Building Multiplayer Games

with Phoenix and Phaser

Created by Keith Salisbury / @ktec

Fun Stack

The fun way to build scalable social games

Phoenix Framework

Productive. Reliable. Fast

A productive web framework that does not compromise speed and maintainability

Elixir

dynamic, functional, meta-programming aware

designed for building scalable and maintainable applications

Erlang

low-latency, distributed and fault-tolerant

build massively scalable soft real-time systems with requirements on high availability

Designed for web, desktop, mobile with amazing tutorials and great documentation

Phaser Game Framework

The fun, fast and free HTML5 Game Framework

Pixi Rendering Engine

Super fast HTML 5 2D rendering engine that uses webGL with canvas fallback

HTML5

Offline Storage, Device Access, Web Sockets

Multimedia, Graphics and Effects, Web Workers, CSS3

Features

Phoenix

easy - scalable - fast - friendly - channels - ES6/7 ready

Elixir

syntax - metaprogramming - macros - tooling

Erlang/OTP

semantics - fault tolerant - scalable - distributed

Pixi

ultra fast - multi platform - multi touch - sprite sheet

asset loader - auto-detect (webgl or canvas) - rich text

Phaser

preloader - physics - groups - sprites - camera

particles - input - sound - tile maps - scale manager

Pointless Social Stats

GitHub, Inc. [US] https://github.com/html5/spec

This repository Search Pull requests Issues Gist

html5 / spec

Watch Star 23 Fork 2

Code Issues 1 Pull requests 0 Wiki Pulse Graphs

The HTML5 Specification - specification editing and generation tools http://whatwg.org/

38 commits 2 branches 0 releases 2 contributors

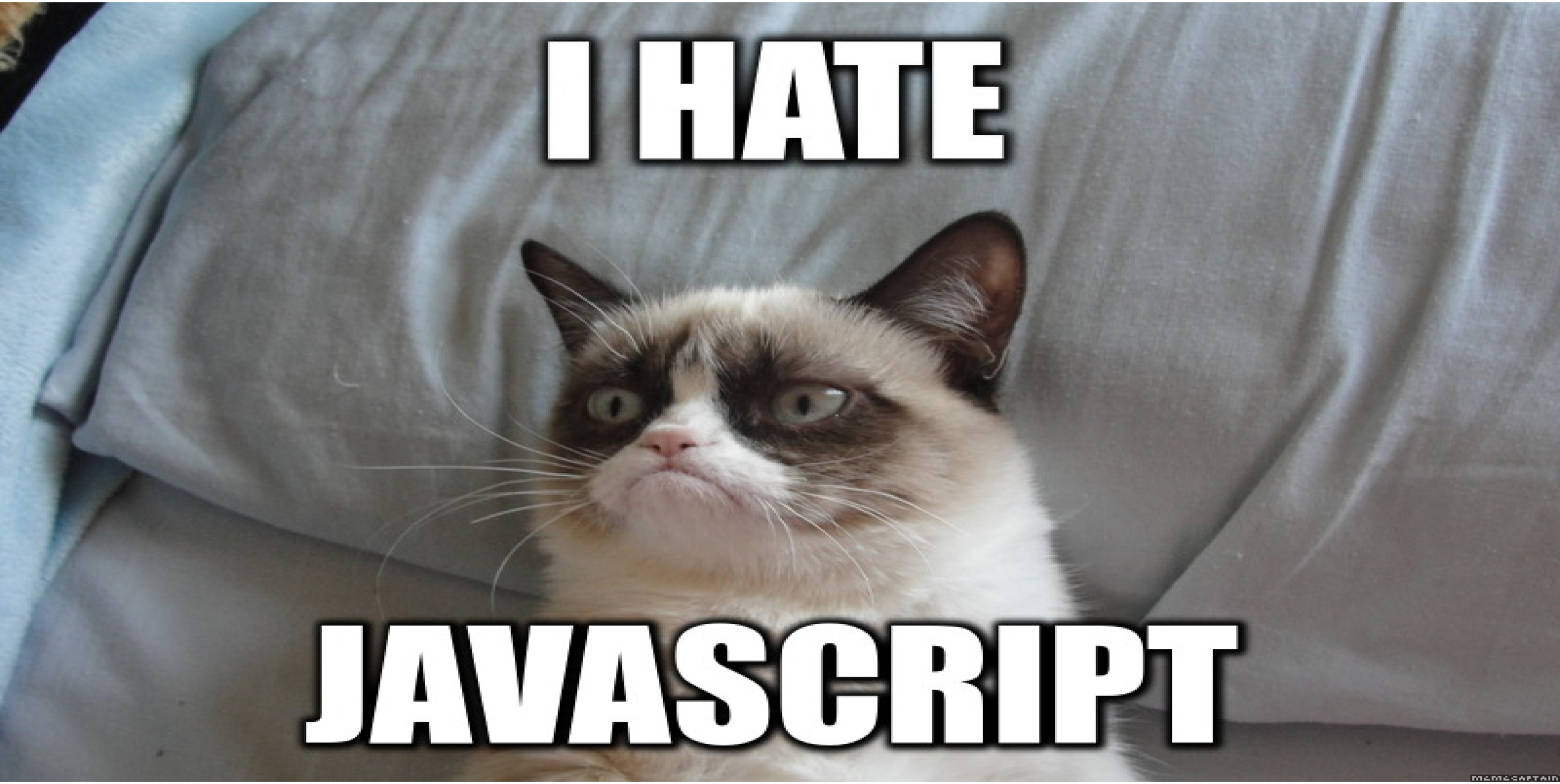
Branch: master New pull request New file Upload files Find file SSH git@github.com:html5/spec Download ZIP

msporny Added perma-location for todays Editors Draft for HTML5 Profiles. Latest commit 98bdd92 on Apr 17, 2010

bin	Fixed minor bugs when inserting bugs against status-of-this-document.	6 years ago
configs	Integrating changes from Larry Masinter into latest HTML+RDFa WD.	6 years ago
docs	Updated and released 2nd HTML+RDFa draft.	7 years ago
drafts/ED-html5-profiles-20040417	Added perma-location for todays Editors Draft for HTML5 Profiles.	6 years ago
headers	Fixed minor bugs when inserting bugs against status-of-this-document.	6 years ago
microsections	Added Infoset and DOM2 specifics to HTML5+RDFa draft.	6 years ago
setup	Added basic configure.ac script, Makefile.in and support scripts for	7 years ago
source	Added HTML5-profiles specification.	6 years ago
utils	Added Anolis and anolislib to the build utilities.	7 years ago
Makefile.in	Added HTML5-profiles specification.	6 years ago
README	Fixed small documentation error in README.	7 years ago

Anyway...

Lets see some code


I HATE
JAVASCRIPT

Hello World

- Create Phoenix project
- Install Phaser
- Create a game
- Update the HTML page
- Start the server
- Create a game state
- Add text "Hello world"
- Make it draggable

Create a new Phoenix project...

```
$ mix phoenix.new demo
...
$ cd demo
```

Install Phaser...

```
$ mkdir web/static/vendor/js/phaser
$ curl -L -o web/static/vendor/js/phaser/phaser
https://raw.githubusercontent.com/photonstorm/p
```

* Use vendor so Brunch doesn't pre-compile

Create a Game class...

```
export class Game extends Phaser.Game {

  // Initialize Phaser
  constructor(width, height, container) {
    super(width, height, Phaser.AUTO, container
  }

}
```

web/static/js/Game.js

Create new Game instance...

```
// Import dependencies
import {Game} from "./Game"

// Lets go!
new Game(700, 450, "phaser");
```

web/static/js/app.js

Set up the page template...

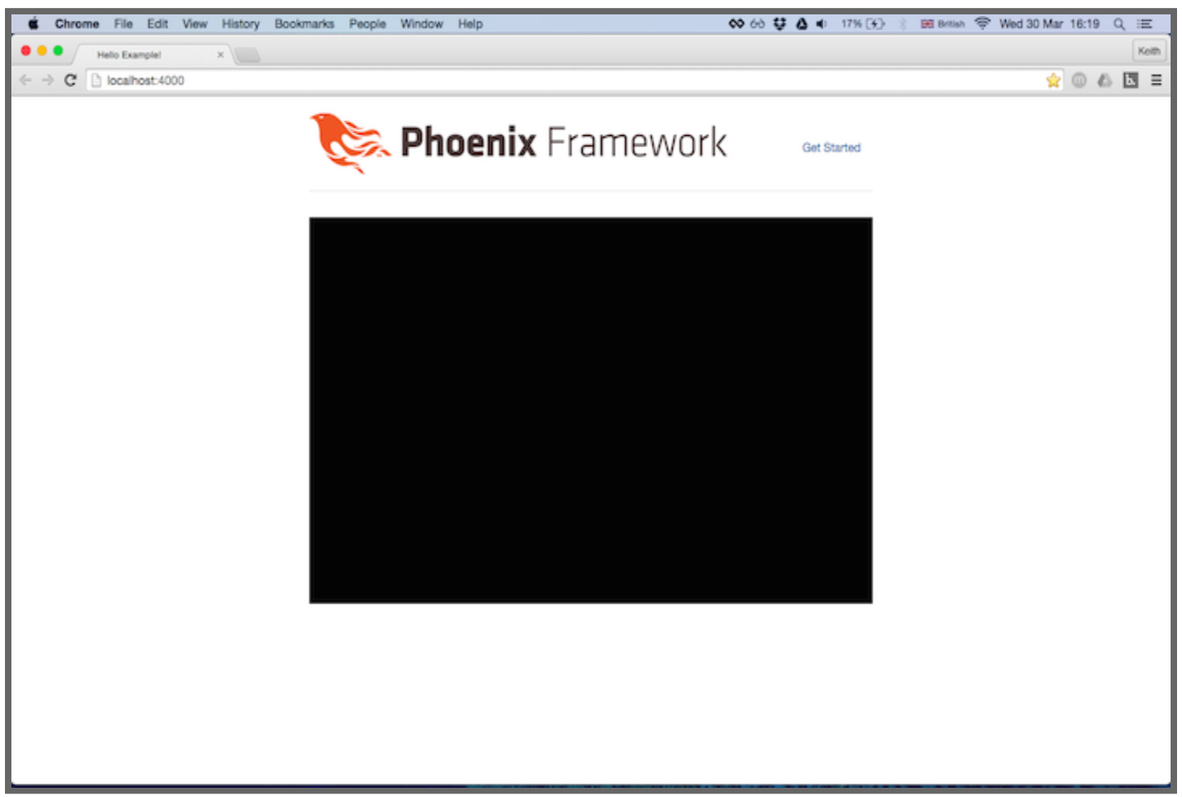
```
<div id="phaser"></div>
```

web/templates/page/index.html.eex

Start up the server...

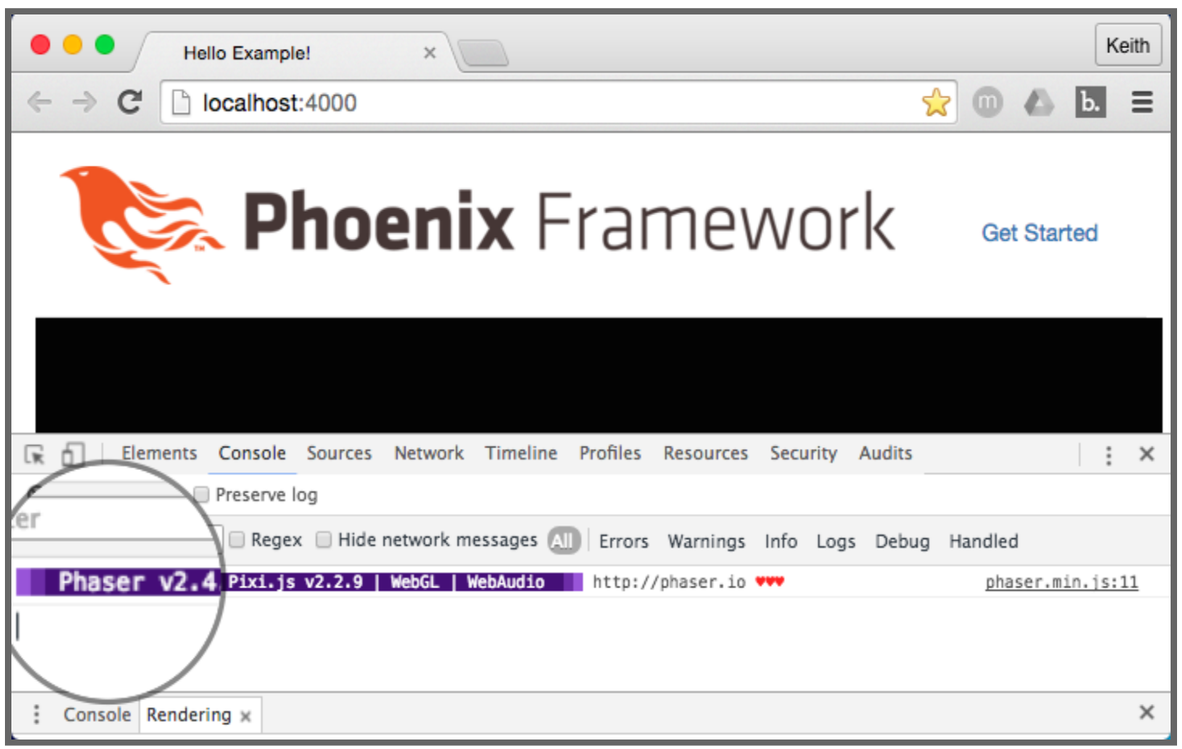
```
$ iex -S mix phoenix.server
```

Boom!



