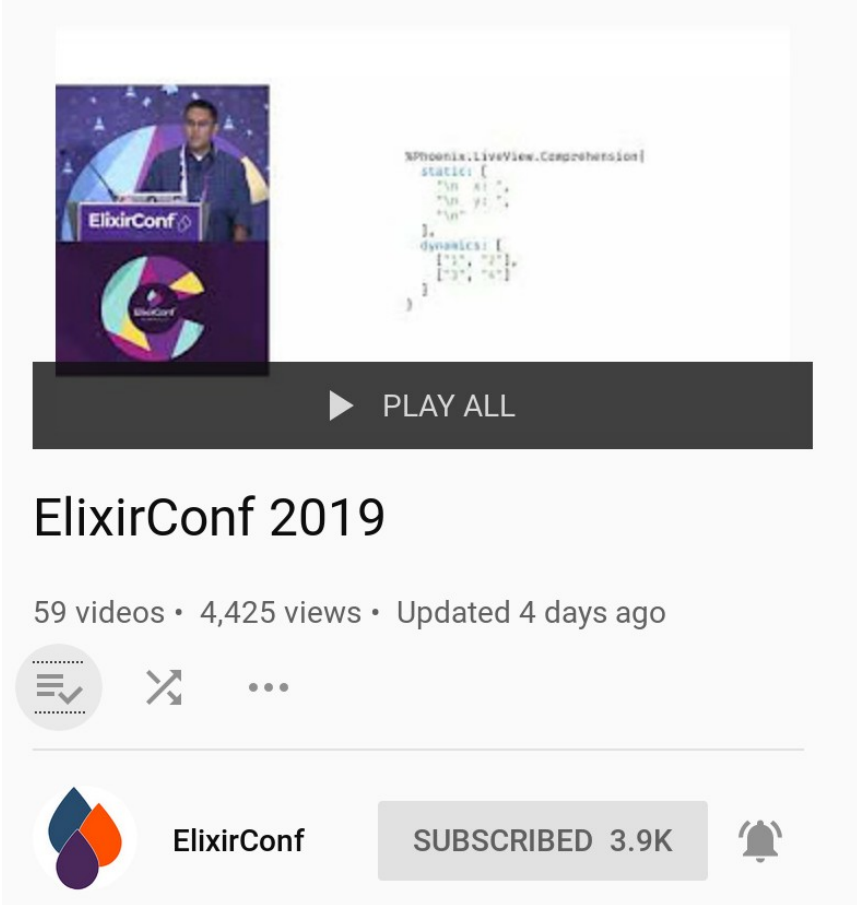


<https://www.thebigelixir.com/>

– THEBIGMEETUP2019 (-20%)



# News



The image shows a YouTube video player interface. The video thumbnail features a man speaking at a podium with the ElixirConf logo. To the right of the thumbnail, a snippet of Elixir code is displayed: 

```
%Phoenix.LiveView.Comprehension|  
static: [  
  %{"id": "1"},  
  %{"id": "2"},  
  %{"id": "3"}  
],  
dynamic: [  
  %{"id": "4"},  
  %{"id": "5"}  
]  
]
```


. Below the video player, the title "ElixirConf 2019" is shown, followed by the statistics "59 videos • 4,425 views • Updated 4 days ago". Below this, there are three icons: a list icon, a share icon, and a more options icon. At the bottom, the channel name "ElixirConf" is displayed next to its logo, followed by a "SUBSCRIBED 3.9K" button and a notification bell icon.

PLAY ALL

## ElixirConf 2019

59 videos • 4,425 views • Updated 4 days ago

⋮ ↗ ⋮

 ElixirConf SUBSCRIBED 3.9K 🔔

# LiveView Talks

1



25:44

**ElixirConf 2019 - Phoenix LiveView Demystified - Alex Garibay**

ElixirConf

5

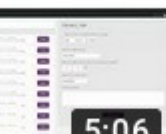


27:56

**ElixirConf 2019 - How LiveView Handles File Uploads - Gary Rennie**

ElixirConf

12

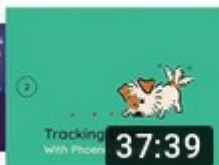


5:06

**ElixirConf 2019 - Lighting Talk - Phoenix LiveView in Healthcare - Tyler Clemens**

ElixirConf

28



37:39

**ElixirConf 2019 - Beyond LiveView: Building Real-Time... - Sophie DeBenedetto**

ElixirConf

59



55:21

**ElixirConf 2019 - Friday Evening Keynote - Chris McCord**

ElixirConf

# Phoenix.LiveView

- Phoenix LiveView enables rich, real-time user experiences with server-rendered HTML.

