elixir

A taste of elixir: Building concurrent and reactive web applications

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


parademem

What’s Elixir

- Functional Programming Language
- Runs on the Erlang Virtual Machine

Design Goals:	Updated Goals(2017)
• Compatibility • Productivity • Extensibility	• Productivity • Maintainability • Reliability

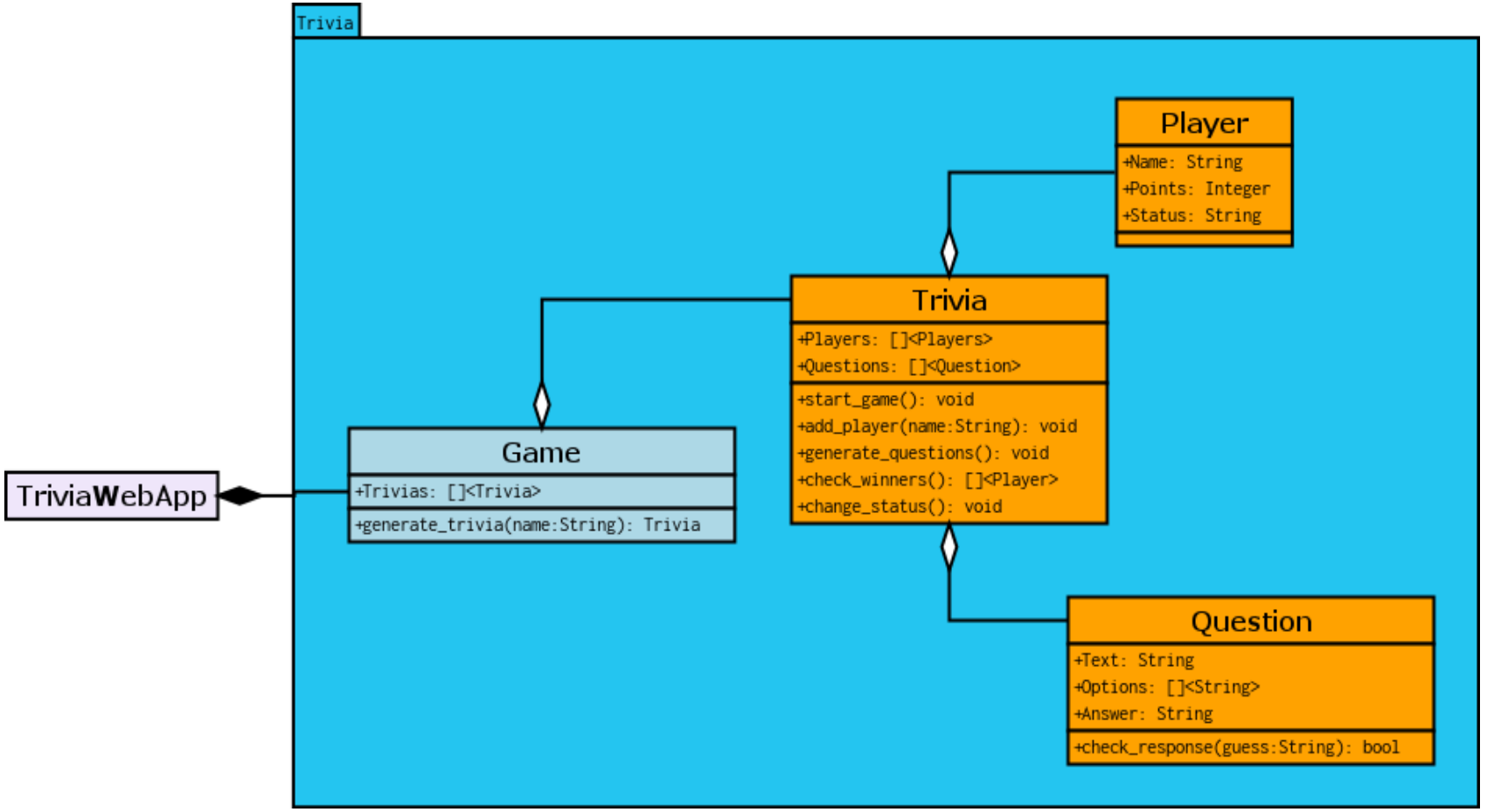
https://elixir-lang.org/blog/2013/08/08/elixir-design-goals/

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

How to create a trivia app in OOP?



Concurrency Issues

- Race conditions
  - Lost updates, access to out-of-date data, etc.
- Workarounds:
  - Confinement.
  - Immutability.
  - Threadsafe data type.
  - Synchronization.

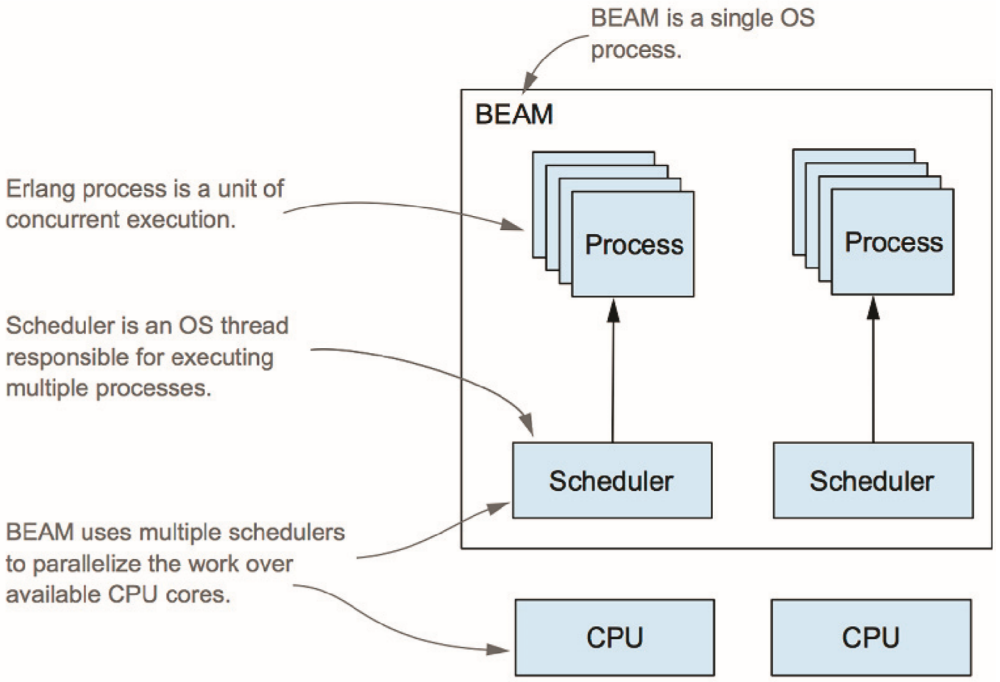
http://web.mit.edu/6.005/www/fa15/classes/20-thread-safety/

https://www.ibm.com/support/knowledgecenter/en/SSEPGG_11.1.0/com.ibm.db2.luw.admin.perf.doc/doc/c0005267.html