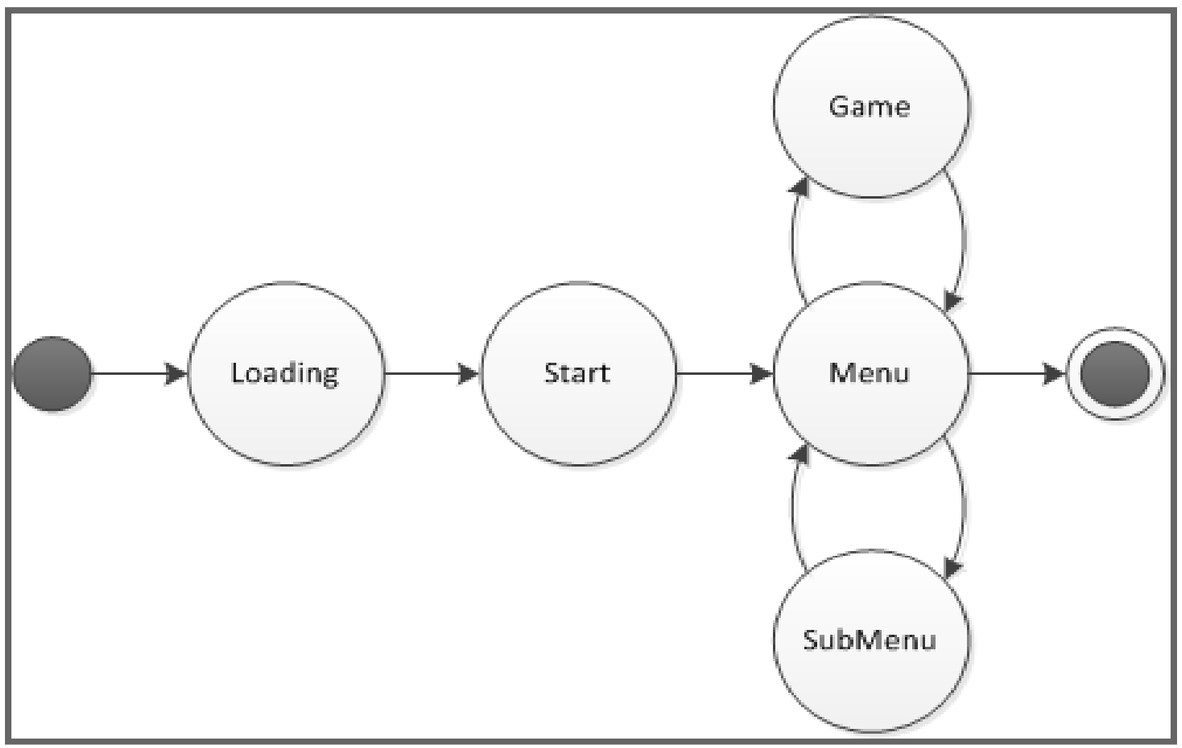
But wait?

Open the console



Where's my "Hello world"?

Game States



Using states allows you to break your game up into smaller pieces that can handle different mechanics of the game, such as a menu.

Create states directory

```
$ mkdir web/static/js/states
```

Create a new "State"

```
import { createLabel } from "../common/labels"

export class Lobby extends Phaser.State {
  create() {
    const label = createLabel(this, "Hello world")
    label.anchor.setTo(0.5)
  }
}
```

web/static/js/states/Lobby.js

Nothing up my sleeves

```
const DEFAULT_STYLE = {font: "65px Arial", fill: "#ffffff" }

// createLabel :: State -> String -> Object -> Sprite
export const createLabel = (state, message, style = DEFAULT_STYLE)

  const {centerX, centerY} = state.world
  return state.add.text(centerX, centerY, message, style)

}
```

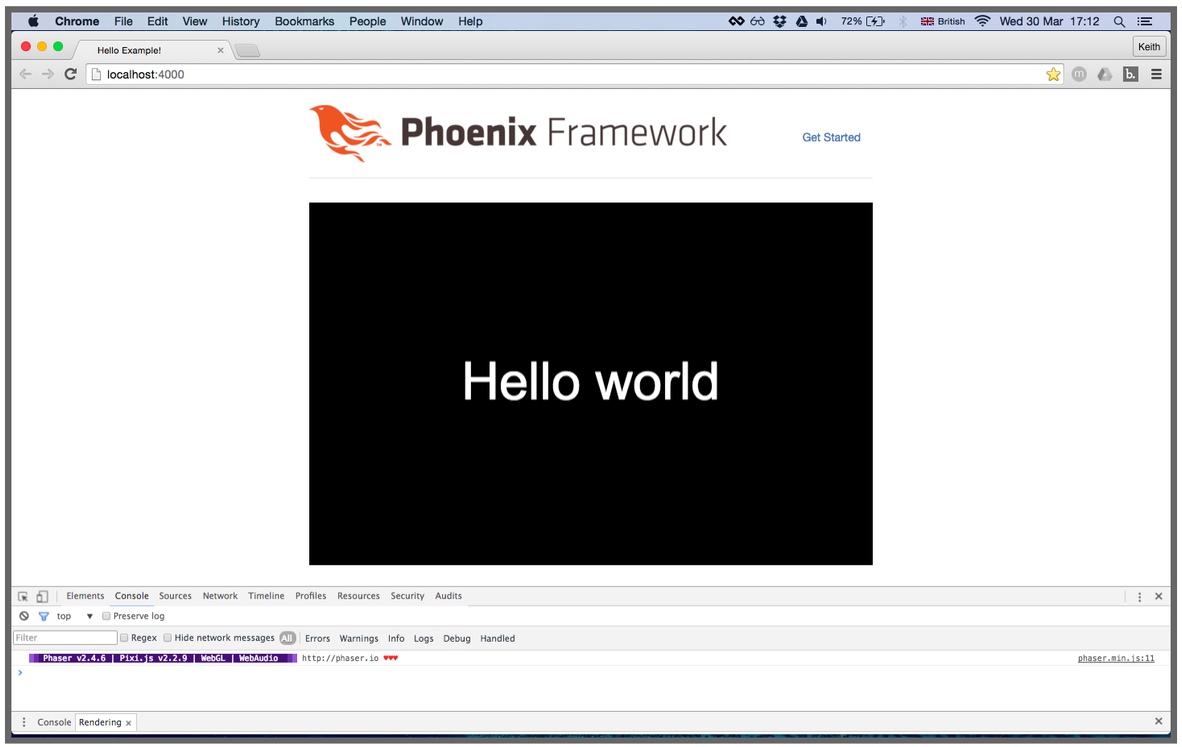
web/static/js/common/labels.js

Update the Game class

```
import {Lobby} from "./states/Lobby"

export class Game extends Phaser.Game {
  constructor(width, height, container) {
    super(width, height, Phaser.AUTO, container

    this.state.add("lobby", Lobby, false)

    this.state.start("lobby")
  }
}
```

web/static/js/Game.js

Yay!


Chrome File Edit View History Bookmarks People Window Help
72% British Wed 30 Mar 17:12
Hello Example!
Keith
localhost:4000
Phoenix Framework
Get Started
Hello world
Elements Console Sources Network Timeline Profiles Resources Security Audits
top
Preserve log
Filter
Regex
Hide network messages
All
Errors Warnings Info Logs Debug Handled
Phaser v2.4.6 | Pixi.js v2.2.9 | WebGL | WebAudio
http://phaser.io
phaser.min.js:11
Console
Rendering

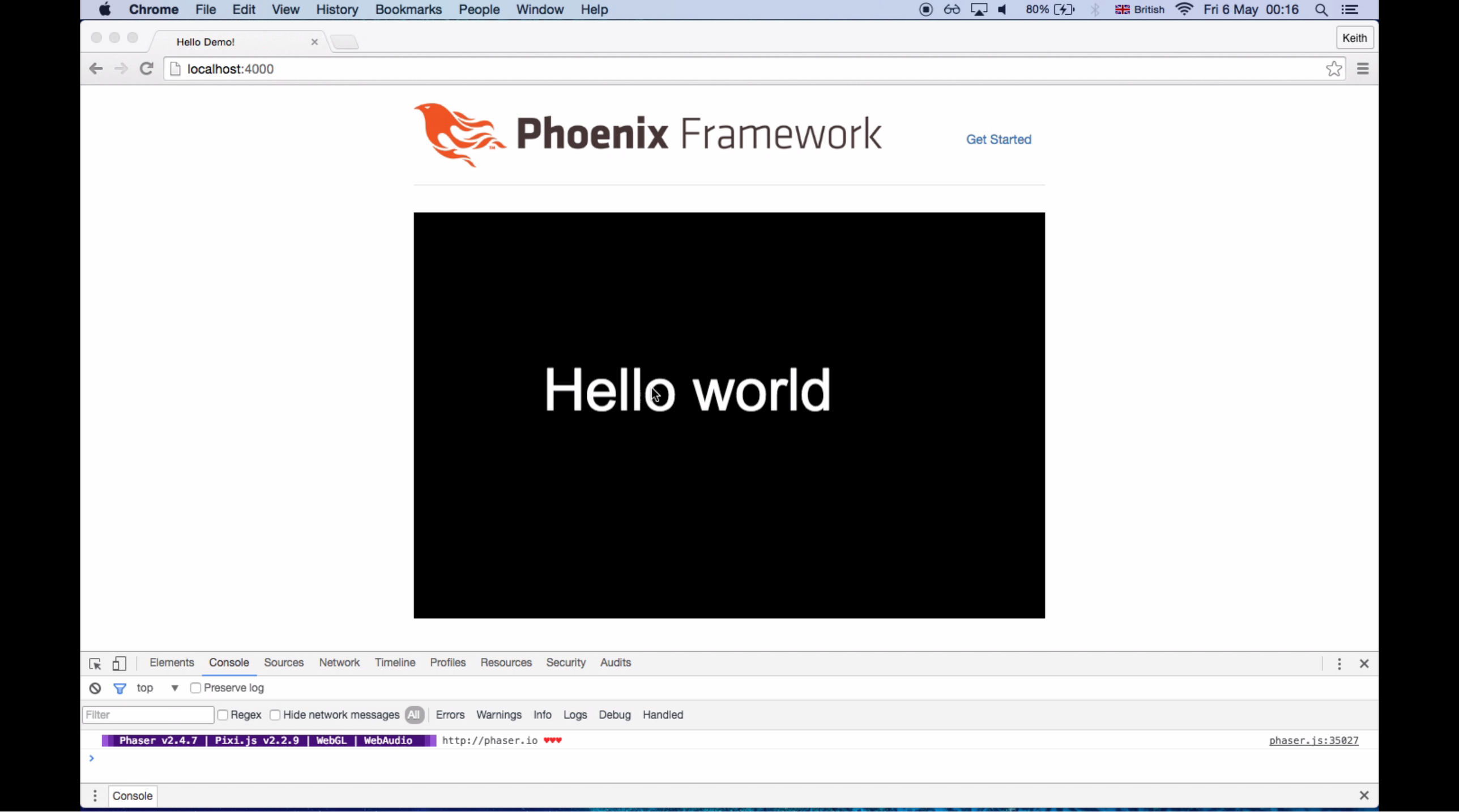
But wait?

We can do better

Lets make the text draggable

```
export class Lobby extends Phaser.State {
  create() {
    const label = createLabel(this, "Hello worl
    label.anchor.setTo(0.5)
    label.inputEnabled = true
    label.input.enableDrag()
  }
}
```

web/static/js/states/Lobby.js


Chrome File Edit View History Bookmarks People Window Help
80% British Fri 6 May 00:16
Hello Demo!
Keith
localhost:4000
Phoenix Framework
Get Started
Hello world
Elements Console Sources Network Timeline Profiles Resources Security Audits
top
Preserve log
Filter
Regex
Hide network messages
All
Errors Warnings Info Logs Debug Handled
Phaser v2.4.7 | Pixi.js v2.2.9 | WebGL | WebAudio
http://phaser.io
phaser.js:35027
Console

Whats next?

Syncronised text?