[https://github.com/phoenixframework/phoenix\\_live\\_view](https://github.com/phoenixframework/phoenix_live_view)

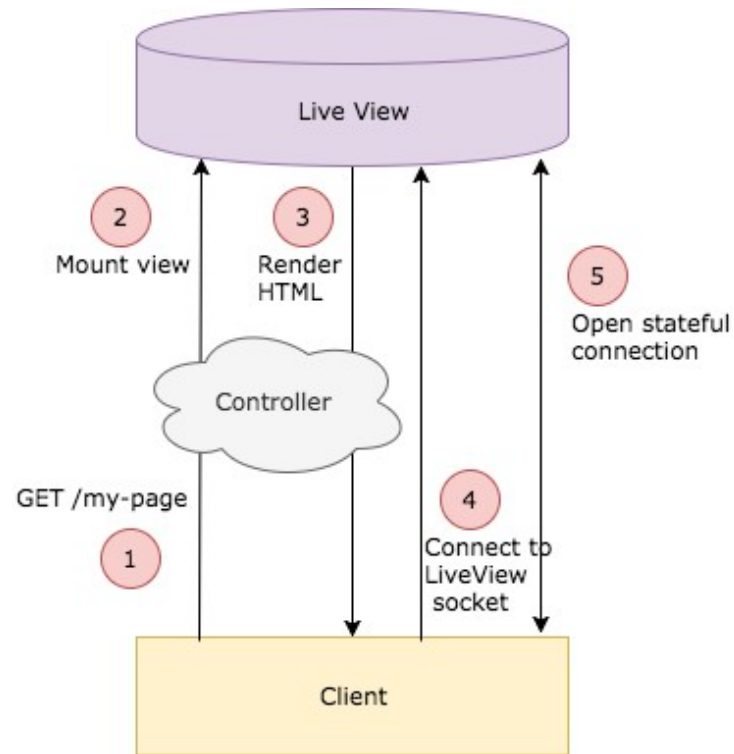


# Phoenix.LiveView

- Library
  - Elixir
    - Build on top of Phoenix.Channels  
<https://hexdocs.pm/phoenix/channels.html>
    - LiveView templates  
[https://hexdocs.pm/phoenix\\_live\\_view/Phoenix.LiveView.Engine.html](https://hexdocs.pm/phoenix_live_view/Phoenix.LiveView.Engine.html)
  - Javascript
    - Manage custom events, and update the DOM using a library called morphdom.
    - <https://github.com/patrick-steele-idem/morphdom>

# Phoenix.LiveView

- How it works?



<https://elixirschool.com/blog/phoenix-live-view/>

# Phoenix.LiveView

## Use cases

	SPA	LiveView
No-fuss SEO / Instant Load		✓
Offline Support	✓	
High latency clients	✓	
Native features / Complex UI	✓	
Minimal JS + payload		✓
Productivity		✓

[https://www.youtube.com/watch?v=XhNv1ikZNLs&list=PLqj39LCvnOWYTNs1n3ZNMSNO3Svv\\_XweT&index=60&t=0s](https://www.youtube.com/watch?v=XhNv1ikZNLs&list=PLqj39LCvnOWYTNs1n3ZNMSNO3Svv_XweT&index=60&t=0s)

# Phoenix.LiveView Samples

- [https://github.com/chris MCCord/phoenix\\_live\\_view\\_example](https://github.com/chris MCCord/phoenix_live_view_example)
- <https://live-view-collection.herokuapp.com/>

# Hands On!

- Counter using Phoenix.LiveView

# Summary Live View

- Add LiveView dependency
- Setup configuration (add signing salt)
- Import functions into the web context
- Enable the socket to connect using the /live route.
- Add Phoenix Live View javascript package as dependency.