Erlang/OTP

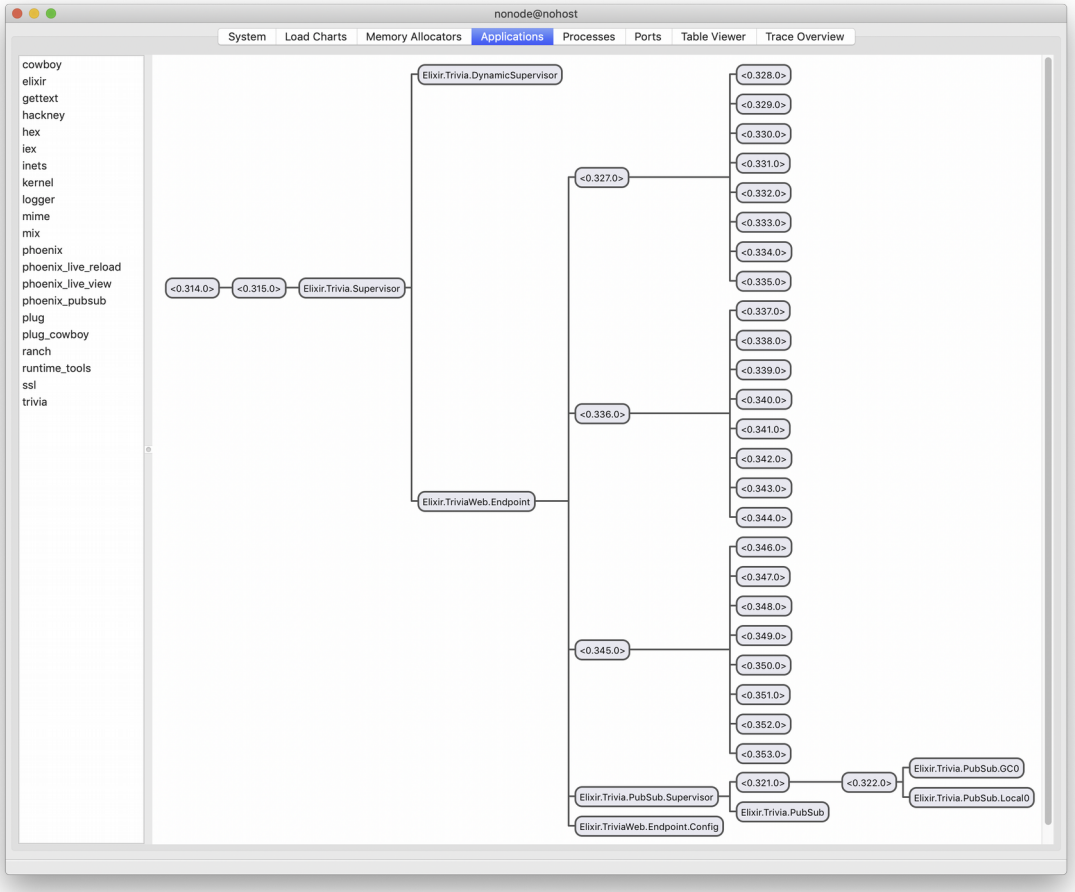
- Complete development environment for concurrent programming.
-  takes advantage of all the ecosystem.

BEAM



https://freecontent.manning.com/dipping-your-toes-into-elixir/

Processes



https://elixir-lang.org/getting-started/processes.html

Phoenix Framework

- MVC
- Productive, Reliable, and Fast.

Creating an app

```
>mix phx.new trivia --no-ecto
```

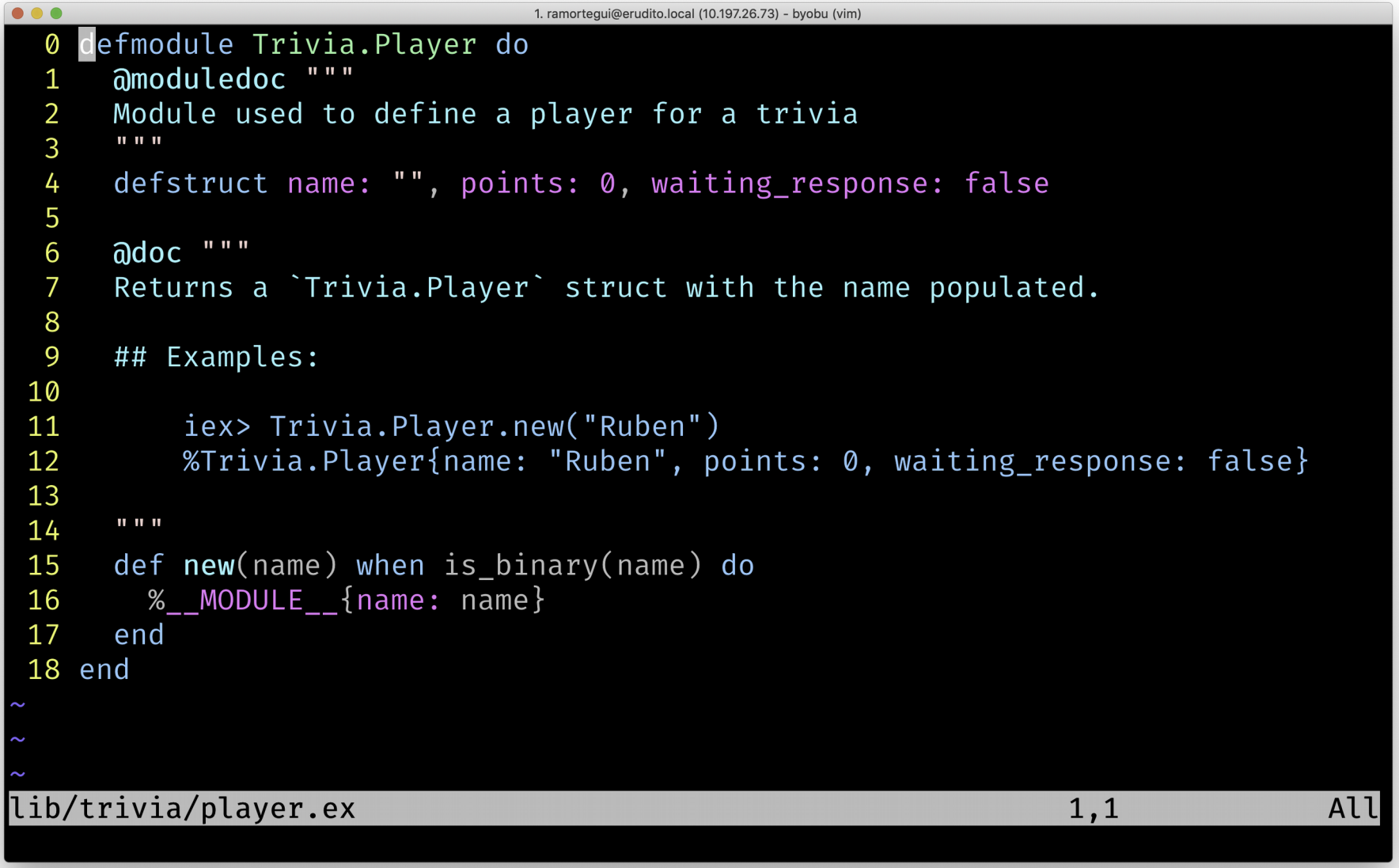
Phoenix app Structure

```
ramortegui trivia> ls -1
README.md
_build
assets
config
deps
lib
mix.exs
mix.lock
priv
test
ramortegui trivia>
```