Because ¯_(ツ)_/¯

Phoenix Channels

Multiplexed on HTML5 WebSockets

Built in

Asynchronous

Create a new Phoenix channel

```
$ mix phoenix.gen.channel Lobby games
* creating web/channels/lobby_channel.ex
* creating test/channels/lobby_channel_test.exs

Add the channel to your `web/channels/user_socket.ex` handler, for

    channel "games:lobby", Demo.LobbyChannel
```

Phoenix creates this...

```
defmodule Demo.LobbyChannel do
  use Demo.Web, :channel

  def join("games:lobby", payload, socket) do
    if authorized?(payload) do
      {:ok, socket}
    else
      {:error, %{reason: "unauthorized"}}
    end
  end

  def handle_in("shout", payload, socket) do
    broadcast socket, "shout", payload
    {:noreply, socket}
  end
  ...
end
```

web/channels/lobby_channel.ex

Add the channel to the UserSocket

```
defmodule Demo.UserSocket do
  use Phoenix.Socket

  ## Channels
  channel "games:lobby", Demo.LobbyChannel
  ...
```

web/channels/user_socket.ex

Pass the socket to the game

```
// Import dependencies
import {Game} from "./Game"
import {Socket} from "phoenix"

const socket = new Socket("/socket", {})
const game = new Game(700, 450, "phaser")

// Lets go!
game.start(socket)
```

web/static/app.js

Connect the socket

```
export class Game extends Phaser.Game {

  constructor(width, height, container) {
    super(width, height, Phaser.AUTO, container, null)

    // set up game states
    this.state.add("lobby", Lobby, false)
  }

  start(socket) {
    socket.connect()
  }
```

web/static/js/Game.js

Create the channel

```
import {joinChannel} from "./common/channels"

export class Game extends Phaser.Game {
  ...
  start(socket) {
    socket.connect()

    // create and join the lobby channel
    const channel = socket.channel("games:lobby", {})

    joinChannel(channel, () => {
      console.log("Joined successfully")
      // start the lobby [name, clearWorld, clearCache, ...stateIni
      this.state.start("lobby", true, false, channel)
    })
  }
```

web/static/js/Game.js

Log the channel reference

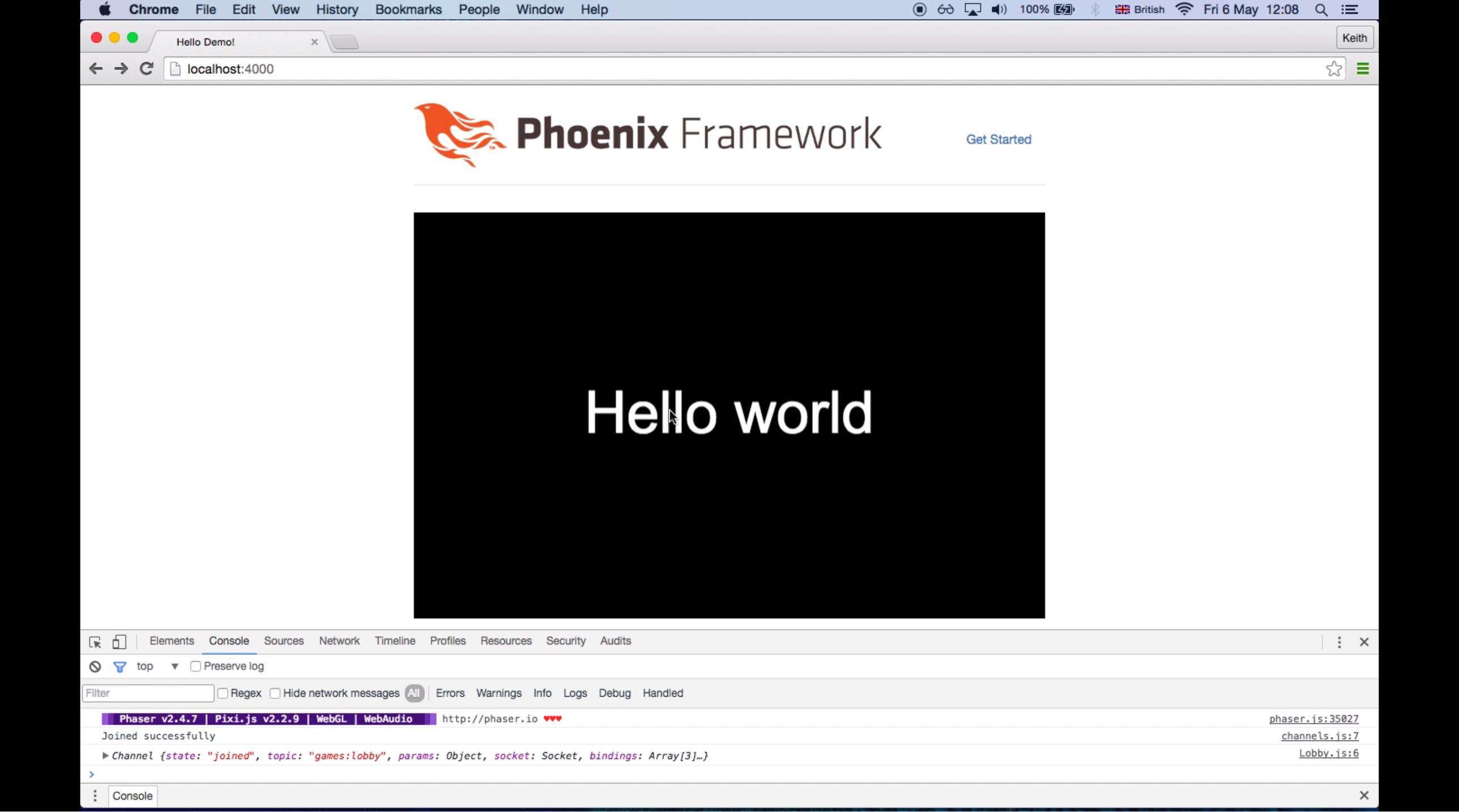
```
export class Lobby extends Phaser.State {
  init(...args) {
    const [channel] = args
    console.log(channel)
  }
```

web/static/js/Lobby.js

Inside channels.js

```
// joinChannel :: Channel -> Channel
export const joinChannel = (channel, success, failure, timeout) =>
  channel
    .join()
    .receive("ok", success || joinOk)
    .receive("error", failure || joinError)
    .receive("timeout", timeout || joinTimeout)
  return channel
}

// joinOk :: Response -> Console
const joinOk = (response) => console.log(`Joined successfully`, res

// joinError :: Response -> Console
const joinError = (response) => console.log(`Failed to join channel

// joinError :: Null -> Console
const joinTimeout = () => console.log("Networking issue. Still wait
```

web/static/js/common/channels.js


Chrome File Edit View History Bookmarks People Window Help
100% British Fri 6 May 12:08
Hello Demo!
Keith
localhost:4000
Phoenix Framework
Get Started
Hello world
Elements Console Sources Network Timeline Profiles Resources Security Audits
top
Preserve log
Filter
Regex
Hide network messages
All
Errors Warnings Info Logs Debug Handled
Phaser v2.4.7 | Pixi.js v2.2.9 | WebGL | WebAudio
http://phaser.io
phaser.js:35027
Joined successfully
channels.js:7
Channel {state: "joined", topic: "games:lobby", params: Object, socket: Socket, bindings: Array[3]…}
Lobby.js:6
Console

Ok nearly there

lets send the position when we stop dragging

Store the channel reference

```
export class Lobby extends Phaser.State {
  init(...options) {
    const [channel] = options
    this.channel = channel
  }
```

web/static/js/Lobby.js

Make the label sync

```
import { createSyncLabel } from "../common/sync_labels"

export class Lobby extends Phaser.State {
  ...
  create() {
    const label = createSyncLabel(this,
                                  "Hello world",
                                  this.channel)
  }

}
```

web/static/js/Lobby.js

And here's the sync label

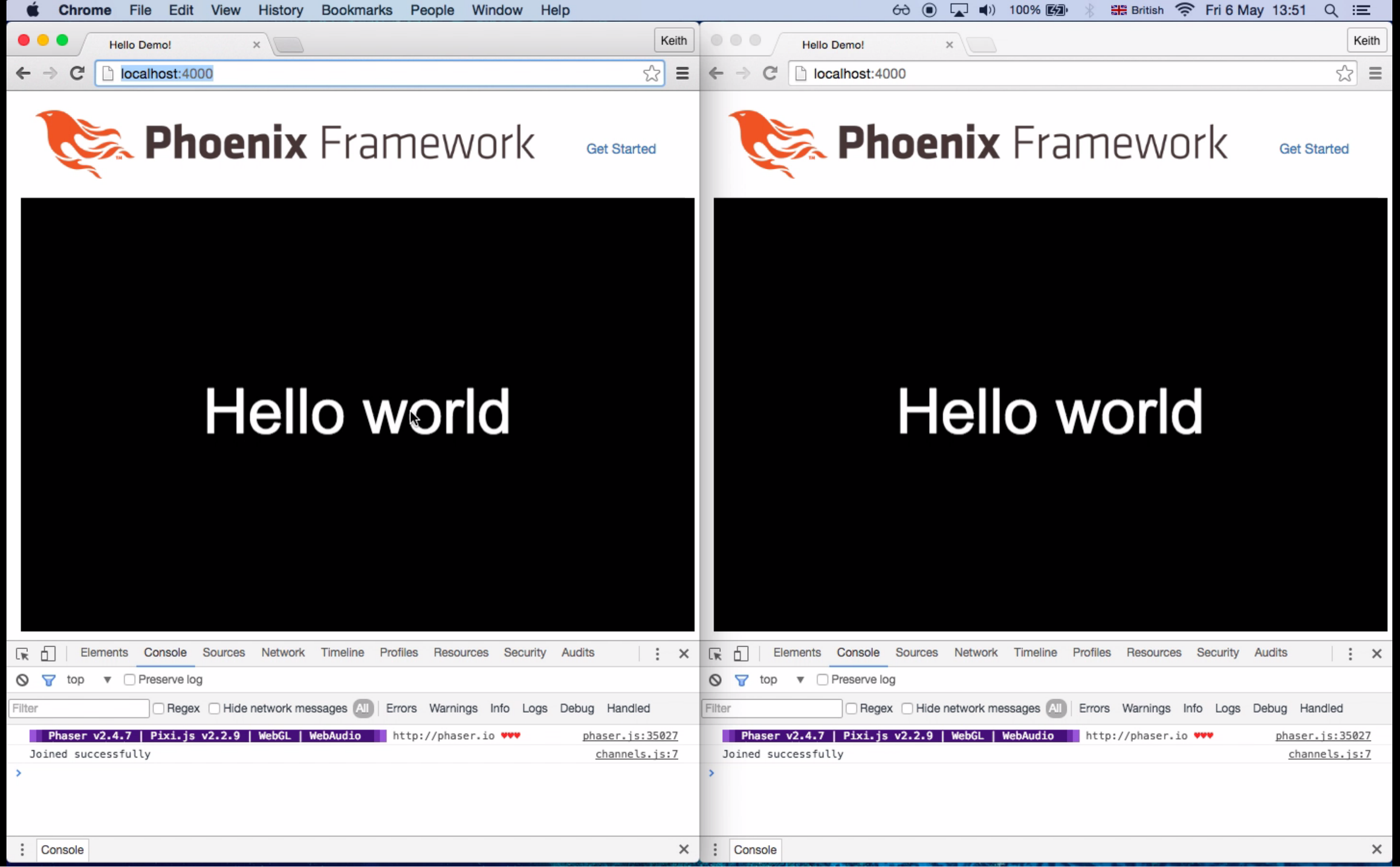
```
import { createLabel } from "./labels"
import { syncPosition } from "./sync"

export const createSyncLabel = (state, message, channel) => {
  const label = createLabel(state, message)
  label.anchor.setTo(0.5)
  label.inputEnabled = true
  label.input.enableDrag()

  // send message on drag stop [sprite, channel, event]
  syncPosition(label, channel, label.events.onDragStop)

  return label
}
```

web/static/js/common/sync_labels.js

And finally the sync position

```
// syncPosition :: Sprite -> Channel -> Event -> Function -> Event
export const syncPosition = (sprite, channel, event) => {
  event.add(sprite => sendPosition(sprite, channel))
}

// sendPosition :: Sprite -> Channel -> String
export const sendPosition = (sprite, channel) => {
  console.log(serializePosition(sprite))
}

// serializePosition :: Sprite -> Object
export const serializePosition = ({x, y}) => Object.assign({x, y})
```

web/static/js/common/sync.js


Chrome File Edit View History Bookmarks People Window Help
100% British Fri 6 May 13:51
Hello Demo!
Keith
localhost:4000
Phoenix Framework
Get Started
Hello world
Elements Console Sources Network Timeline Profiles Resources Security Audits
top
Preserve log
Filter
Regex
Hide network messages
All
Errors Warnings Info Logs Debug Handled
Phaser v2.4.7 | Pixi.js v2.2.9 | WebGL | WebAudio
http://phaser.io
phaser.js:35027
Joined successfully
channels.js:7
Console