[https://github.com/ramortegui/phoenix\\_trivia/commit/bc1bae7d32a24a51ca510e2ee3b2971d3c8b2735](https://github.com/ramortegui/phoenix_trivia/commit/bc1bae7d32a24a51ca510e2ee3b2971d3c8b2735)

# Sample Web App

- Trivia Game
  - Multi-Player
  - Questions come from external service
  - Players can create a trivia game
  - Players can join a trivia game
  - Game timed
  - “Real-time” updates

<https://en.wikipedia.org/wiki/Trivia>

# Data Structures

- Trivia.Player
- Trivia.Question
- Trivia.Game
- Trivia.GameServer
- Trivia.DynamicSupervisor



# LiveView

```
defmodule TriviaWeb.Live.TriviaView do
  use Phoenix.LiveView

  alias Trivia.Game
  alias Trivia.GameServer
  alias Trivia.Player

  def render(assigns) do
    TriviaWeb.PageView.render("trivia.html", assigns)
  end

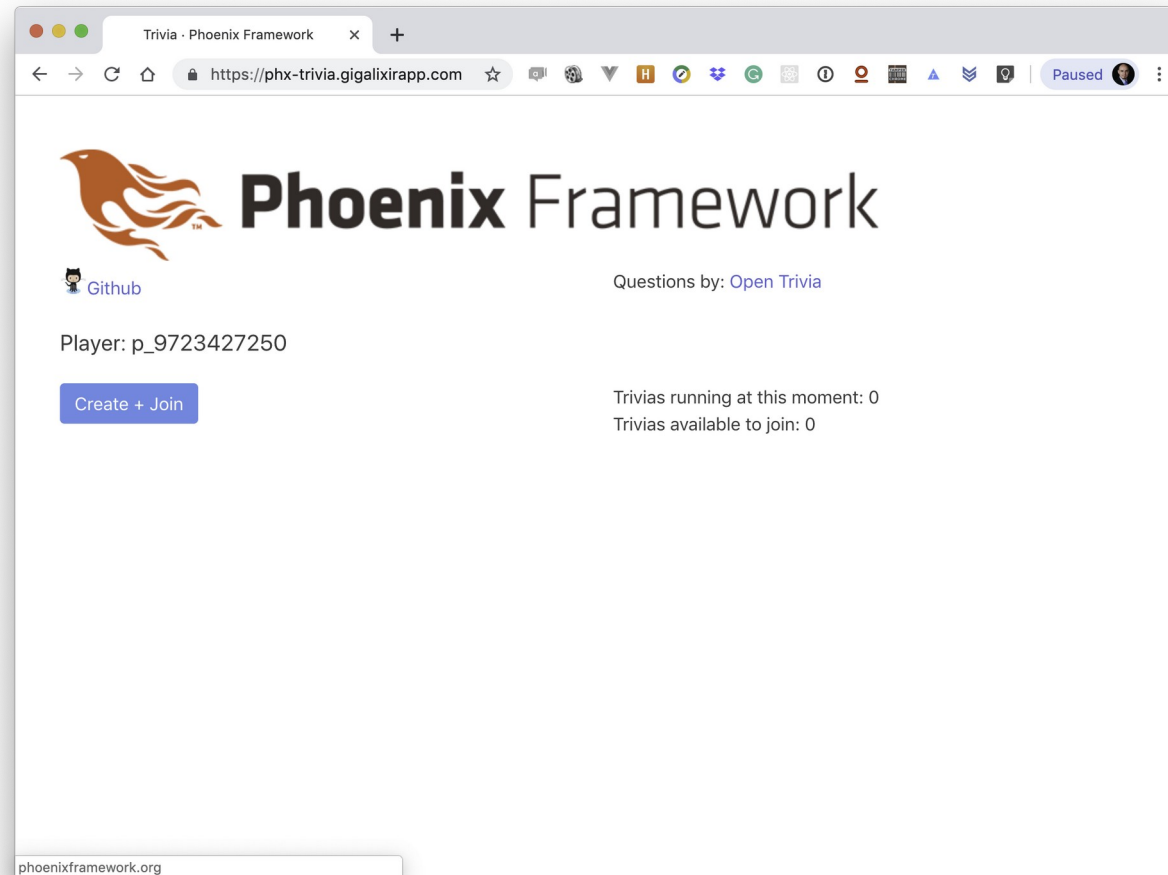
  def mount(_session, socket) do
    if(connected?(socket), do: :timer.send_interval(100, self(), :tick))

    player_name = "p_#{:rand.uniform(10_000_000_000)}"

    socket =
      socket
      |> assign(
        trivia: nil,
        process: nil,
        number_of_trivias: 0,
        number_of_available_trivias: 0,
        available_trivias: [],
        player_name: player_name,
        player_info: nil
      )

    {:ok, socket}
  end
end
```

# https://phx-trivia.gigalixirapp.com



# Q & A?

<https://www.meetup.com/Elixir-Calgary/>

Thanks!

[@ramortegui](#)

<https://github.com/ramortegui>