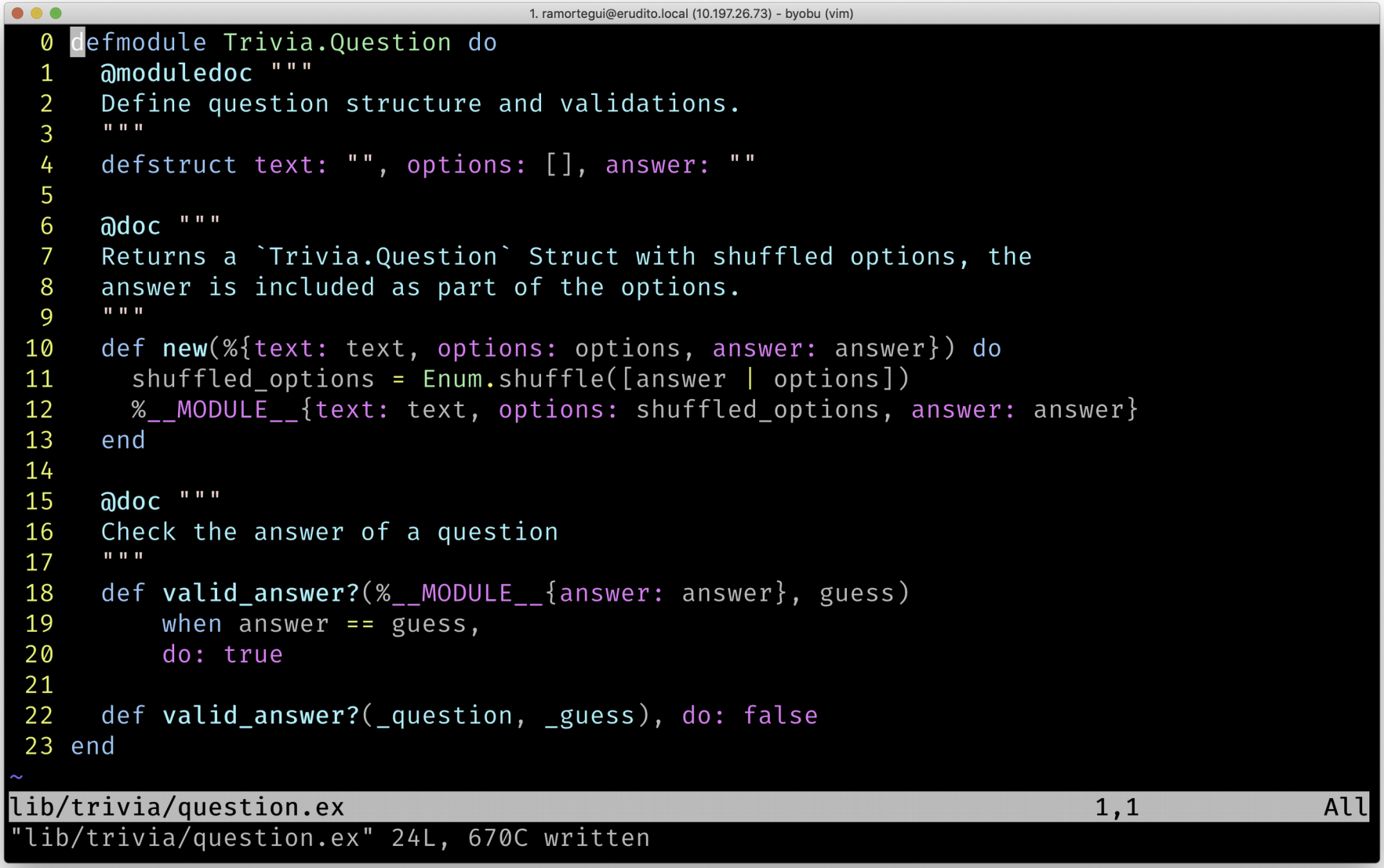
Data Structures

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

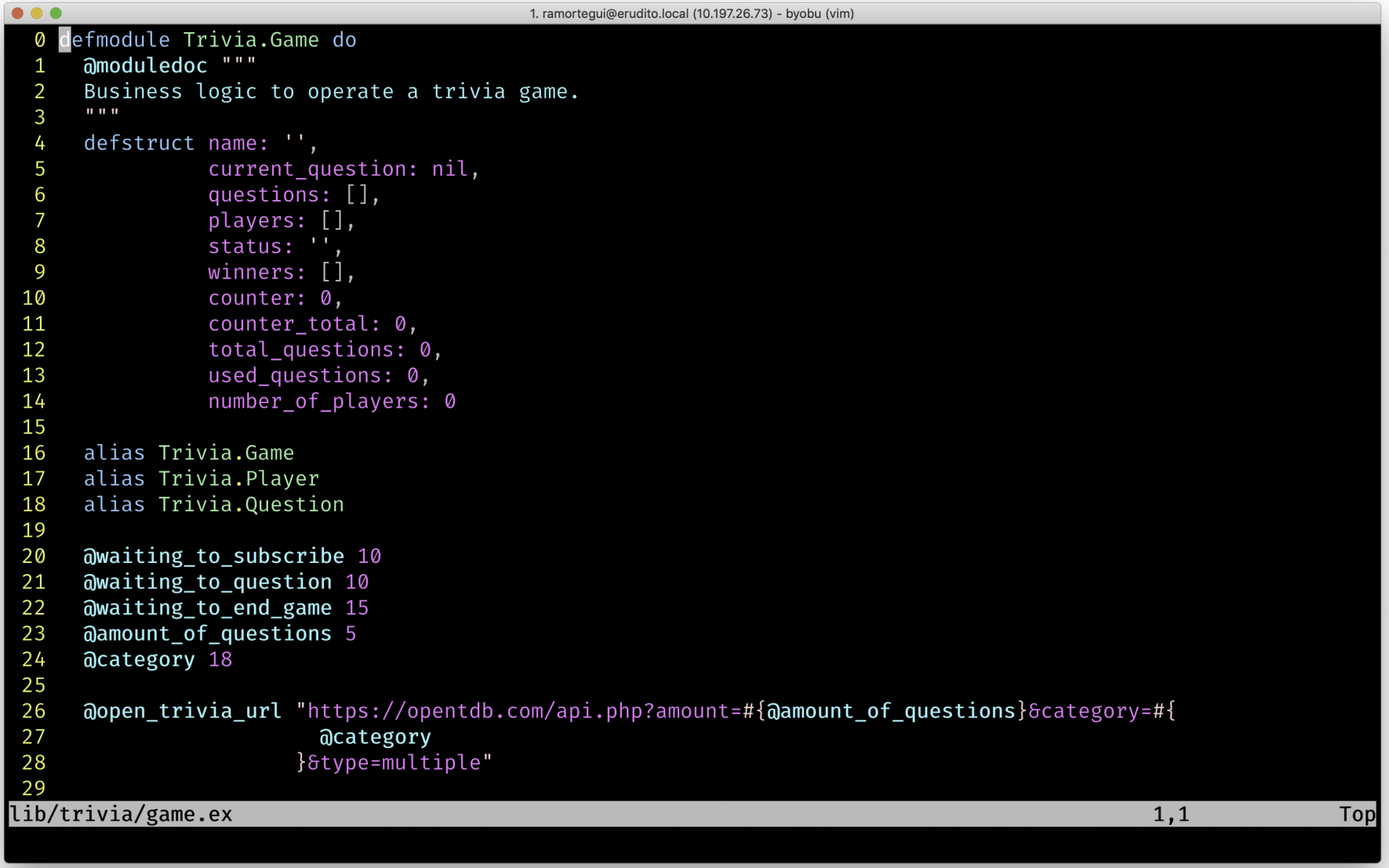
Trivia.Player



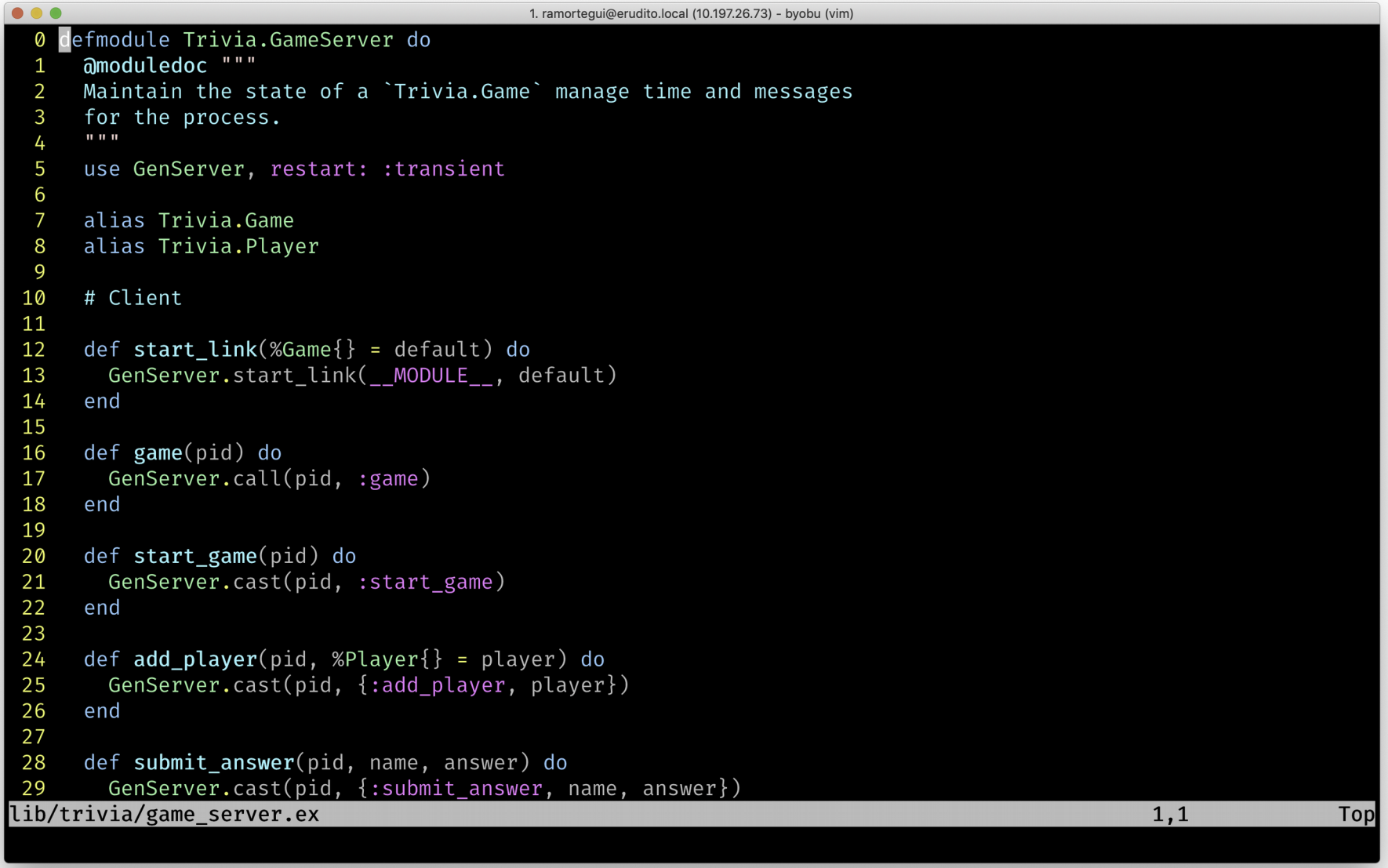
Player.Question



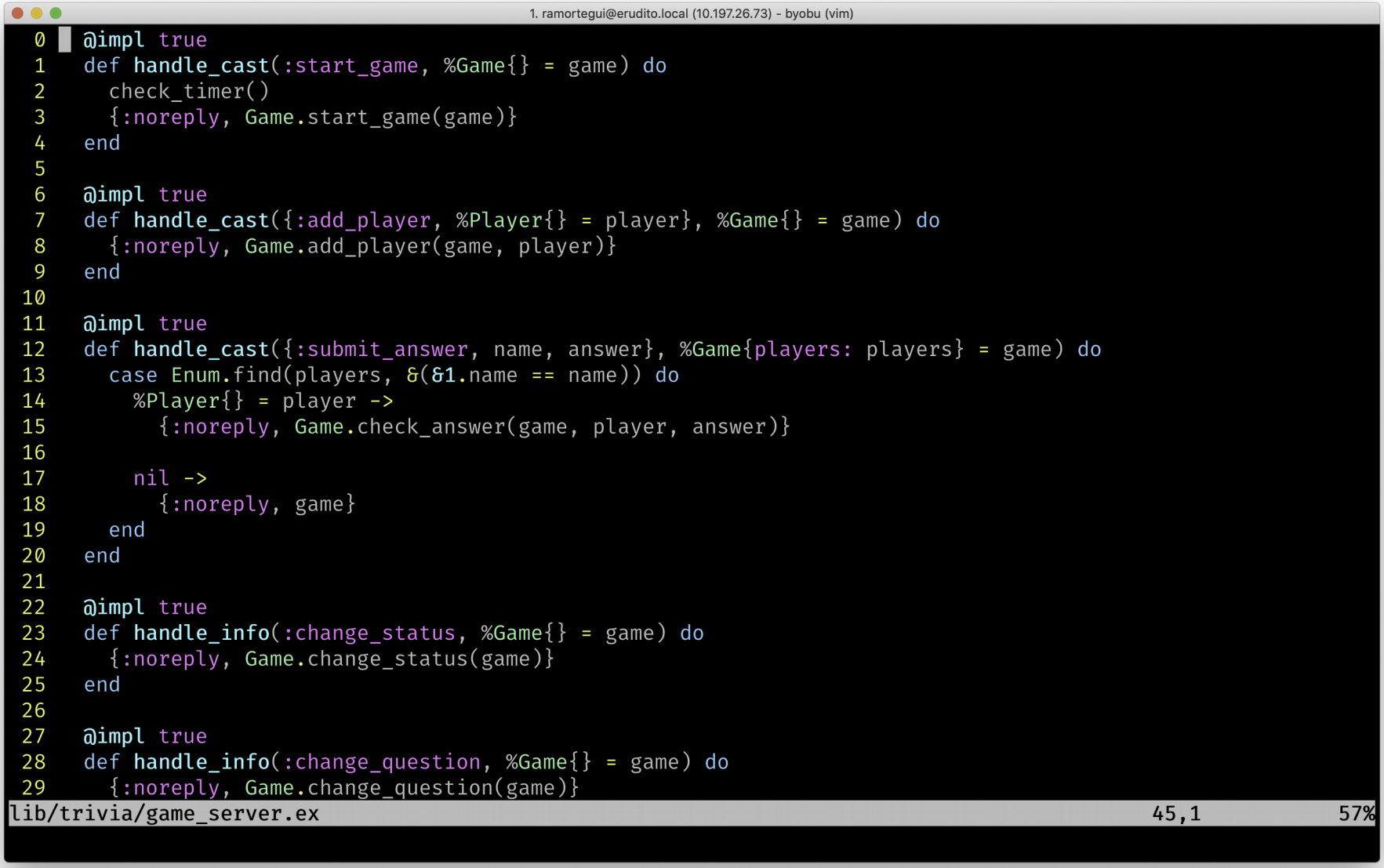
Trivia.Game


1. ramortegui@erudito.local (10.197.26.73) - byobu (vim)
0 defmodule Trivia.Game do
1 @moduledoc """
2 Business logic to operate a trivia game.
3 """
4 defstruct name: '',
5 current_question: nil,
6 questions: [],
7 players: [],
8 status: '',
9 winners: [],
10 counter: 0,
11 counter_total: 0,
12 total_questions: 0,
13 used_questions: 0,
14 number_of_players: 0
15
16 alias Trivia.Game
17 alias Trivia.Player
18 alias Trivia.Question
19
20 @waiting_to_subscribe 10
21 @waiting_to_question 10
22 @waiting_to_end_game 15
23 @amount_of_questions 5
24 @category 18
25
26 @open_trivia_url "https://opentdb.com/api.php?amount=#{@amount_of_questions}&category=#{
27 @category
28 }&type=multiple"
29
lib/trivia/game.ex 1,1 Top