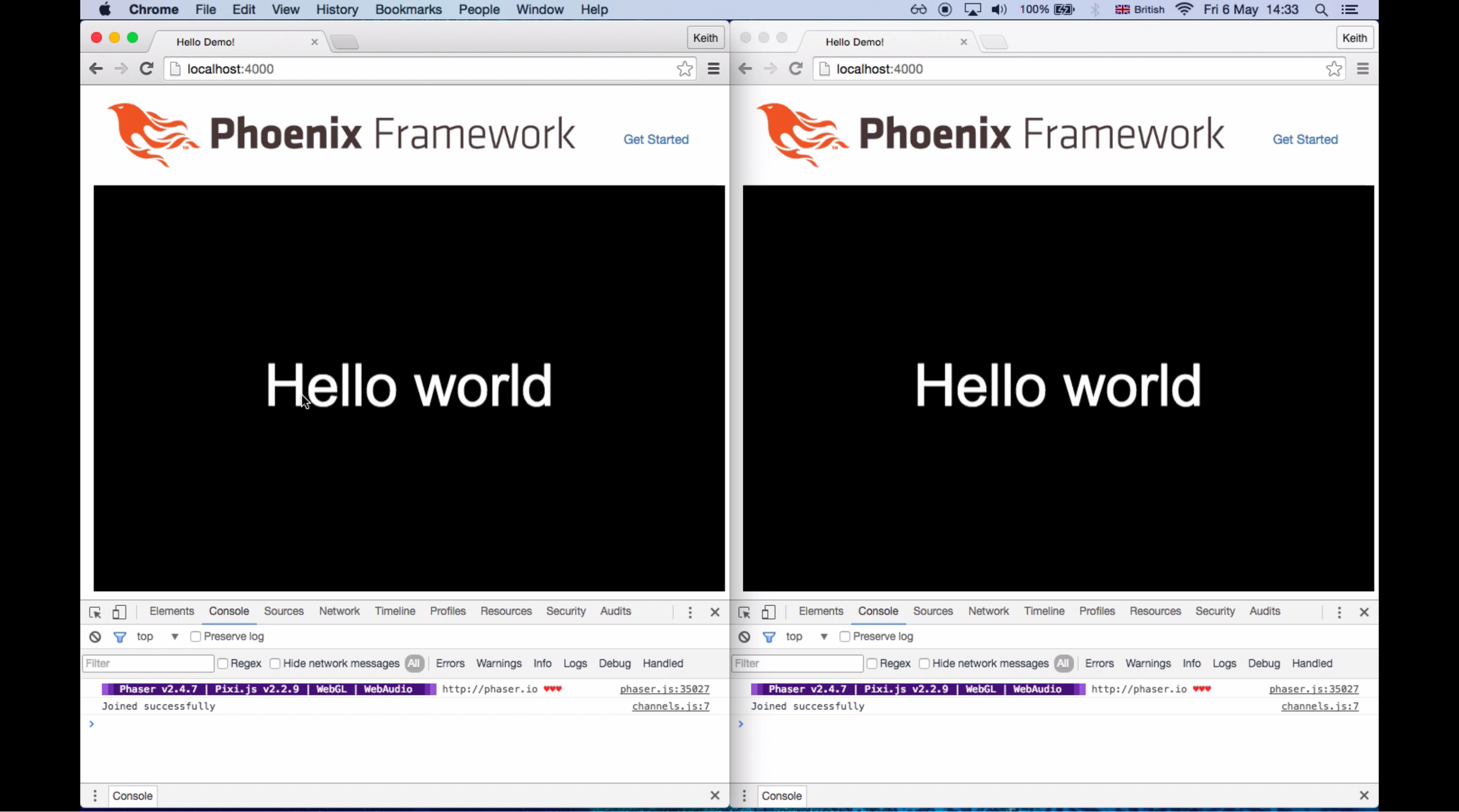
We're ready to

SEND the message

Send the message

```
// sendPosition :: Sprite -> Channel -> Push
export const sendPosition = (sprite, channel) => {
  const message = serializePosition(sprite)
  console.log("Sending message", message)
  channel.push("shout", message)
}
```

web/static/js/common/sync.js


Chrome File Edit View History Bookmarks People Window Help
100% British Fri 6 May 14:33
Hello Demo!
Keith
localhost:4000
Phoenix Framework
Get Started
Hello world
Elements Console Sources Network Timeline Profiles Resources Security Audits
top
Preserve log
Filter
Regex
Hide network messages
All
Errors Warnings Info Logs Debug Handled
Phaser v2.4.7 | Pixi.js v2.2.9 | WebGL | WebAudio
http://phaser.io
phaser.js:35027
Joined successfully
channels.js:7
Console

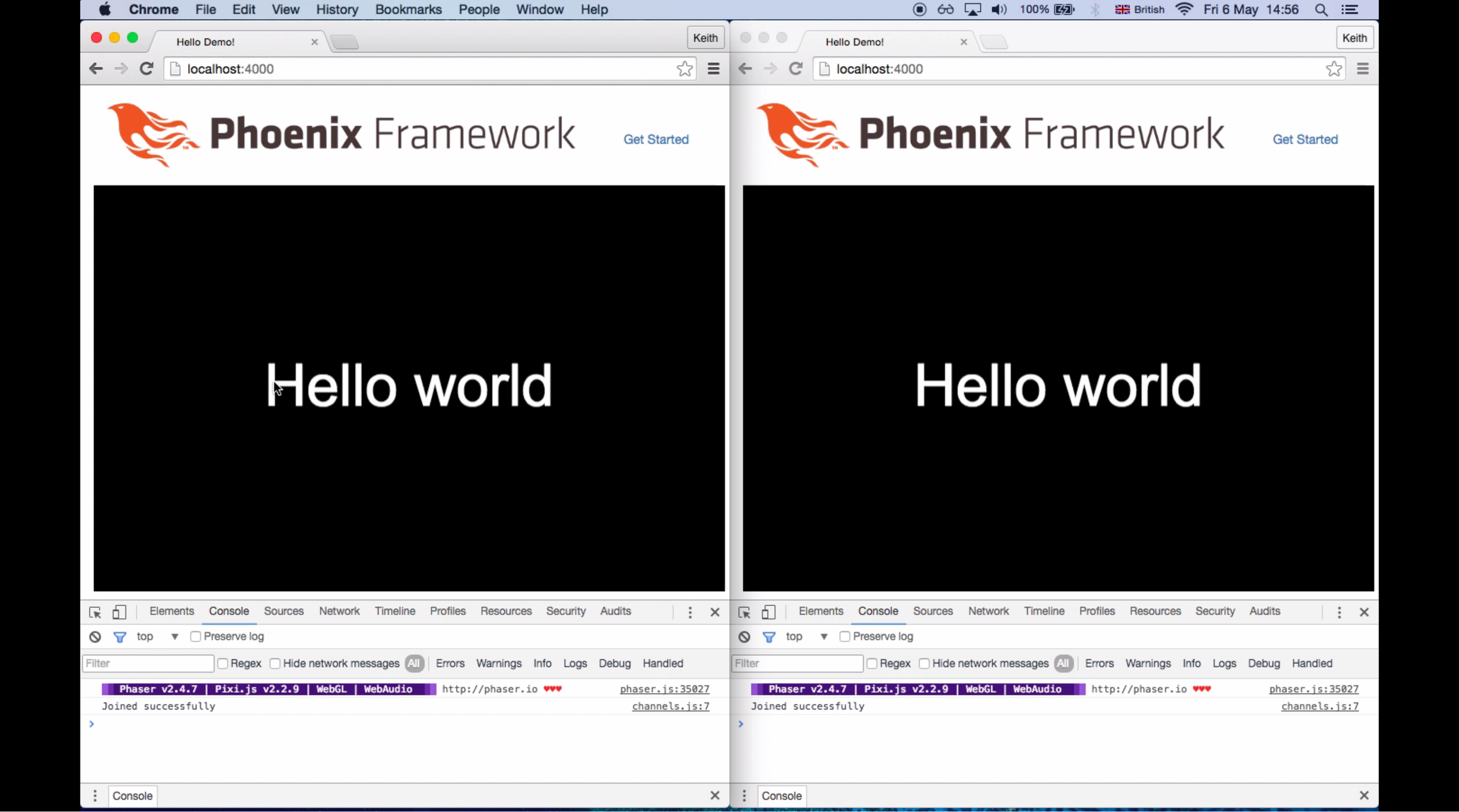
We're ready to

RECEIVE the message

Receive the message

```
// syncPosition :: Sprite -> Channel -> Event -> Function -> Event
export const syncPosition = (sprite, channel, event) => {
  event.add(sprite => sendPosition(sprite, channel))
  receivePosition(sprite, channel)
}

// receivePosition = Sprite -> Channel -> Push
export const receivePosition = (sprite, channel) => {
  channel.on("shout", (message) => {
    console.log("Received message", message)
  })
}
```

web/static/js/common/sync.js


Chrome File Edit View History Bookmarks People Window Help
100% British Fri 6 May 14:56
Hello Demo!
Keith
localhost:4000
Phoenix Framework
Get Started
Hello world
Elements Console Sources Network Timeline Profiles Resources Security Audits
top
Preserve log
Filter
Regex
Hide network messages
All
Errors Warnings Info Logs Debug Handled
Phaser v2.4.7 | Pixi.js v2.2.9 | WebGL | WebAudio
http://phaser.io
phaser.js:35027
Joined successfully
channels.js:7
Console

Remember this?

```
defmodule Demo.LobbyChannel do
  ...
  def handle_in("shout", payload, socket) do
    broadcast socket, "shout", payload
    {:noreply, socket}
  end
  ...
end
```

web/channels/lobby_channel.ex

broadcast_from

Lets make a position handler

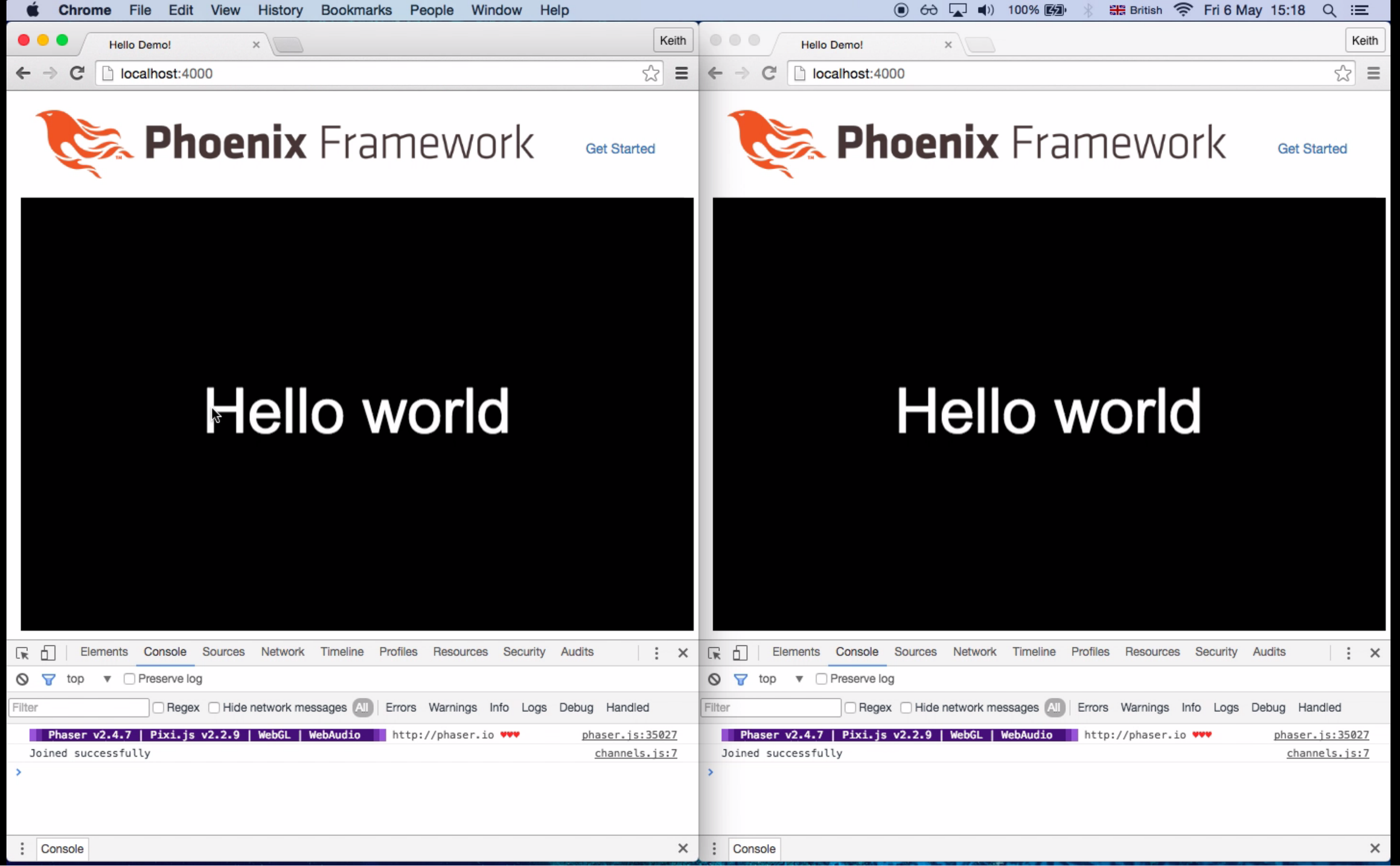
```
defmodule Demo.LobbyChannel do
  ...
  # broadcast position data to everyone else
  def handle_in("position", payload, socket) do
    broadcast_from socket, "position", payload
    {:noreply, socket}
  end
  ...
end
```

web/channels/lobby_channel.ex

Update the sync to use position

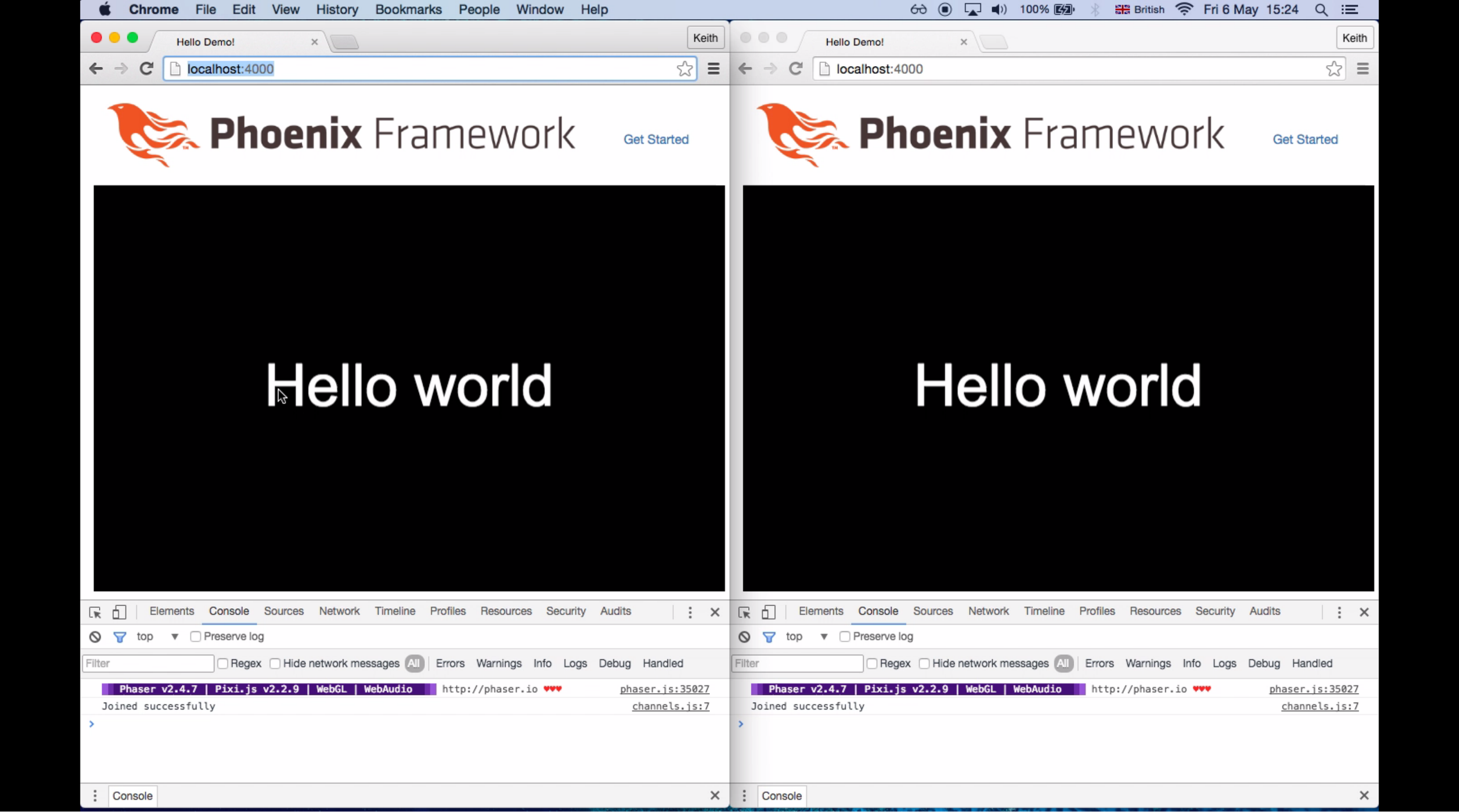
```
// sendPosition :: Sprite -> Channel -> Push
export const sendPosition = (sprite, channel) => {
  ...
  channel.push("position", message)
}

// receivePosition = Sprite -> Channel -> Push
export const receivePosition = (sprite, channel) => {
  ...
  channel.on("position", callback)
}
```

web/static/js/common/sync.js


Chrome File Edit View History Bookmarks People Window Help
100% British Fri 6 May 15:18
Hello Demo!
Keith
localhost:4000
Phoenix Framework
Get Started
Hello world
Elements Console Sources Network Timeline Profiles Resources Security Audits
top
Preserve log
Filter
Regex
Hide network messages
All
Errors Warnings Info Logs Debug Handled
Phaser v2.4.7 | Pixi.js v2.2.9 | WebGL | WebAudio
http://phaser.io
phaser.js:35027
Joined successfully
channels.js:7
Console
Hello Demo!
Keith
localhost:4000
Phoenix Framework
Get Started
Hello world
Elements Console Sources Network Timeline Profiles Resources Security Audits
top
Preserve log
Filter
Regex
Hide network messages
All
Errors Warnings Info Logs Debug Handled
Phaser v2.4.7 | Pixi.js v2.2.9 | WebGL | WebAudio
http://phaser.io
phaser.js:35027
Joined successfully
channels.js:7
Console

Remember to restart your server

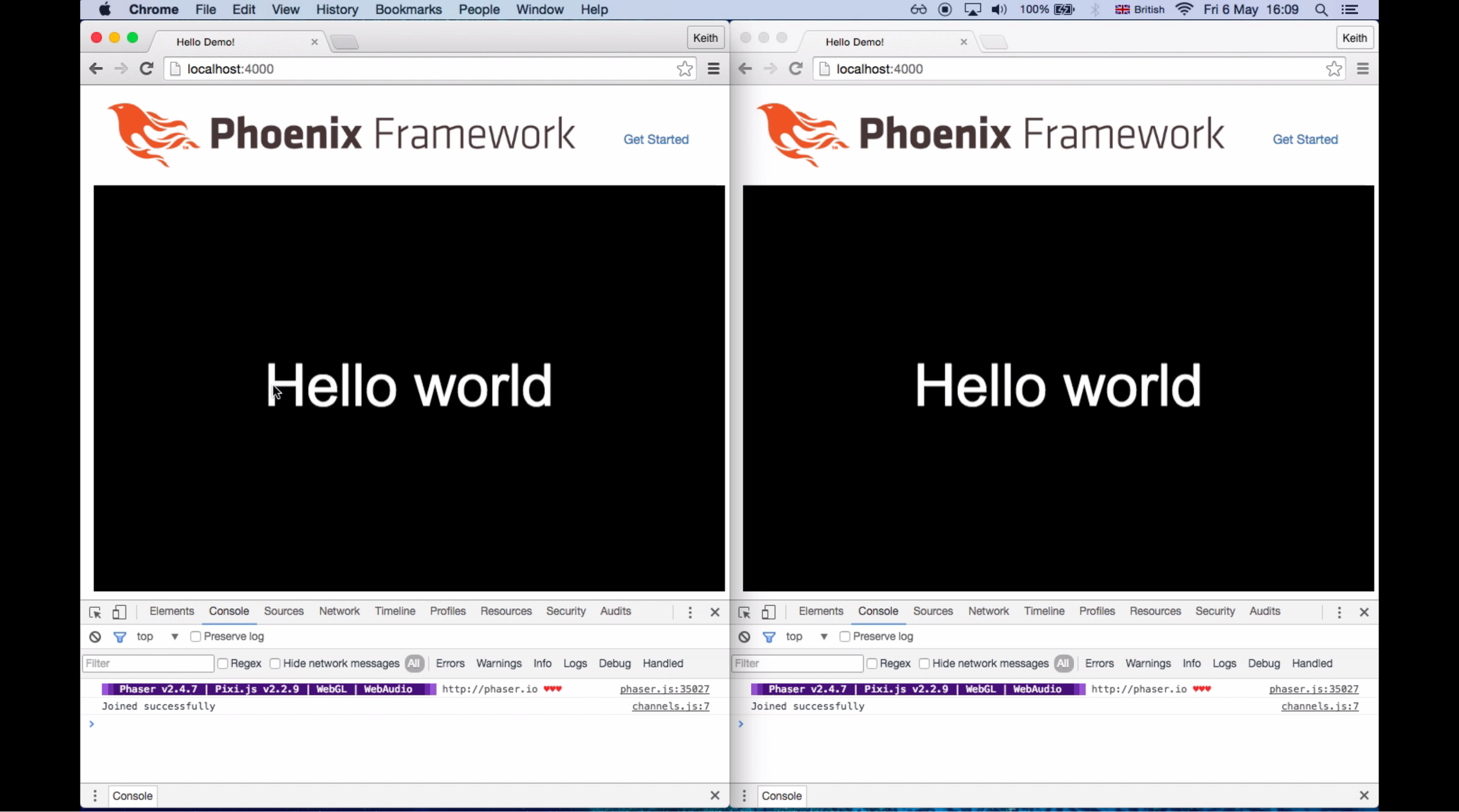
```
[error] GenServer #PID<0.413.0> terminating
** (FunctionClauseError) no function clause matching in Demo.LobbyC
    (demo) web/channels/lobby_channel.ex:20:
    Demo.GameChannel.handle_in("position", %{"x" => 249, "y" => 109
    %Phoenix.Socket{assigns: %{}, channel: Demo.LobbyChannel, chann
    #PID<0.413.0>, endpoint: Demo.Endpoint, handler: Demo.UserSocke
    joined: true, pubsub_server: Demo.PubSub, ref: "10", serializer
    Phoenix.Transports.WebSocketSerializer, topic: "games:lobby", t
    Phoenix.Transports.WebSocket, transport_name: :websocket, trans
    #PID<0.396.0>})
```


Chrome File Edit View History Bookmarks People Window Help
100% British Fri 6 May 15:24
Hello Demo!
Keith
localhost:4000
Phoenix Framework
Get Started
Hello world
Elements Console Sources Network Timeline Profiles Resources Security Audits
top
Preserve log
Filter
Regex
Hide network messages
All
Errors Warnings Info Logs Debug Handled
Phaser v2.4.7 | Pixi.js v2.2.9 | WebGL | WebAudio
http://phaser.io
phaser.js:35027
Joined successfully
channels.js:7
Console

Now finally we can update the sprite position!

```
// receivePosition = Sprite -> Channel -> Push
export const receivePosition = (sprite, channel) => {
  const callback = (message) => {
    console.log("Received message", message)
    const {x,y} = message
    sprite.position.setTo(x, y)
  }
  channel.on("position", callback)
}
```

web/static/js/common/sync.js


Chrome File Edit View History Bookmarks People Window Help
100% British Fri 6 May 16:09
Hello Demo!
Keith
localhost:4000
Phoenix Framework
Get Started
Hello world
Elements Console Sources Network Timeline Profiles Resources Security Audits
top
Preserve log
Filter
Regex
Hide network messages
All
Errors Warnings Info Logs Debug Handled
Phaser v2.4.7 | Pixi.js v2.2.9 | WebGL | WebAudio
http://phaser.io
phaser.js:35027
Joined successfully
channels.js:7
Console
Hello Demo!
Keith
localhost:4000
Phoenix Framework
Get Started
Hello world
Elements Console Sources Network Timeline Profiles Resources Security Audits
top
Preserve log
Filter
Regex
Hide network messages
All
Errors Warnings Info Logs Debug Handled
Phaser v2.4.7 | Pixi.js v2.2.9 | WebGL | WebAudio
http://phaser.io
phaser.js:35027
Joined successfully
channels.js:7
Console

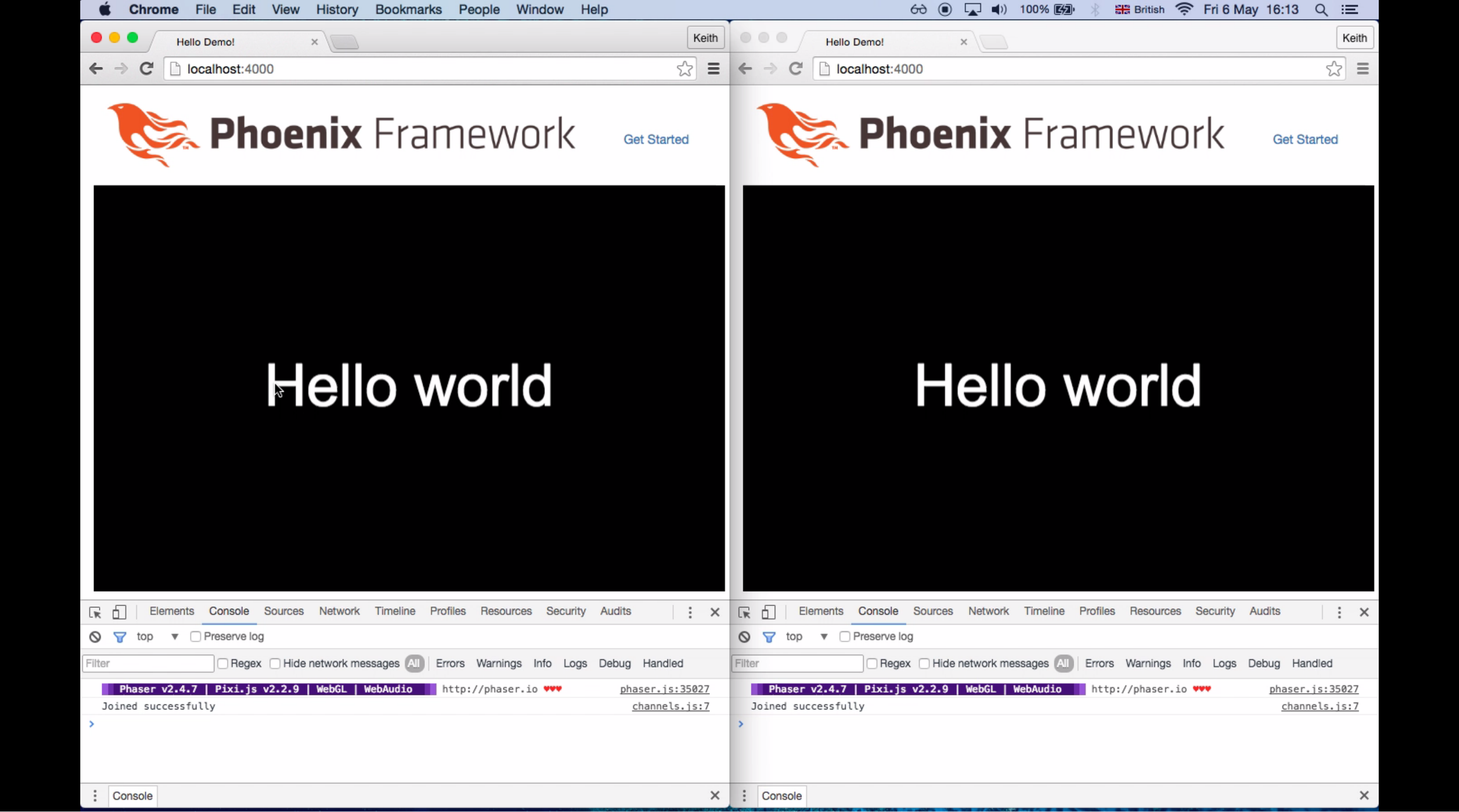
But wait?