Trivia.GameServer

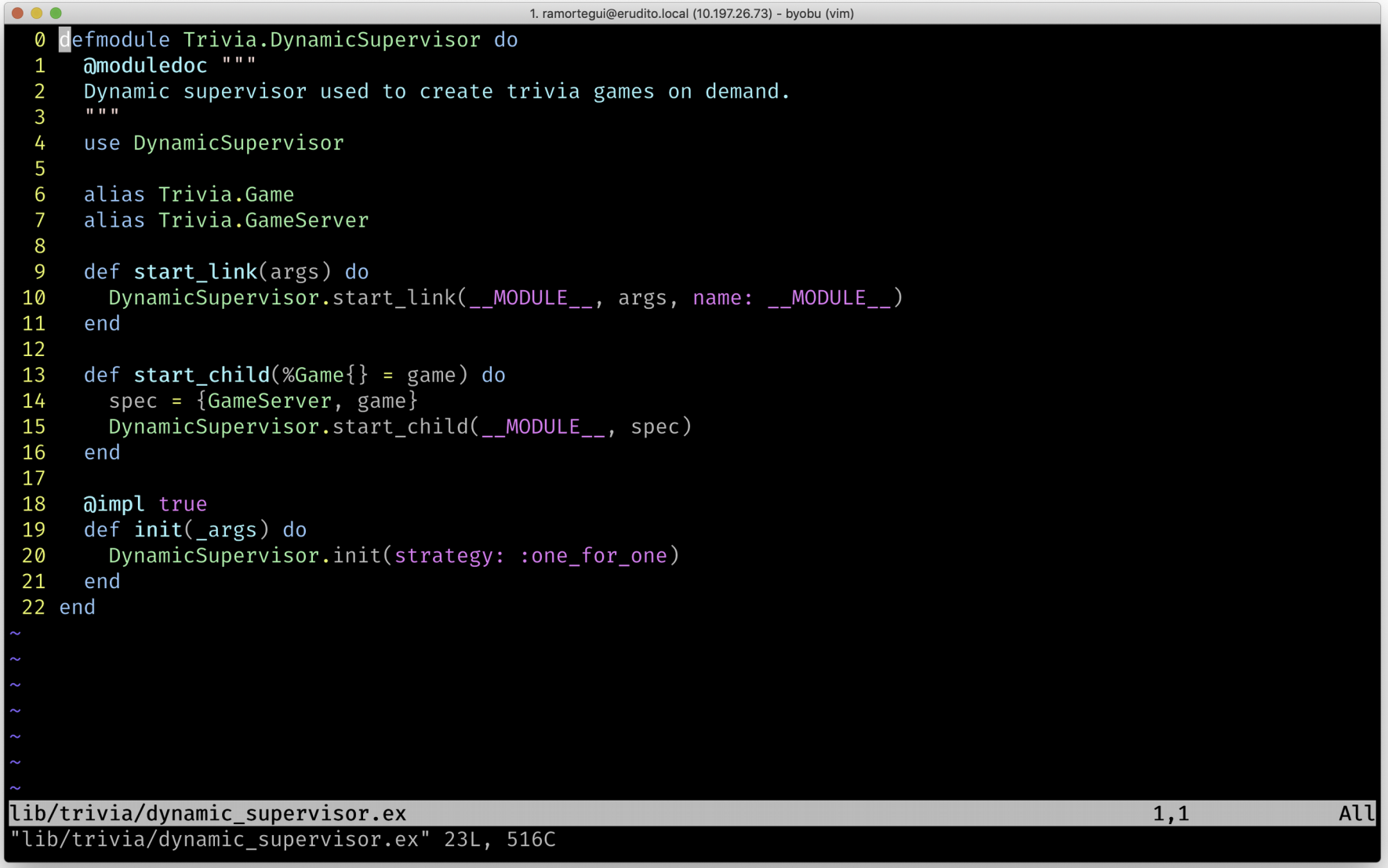


Trivia.GameServer 2



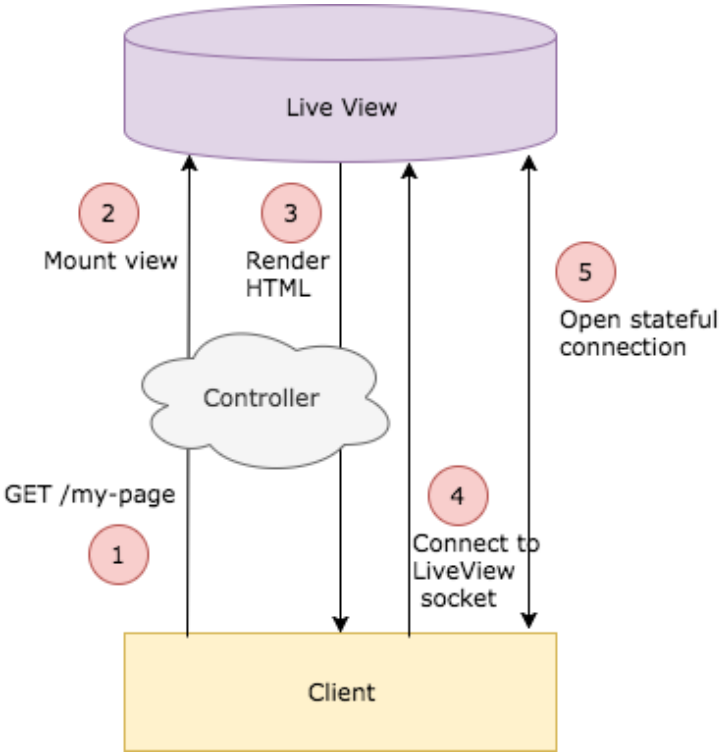
Trivia.GameServer 3

Trivia.DynamicSupervisor



Phoenix Live View

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



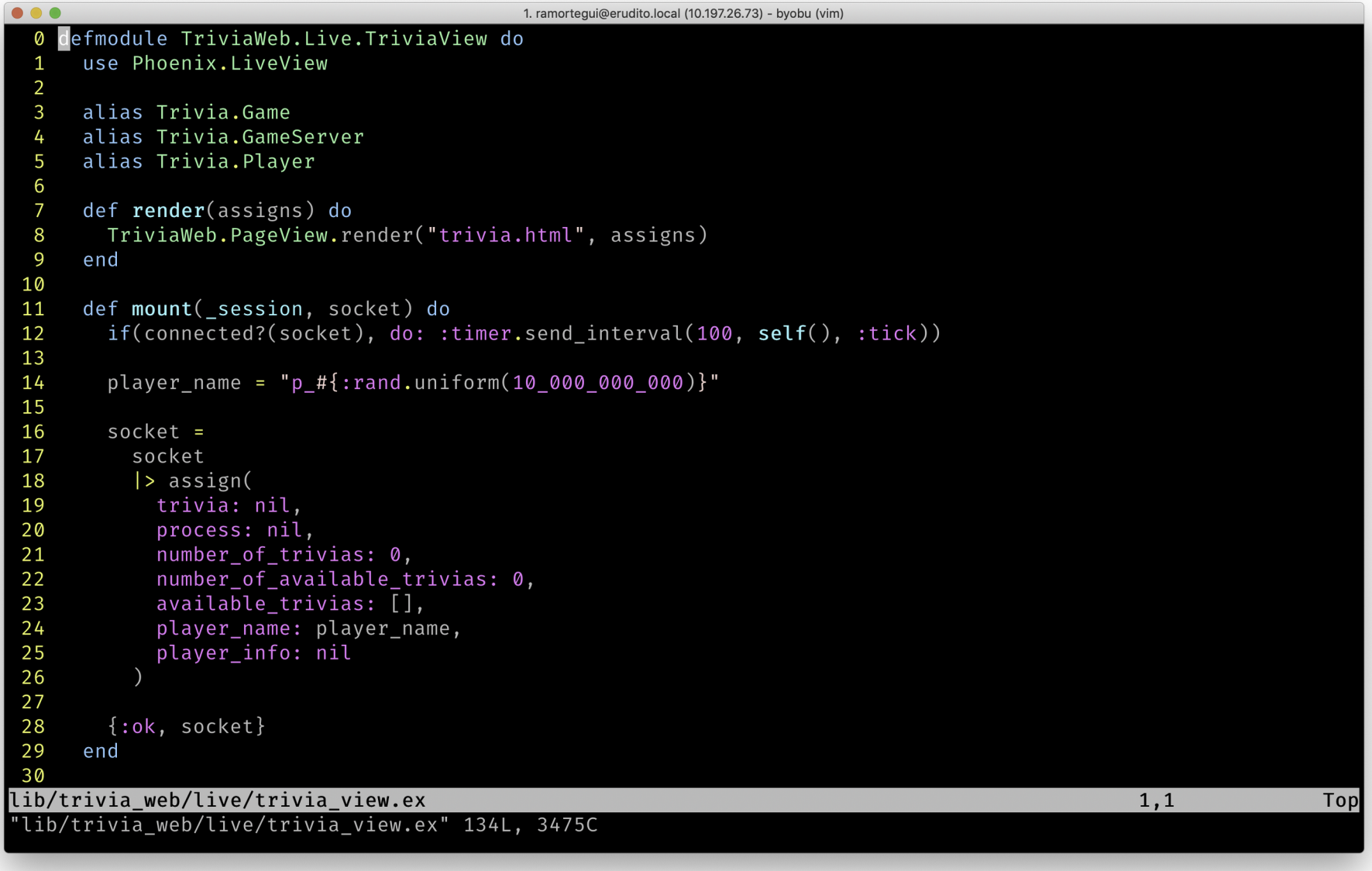
https://elixirschool.com/blog/phoenix-live-view/