We can do better

Use the onDragUpdate event

```
export const createSyncLabel = (state, message, channel) => {
  ...

  syncPosition(label, channel, label.events.onDragUpdate)

  return label
}
```

web/static/js/Lobby.js


Chrome File Edit View History Bookmarks People Window Help
100% British Fri 6 May 16:13
Hello Demo!
Keith
localhost:4000
Phoenix Framework
Get Started
Hello world
Elements Console Sources Network Timeline Profiles Resources Security Audits
top
Preserve log
Filter
Regex
Hide network messages
All
Errors Warnings Info Logs Debug Handled
Phaser v2.4.7 | Pixi.js v2.2.9 | WebGL | WebAudio
http://phaser.io
phaser.js:35027
Joined successfully
channels.js:7
Console
Hello Demo!
Keith
localhost:4000
Phoenix Framework
Get Started
Hello world
Elements Console Sources Network Timeline Profiles Resources Security Audits
top
Preserve log
Filter
Regex
Hide network messages
All
Errors Warnings Info Logs Debug Handled
Phaser v2.4.7 | Pixi.js v2.2.9 | WebGL | WebAudio
http://phaser.io
phaser.js:35027
Joined successfully
channels.js:7
Console

Lets add some juice!

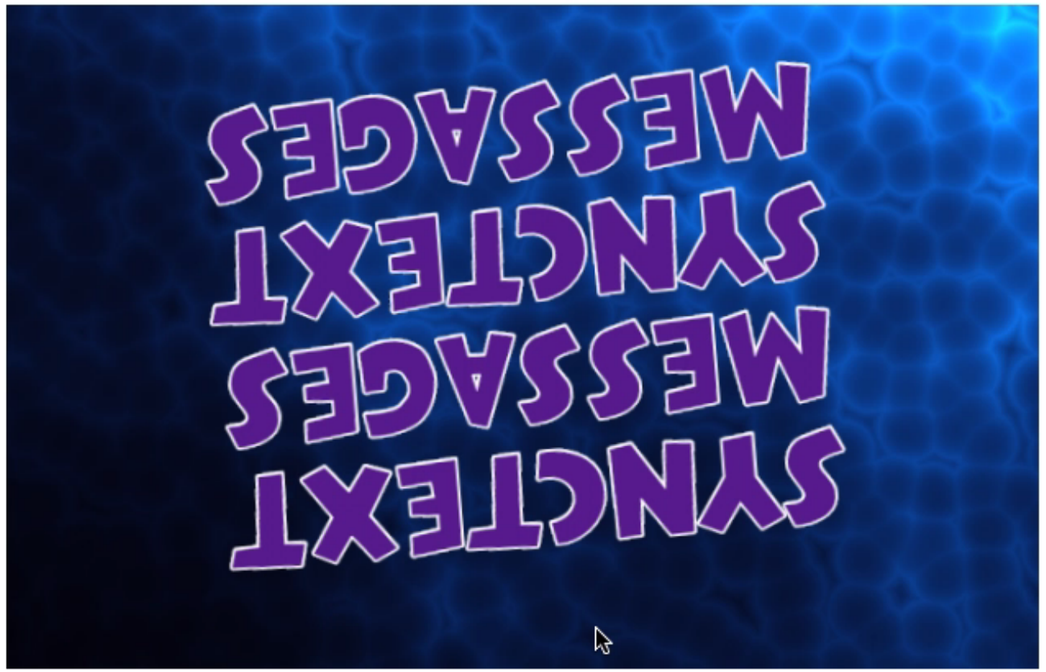
Hello PhaserModules!

Keith

localhost:4000

Phoenix Framework

Get Started



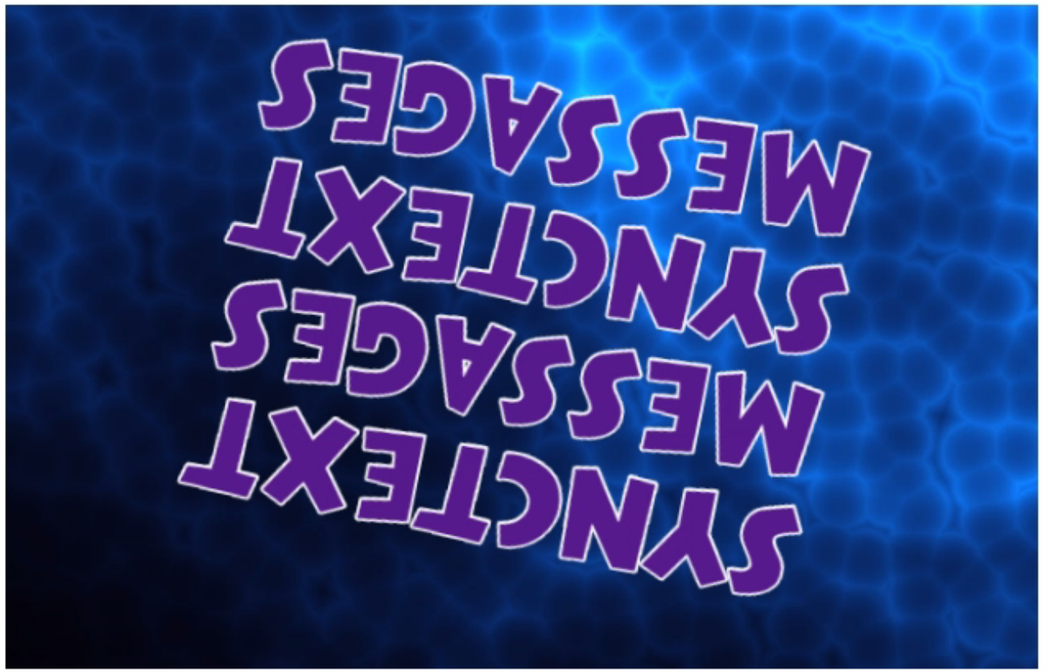
Hello PhaserModules!

Keith

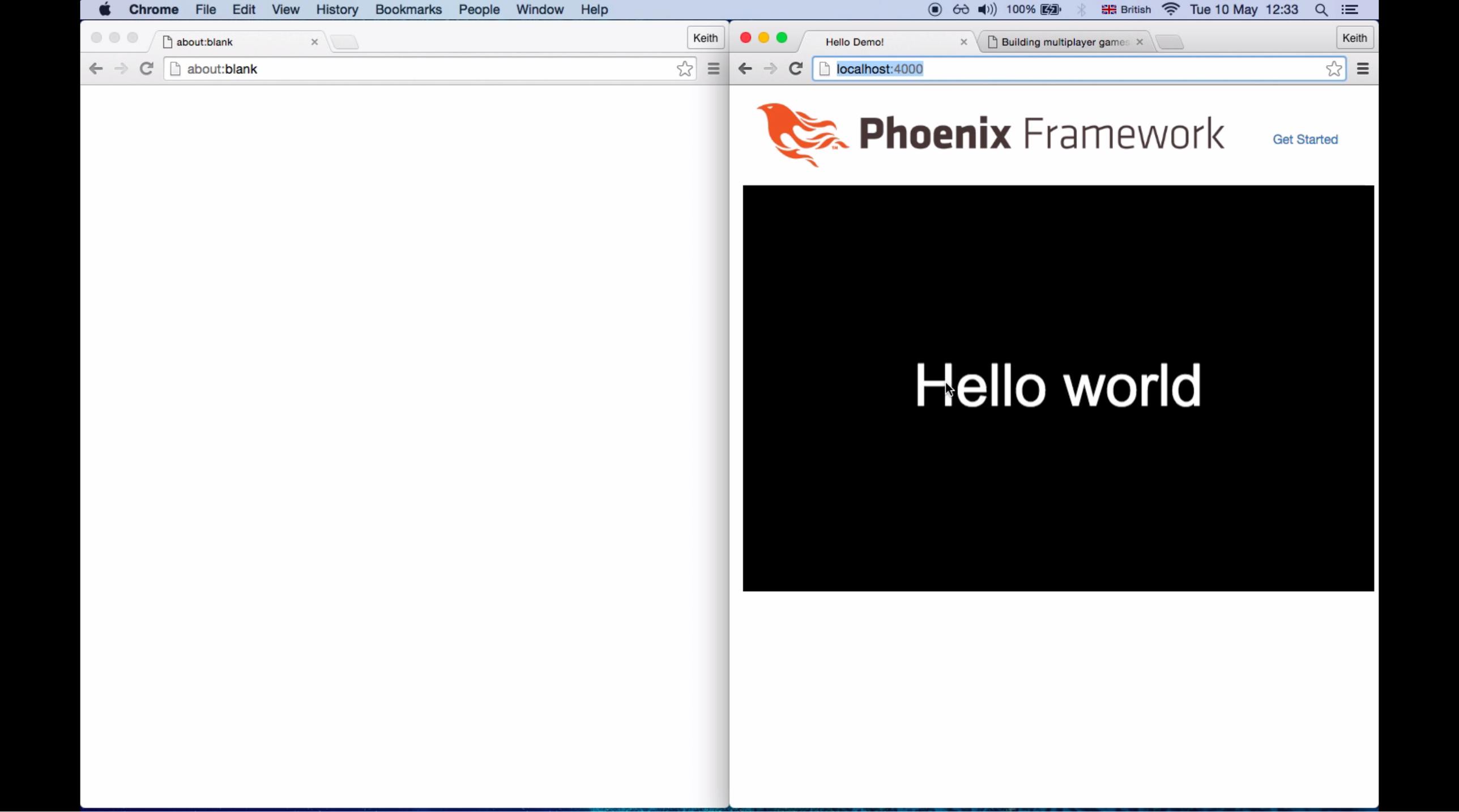
localhost:4000

Phoenix Framework

Get Started



Clientside state problem


Chrome File Edit View History Bookmarks People Window Help
100% British Tue 10 May 12:33
about:blank
Keith
about:blank
Hello Demo!
Building multiplayer games
Keith
localhost:4000
Phoenix Framework
Get Started
Hello world

Serverside State

Use a Genserver

But this is Erlang?

Lets use OTP

Supervision Trees

GitHub, Inc. [US] https://github.com/ktec/phoenixphaserdemo/

This repository Search Pull requests Issues Gist

ktec / phoenixphaserdemo

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

Building games with Phoenix and Phaser demo — Edit

19 commits 6 branches 0 releases 1 contributor

Branch: master New pull request

New file Upload files Find file SSH git@github.com:ktec/phoen Download ZIP

Switch branches/tags

Find or create a branch...

Branches Tags

01_synchronised_text

02_uuid_user_token

03_multiple_channels

04_supervision_tree

05_rocks

master

Latest commit fe87881 4 days ago

	al commit with phoenix new (v1.1.4)	5 days ago
	al commit with phoenix new (v1.1.4)	5 days ago
	al commit with phoenix new (v1.1.4)	5 days ago
	gen channel	19 hours ago
	nge to drag update event	12 hours ago
	al commit with phoenix new (v1.1.4)	5 days ago
	al commit with phoenix new (v1.1.4)	5 days ago
	al commit with phoenix new (v1.1.4)	5 days ago
mix.exs	initial commit with phoenix new (v1.1.4)	5 days ago
mix.lock	initial commit with phoenix new (v1.1.4)	5 days ago
package.json	initial commit with phoenix new (v1.1.4)	5 days ago

README.md

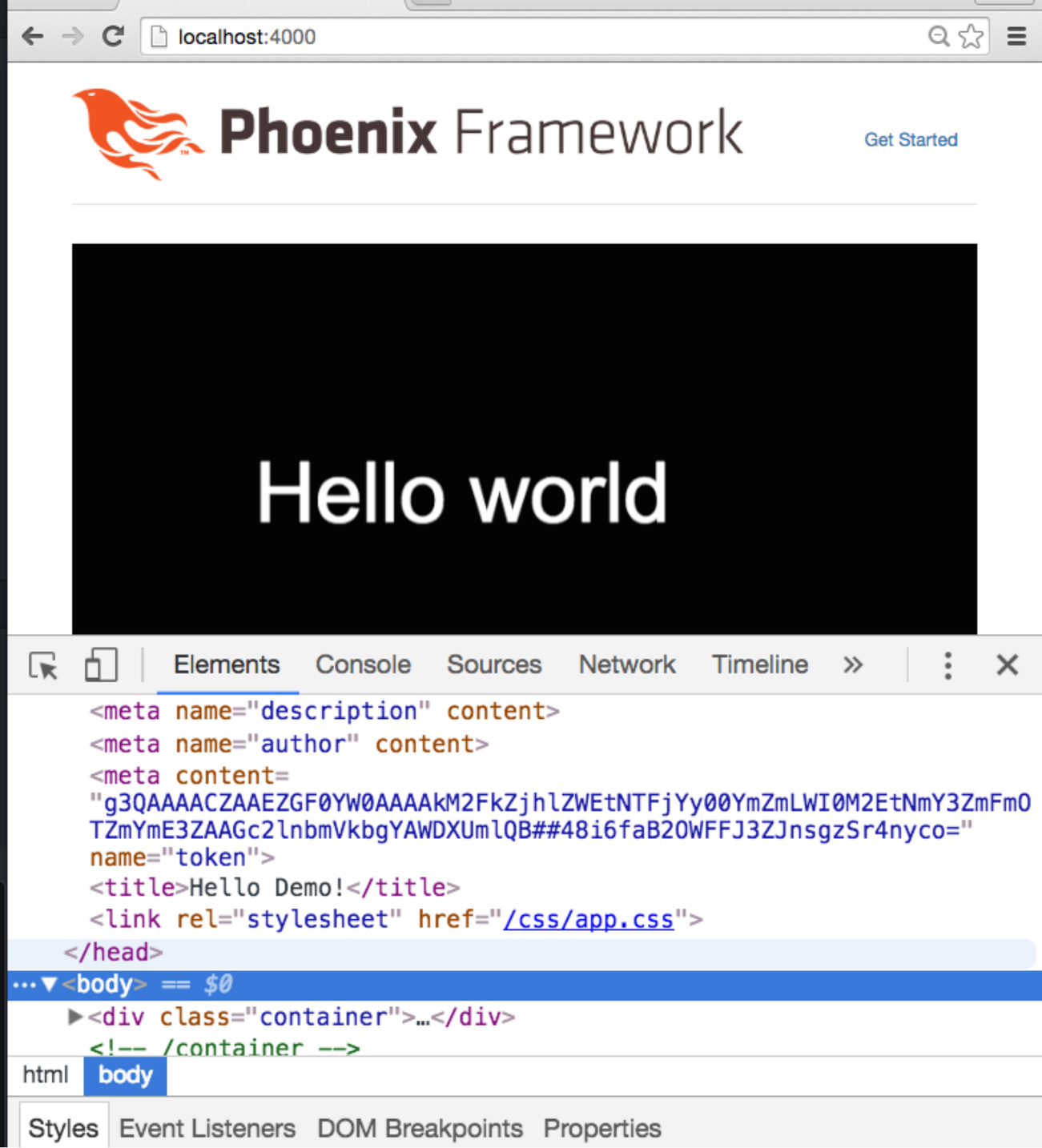
Identify each user using token

app.html.eex

```
    <meta http-equiv="X-UA-Compatible"
    content="IE=edge">
    <meta name="viewport" content="width=device-width,
    initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <%= tag(:meta, name: "token", content: @token) %>

    <title>Hello Demo!</title>
    <link rel="stylesheet" href="<%=
    static_path(@conn, "/css/app.css") %>">
  </head>

  <body>
```

app.js

```
import {socket} from "phoenix"

const token =
document.head.querySelector("[name=token]").content
const socket = new Socket("/socket", {
  params: {token: token},
  // logger: (kind, msg, data) => {
  console.log(`${kind}: ${msg}`, data) }
})
const game = new Game(700, 450, "phaser")

// Lets go!
game.start(socket)
```



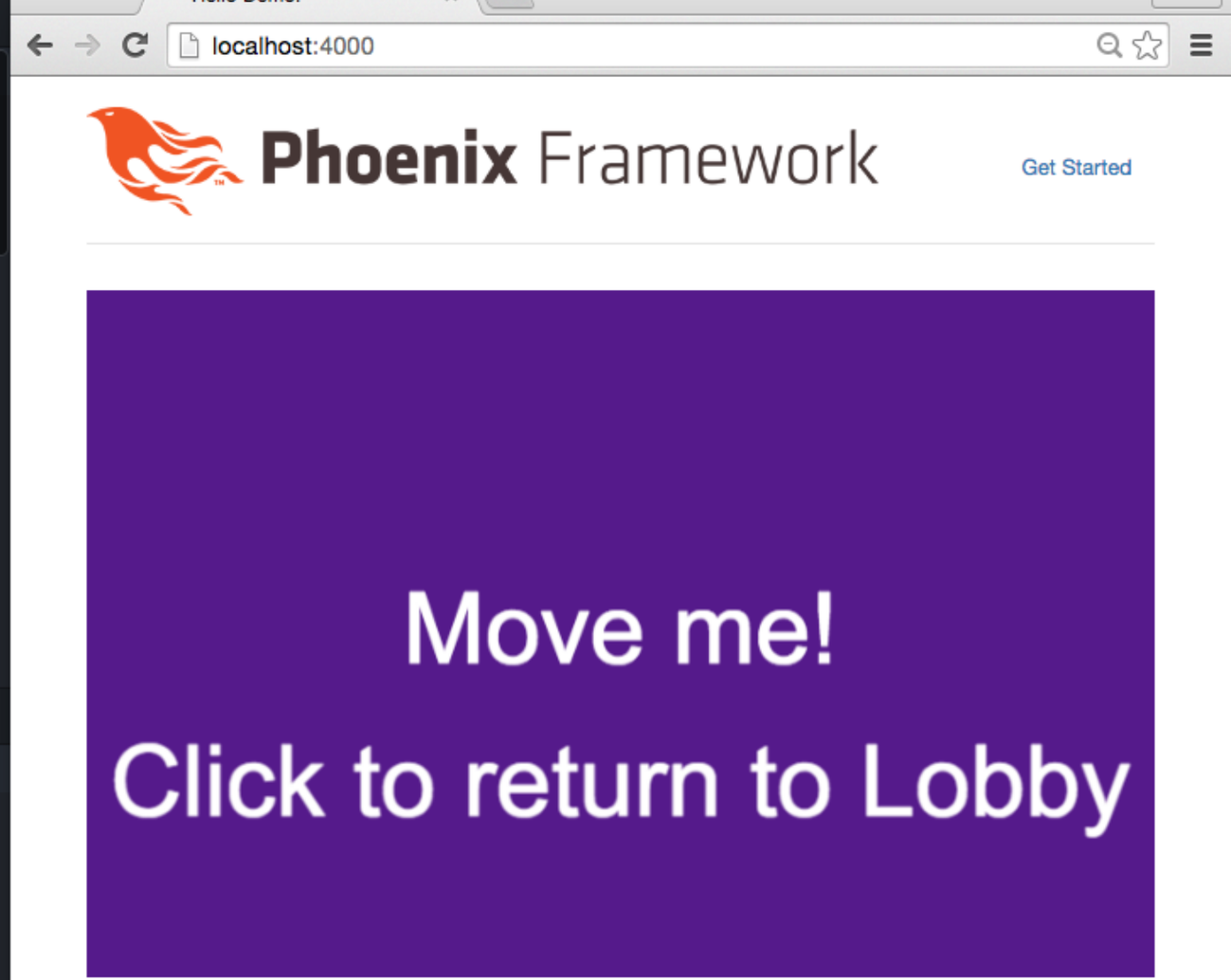
Muliple channels and Game states

play_channel.ex

```elixir
defmodule Demo.PlayChannel do
  use Demo.Web, :channel
  require Logger

  def join("games:play", payload, socket) do
    if authorized?(payload) do
      Logger.debug "#{socket.assigns.user_id} joined the Play channel"
      {:ok, socket}
    else
      {:error, %{reason: "unauthorized"}}
    end
  end
```

play_channel.ex | Play.js

```javascript
import { createLabel } from "../common/labels"
import { createSyncLabel } from "../common/sync_labels"
import { leaveChannel } from "../common/channels"

export class Play extends Phaser.State {
  create(game) {
    game.stage.backgroundColor = 0x551A8B

    const label = createSyncLabel(this, "Move me!", this.channel, "1")

    const label2 = createLabel(this, "Click to return
```