Config Live View

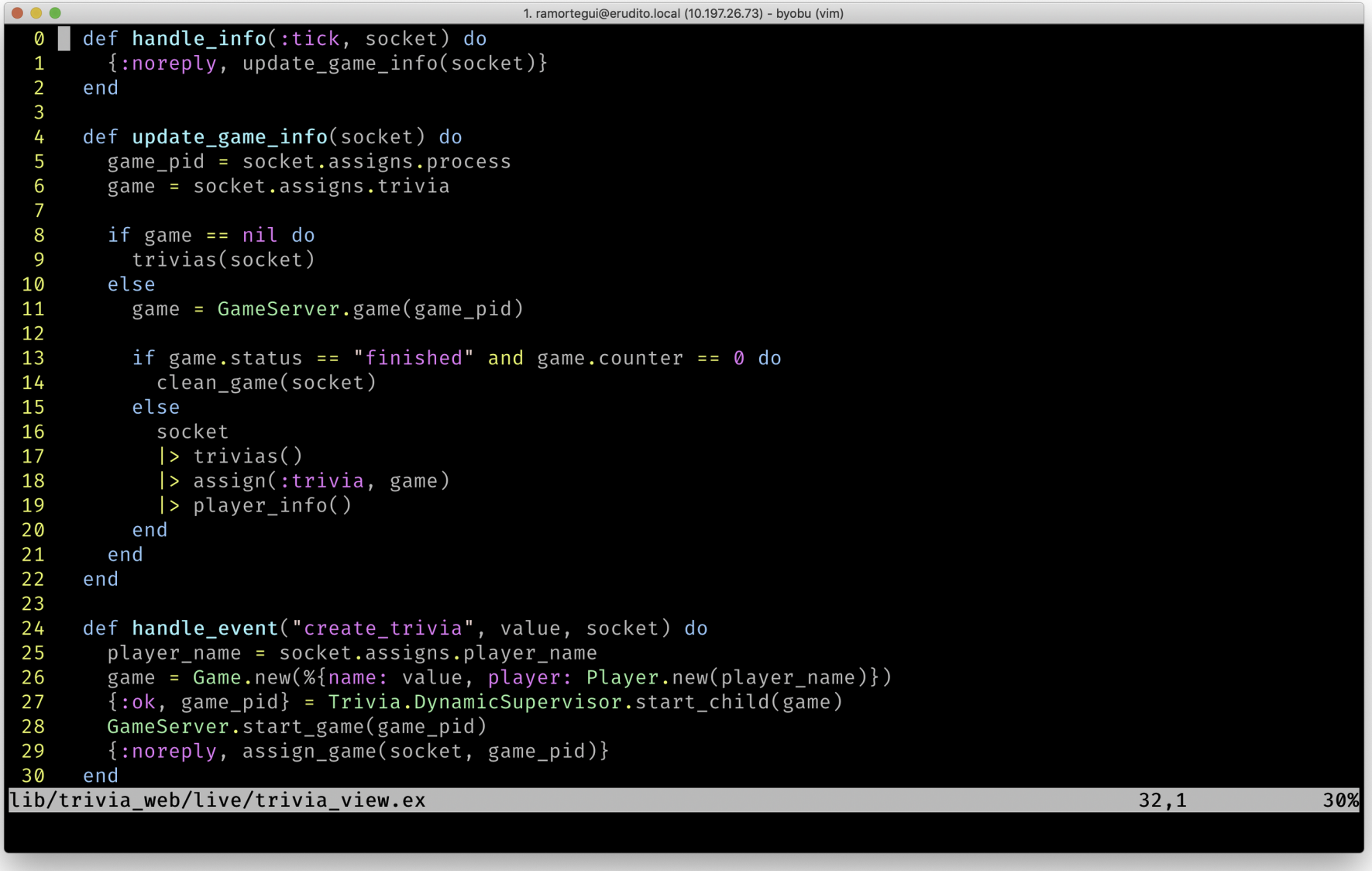
- Add LiveView dependency
- Setup configuration (add signing salt)
- Import functions into the web context
- Enable the socket to connect using the /live route.
- Add Phoenix Live View javascript package as dependency.

https://github.com/ramortegui/phoenix_trivia/commit/bc1bae7d32a24a51ca510e2ee3b2971d3c8b2735