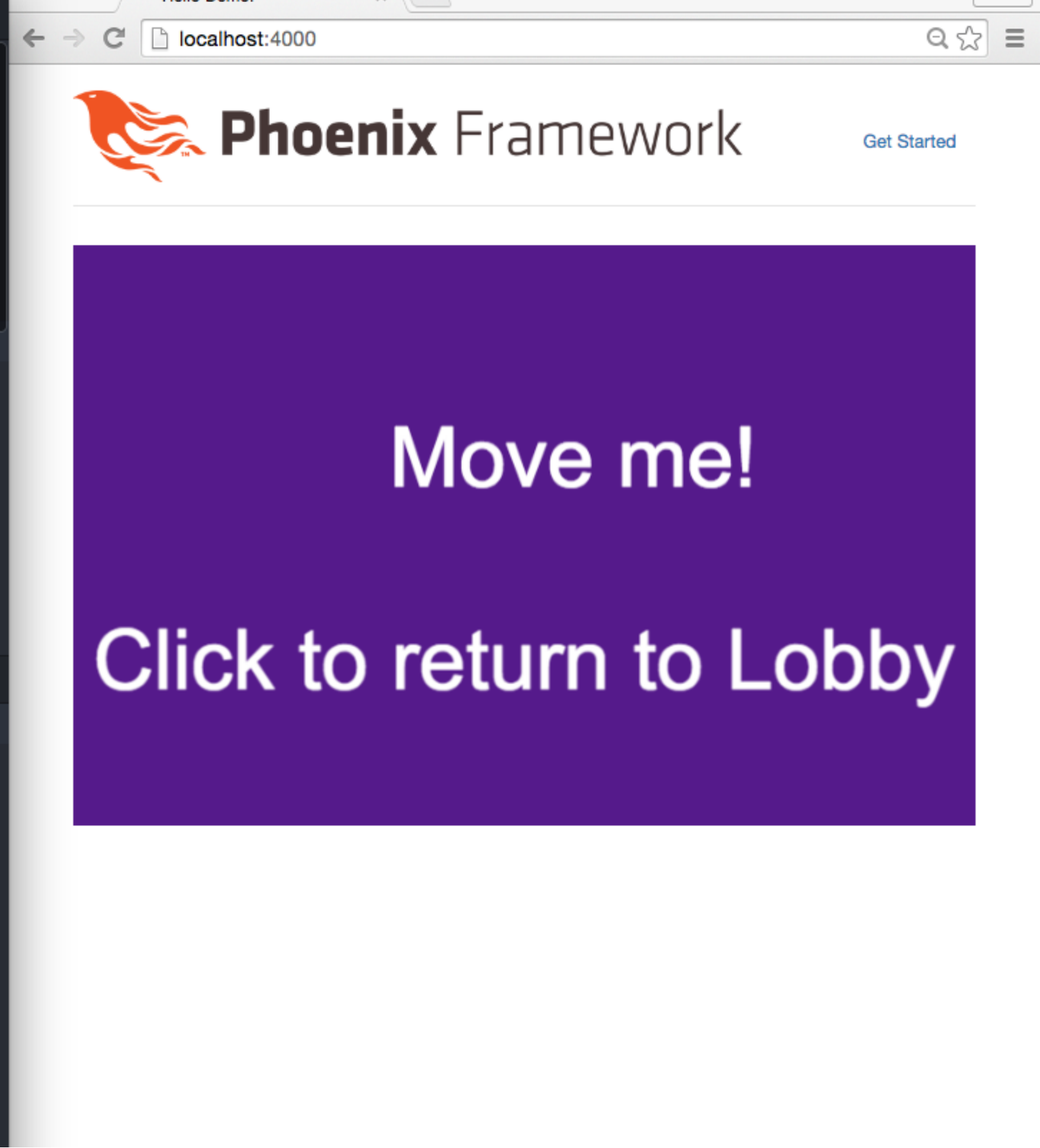
Dynamic Processes and Supervision Trees

player_supervisor.ex

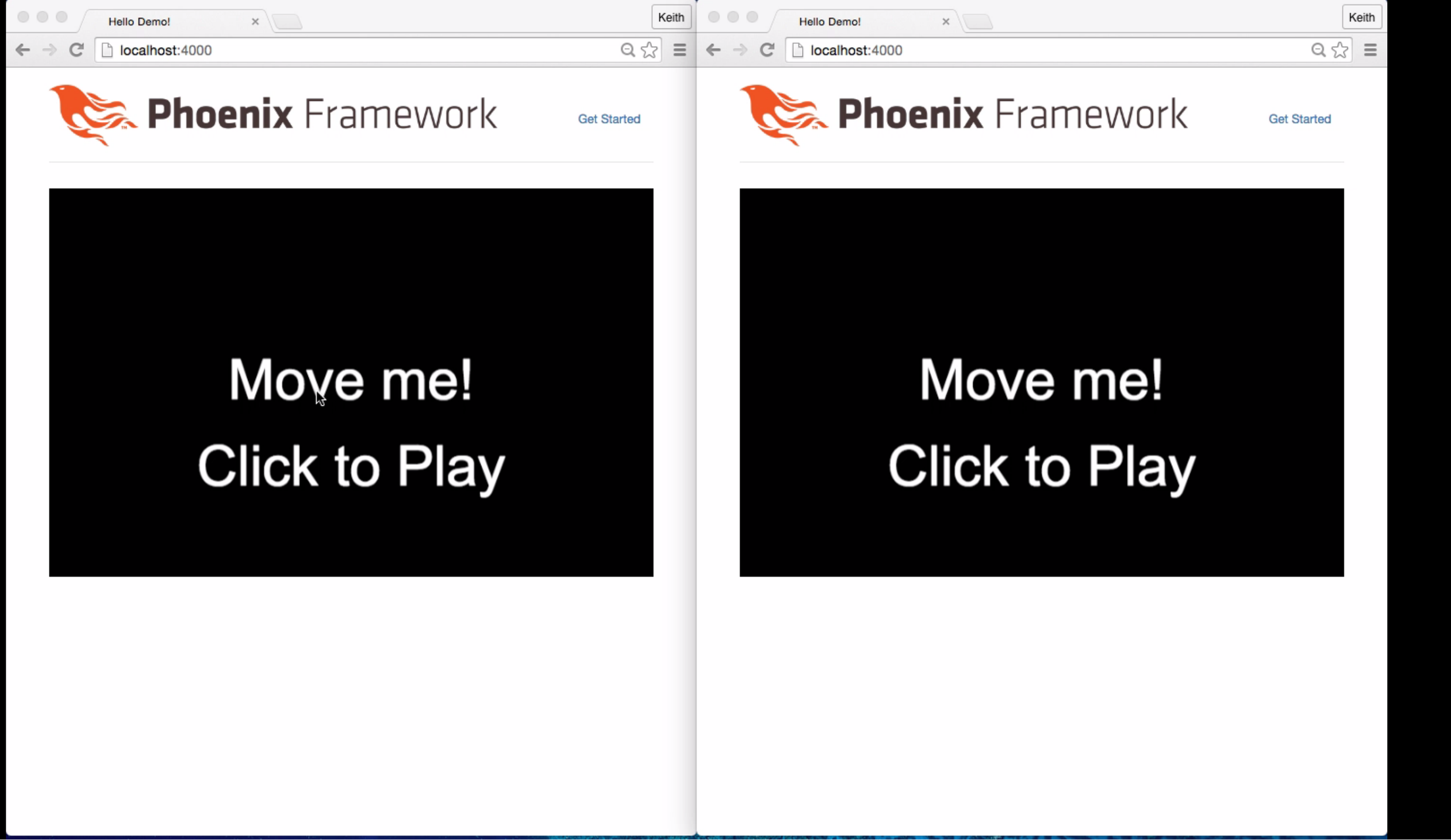
```elixir
defmodule Demo.PlayerSupervisor do
  alias Demo.Player

  def start_link do
    import Supervisor.Spec, warn: false
    children = [
      worker(Player, [], [restart: :transient])
    ]
    opts = [strategy: :simple_one_for_one,
    max_restart: 0, name: __MODULE__]
    Supervisor.start_link(children, opts)
  end

  def start(id) do
    Supervisor.start_child(__MODULE__, [[id]])
```

player.ex

```elixir
defmodule Demo.Player do
  use GenServer
  alias Demo.{Endpoint, Randomise}

  defmodule State do
    defstruct id: nil,
              position: %{x: 0, y: 0},
              type: "square"
  end

  ### PUBLIC API ###
```



Rock Paper Scissors


Hello Demo!
Keith
localhost:4000
Phoenix Framework
Get Started
Move me!
Click to Play
Hello Demo!
Keith
localhost:4000
Phoenix Framework
Get Started
Move me!
Click to Play

github.com/ktec/phoenixphaserdemo

Message Flow


Even Elixir/Erlang/OTP is not that
fast

Using broadcast_from! is not enough

So what to do?

One option is to have the server gather and send only one update,

and set a broadcast frequency for the node

```
def start_broadcasting do
  {:ok, _} = :timer.send_interval(200, :broadcast)
end

def handle_info(:broadcast, state) do
  # gather all the information the client needs here....
  broadcast_update(state)
  {:noreply, state}
end
```

Demos

rocks.globalkeith.com

dots.globalkeith.com

snake.globalkeith.com

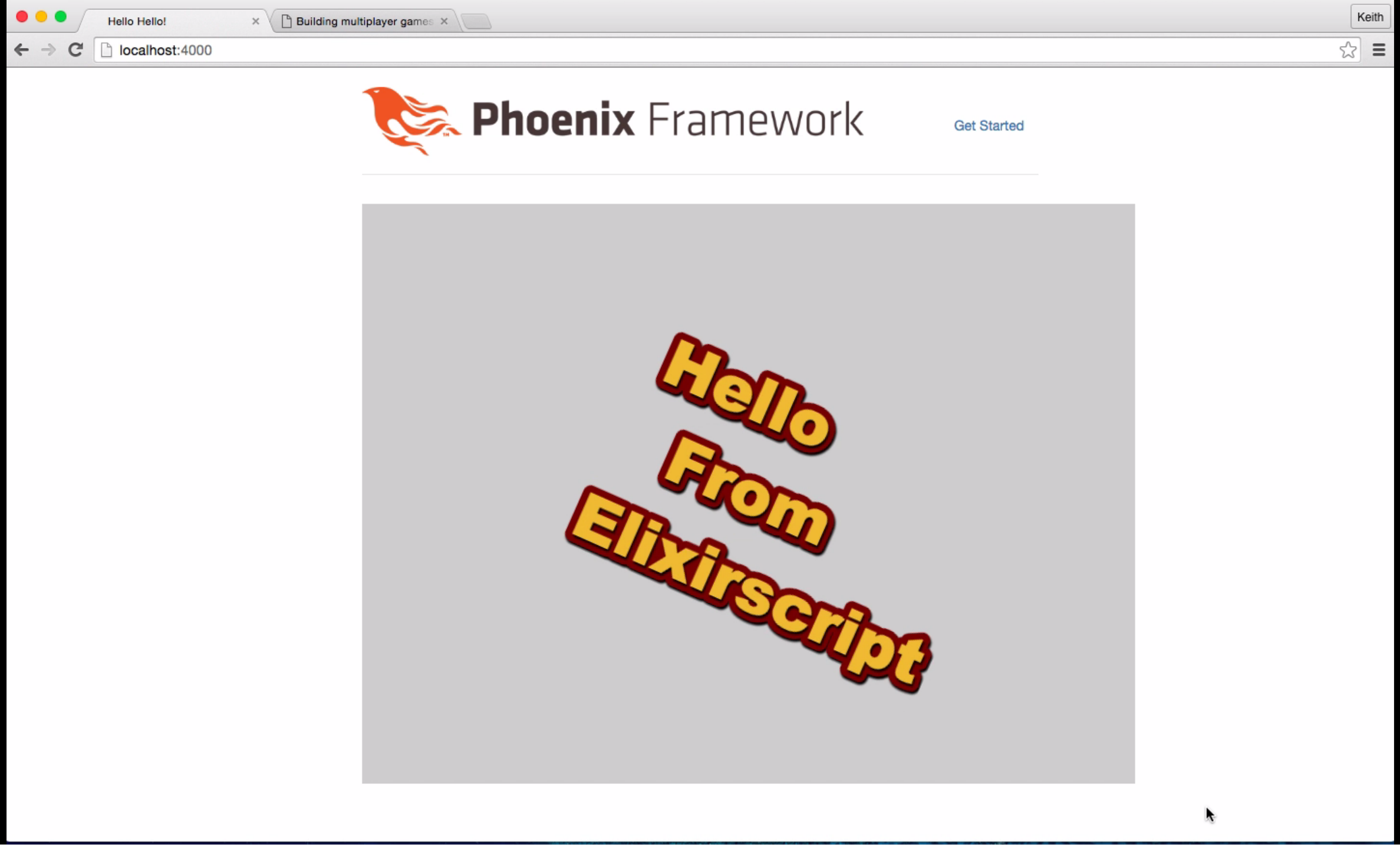
Future

ElixirScript

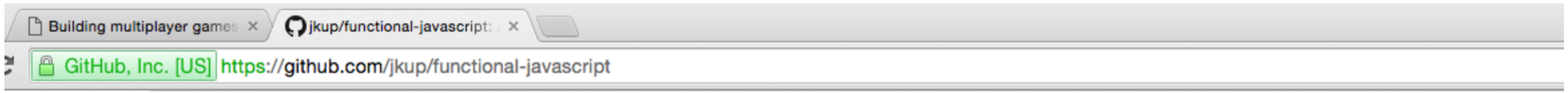
```
defmodule Game do
  JS.import Menu, "./Elixir.Menu"

  def init() do
    game = JS.new Phaser.Game,
           [800, 600, Phaser.CANVAS, "phaser"]

    game.state.add("menu", %{
      "create" => &Menu.create/1,
      "update" => &Menu.update/1
      })

    game.state.start("menu")
  end
end
```

ElixirScript

```
def update(game) do
  Elixir.Enum.each(sprites, fn(sprite) ->
    JS.update(sprite, %{ "rotation" => sprite.rotation + 0.1 })
  end)
end

defp init() do
  Elixir.Agent.start(fn -> %{ "sprites" => [] } end, [name: __MODUL
end

defp sprites() do
  Elixir.Agent.get(__MODULE__, fn(x) -> x.sprites end)
end

defp add_sprite(sprite) do
  Elixir.Agent.update(__MODULE__, fn
    (x) -> %{ x | "sprites" => x.sprites ++ [sprite] }
  end)
end
```


Hello Hello!
Building multiplayer games
Keith
localhost:4000
Phoenix Framework
Get Started
Hello
From
Elixirscript

Functional Javascript



README.md

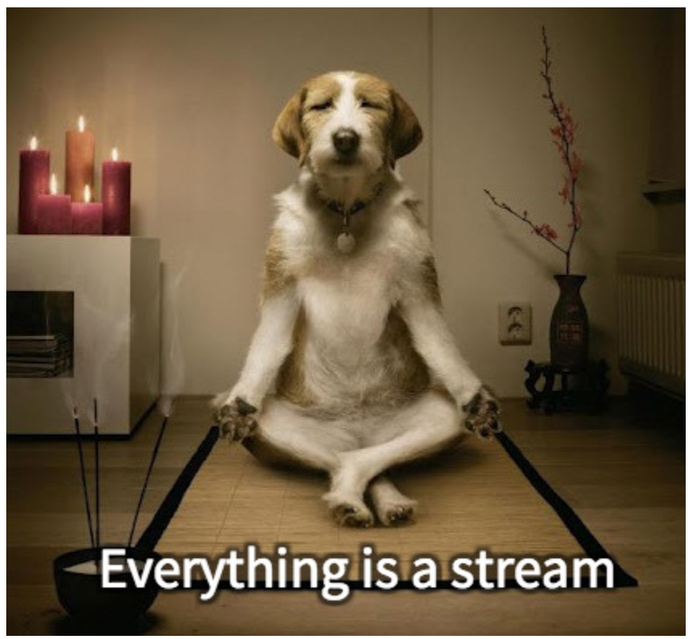
Functional JavaScript Resources

This is a list of resources for functional programming in JavaScript. Please feel free to submit any of your own via pull request.

Videos

Free

- **Brian Lonsdorf** - Hey Underscore, You're Doing It Wrong!
- **David Nolen** - Immutability: Putting The Dream Machine To Work
- **John-David Dalton** - Unorthodox Performance
- **Jafar Husain** - Asynchronous Programming at Netflix
- **Brian Lonsdorf** - Functional programming patterns for the non-mathematician
- **Egghead.io** - Asynchronous Programming: The End of The Loop
- **James Coglan** - Practical functional programming: pick two
- **Mattias Petter Johansson** - Functional Programming in Javascript



ramda.js or kefir.js

Elm

Hex Package

Generators

Phoenix Presence

Explore Gamification

- Turn based games
- Winning/Losing
- Levels
- Challenges
- Virtual Goods
- Leaderboards
- Power/Energy
- High Scores
- Badges
- Authentication/Sign in
- Social sign in - Facebook/Twitter/Oauth
- Gestures on mobile (http://hammerjs.github.io/)

Phaser 3 aka Lazer

Building multiplayer games | Overview - Building multipl

https://leanpub.com/buildingmultiplayergameswithphoenixandphaser/overview

Store Read Write Support Blog

Building multiplayer games with Phoenix and Phaser

- Overview
- Getting Started
- Upload Book Cover

Overview

Title: Building multiplayer games with Phoenix and Phaser

Authors: Keith Salisbury

Teaser Text: Phoenix and Phaser might sound like a cheesy pop band but get these two frameworks...

Categories: Functional Programming, Gaming, HTML5 and JavaScript

Preview Book

Thanks for Watching

Created by Keith Salisbury / @ktec