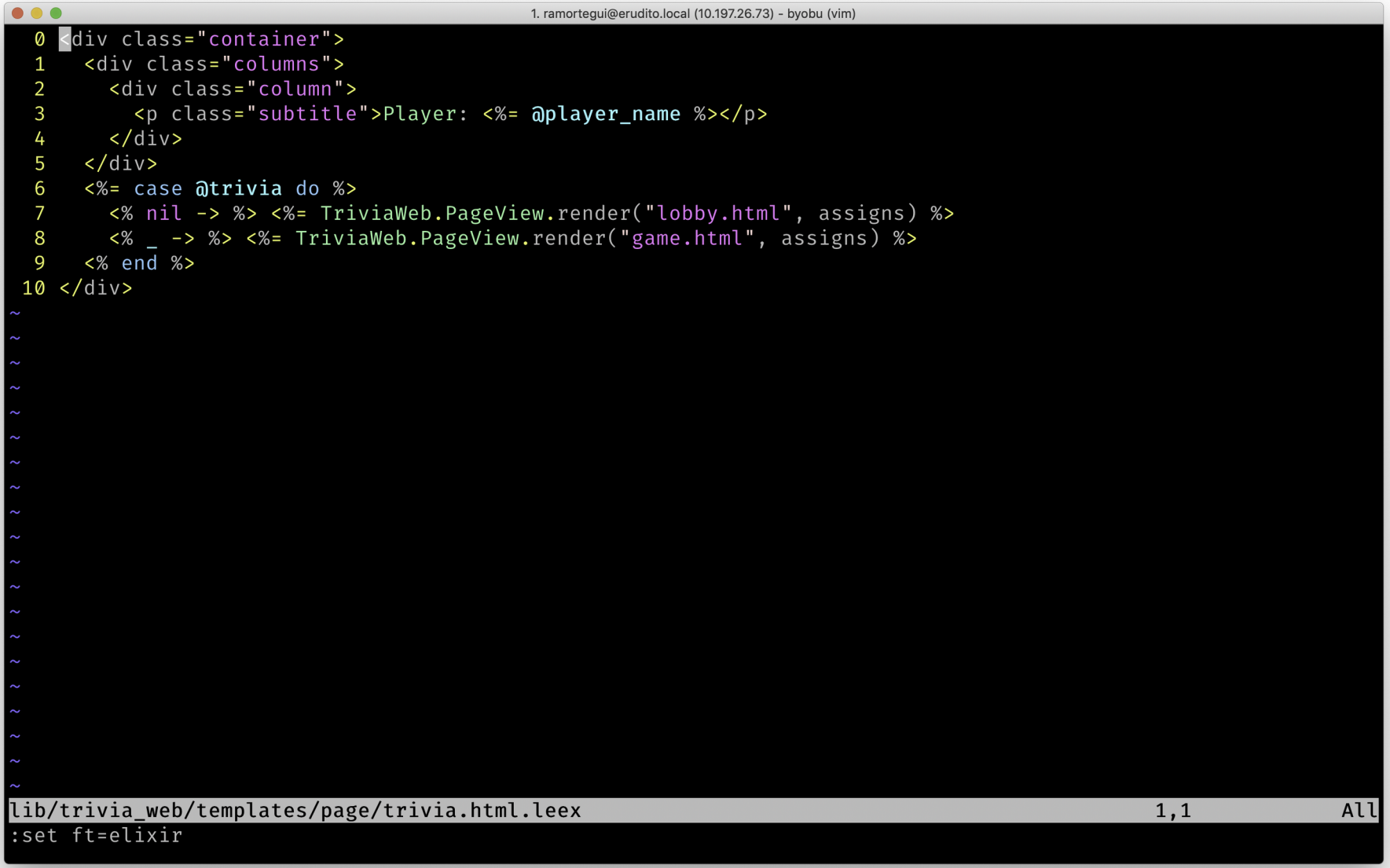
TriviaWeb.Live.TriviaView



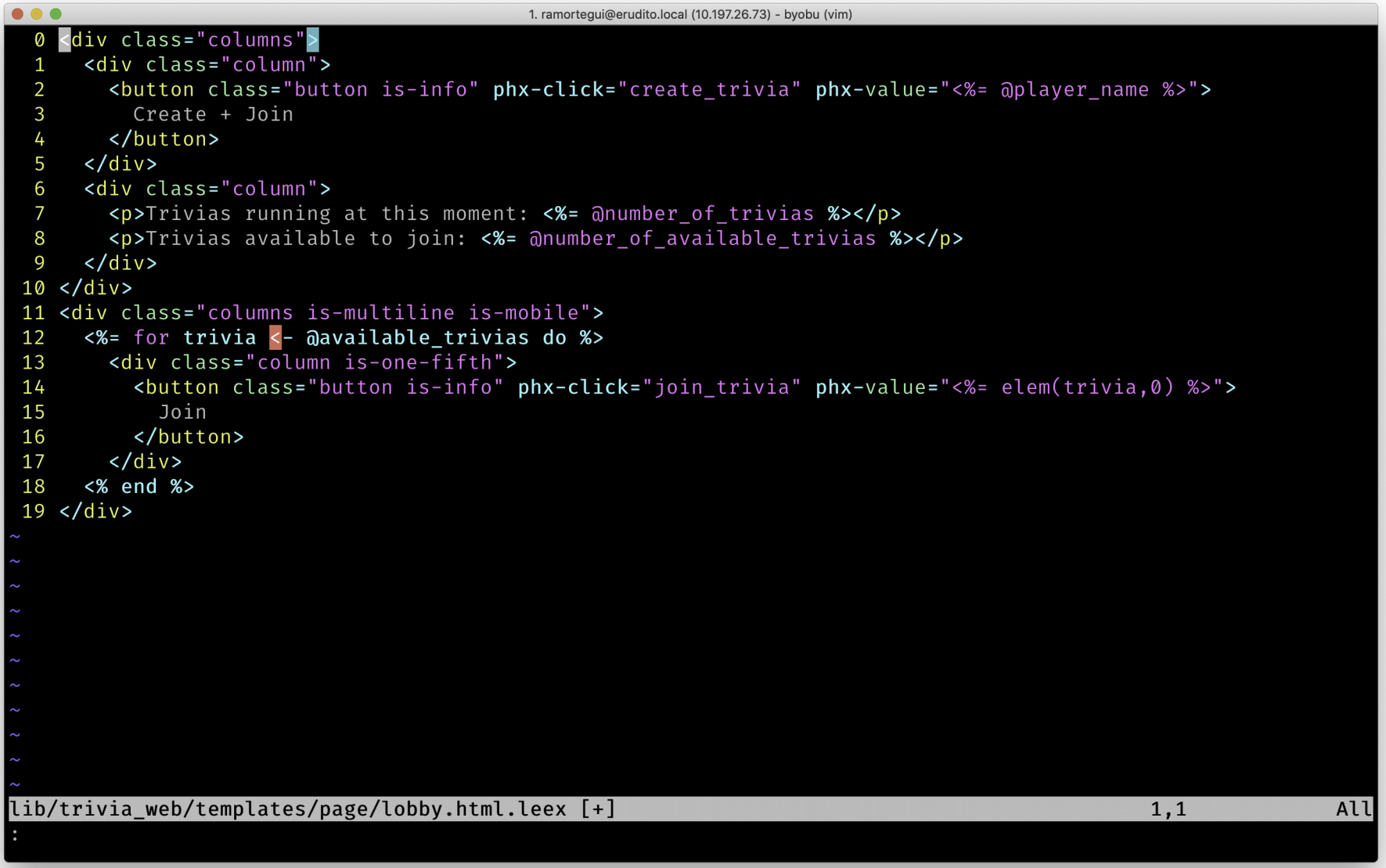
TriviaWeb.Live.TriviaView



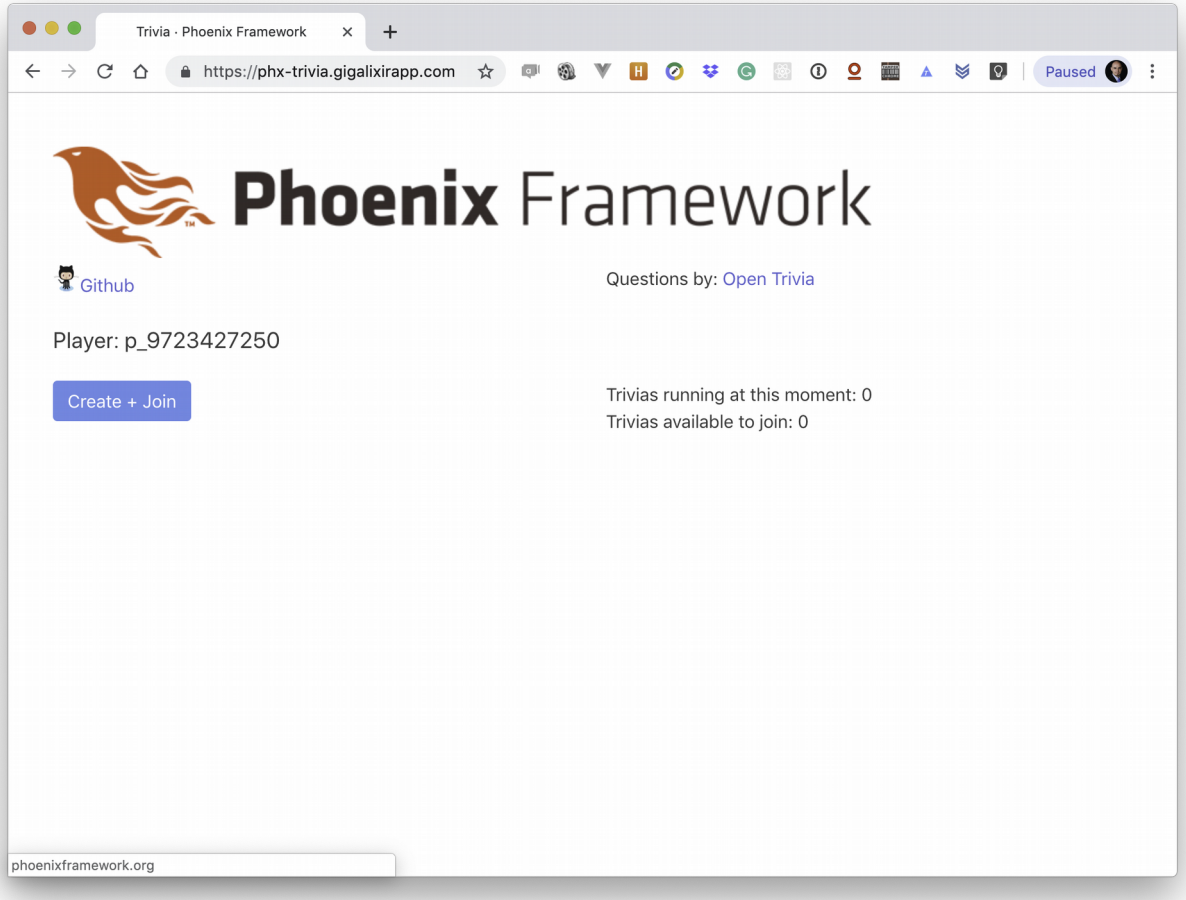
Trivia Live View Template



Trivia Live View Template



https://phx-trivia.gigalixirapp.com



Summary

- Elixir and Phoenix provide an ecosystem to develop applications with concurrency in mind.
- You don’t need to choose between fast and productive when you are going to develop a web application, you could have both.

Q & A?

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

Thanks!

@ramortegui

https://github.com/ramortegui