

**Be a better parent to
your children**

Whoami

Herminio Torres

- email: herminiocesar@gmail.com
- github: [@herminitorres](https://github.com/herminitorres)
- twitter: [@herminitorres](https://twitter.com/herminitorres)



Setting Expectations

Our issues:

Duplicate name to
unique processes

**Working in a
concurrent and
distributed system**

**Storing and
referencing the
naming of our
processes**

Let's go and find out...

PID

```
1 defmodule Actor do
2   use GenServer
3   defstruct name: nil, age: nil, role: nil
4   ...
5 end
```


PID?



- Hard to remember the process
- Starting many processes

MODULE


```
1 GenServer.whereis(Actor)  
2 #PID<0.189.0>
```

MODULE?



- Easy to remember the process
- Starting one process

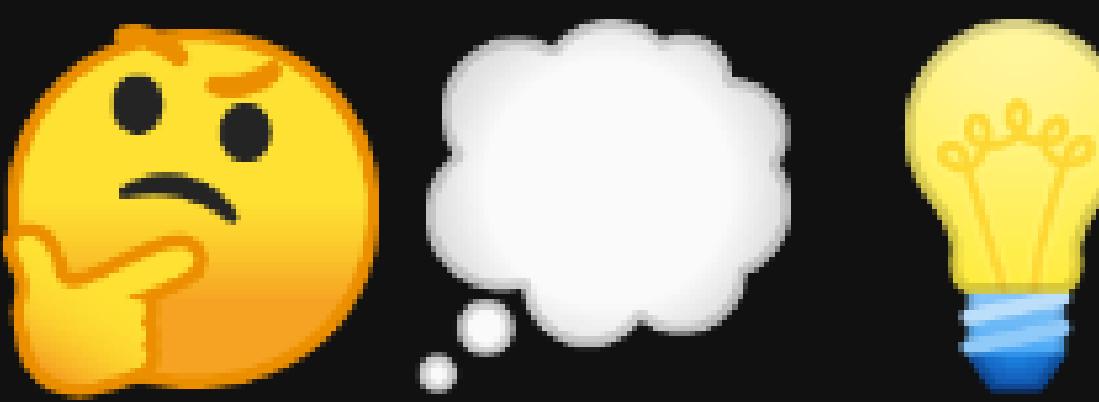
:atom


```
1 # Ex: input: "Christian Bale" | output: :christian_bale"
2 def naming_process_name(actor_name) do
3   actor_name
4   |> String.trim
5   |> String.downcase
6   |> String.split
7   |> Enum.join("_")
8   |> String.to_atom
9 end
```



```
1 GenServer.whereis(:christian_bale)
2 #PID<0.189.0>
```


:atom?



- Easy to remember the process
- Starting many processes
- The BEAM has a maximum size to handling atoms
- Use the same name to many processes

Registry Module

Registry Concepts:

- A local, decentralized and scalable key-value process storage.
- It allows developers to lookup one or more processes with a given key.
- Each entry in the registry is associated to the process that has registered the key.
- If the process crashes, the keys associated to that process are automatically removed.

Where we use?



How we use?



Where we use?

Using in :via tuple

it can be used to register and access named processes.

Using as a dispatcher

it can be used to dispatch based on named to triggered from the caller.

Using as a PubSub

it can be used to implement a local, non-distributed, scalable PubSub

Registrations

Looking up, dispatching and registering are efficient and immediate at the cost of delayed unsubscription.

ETS(Erlang Term Storage)

Note that the registry uses one ETS table plus two ETS tables per partition.

How we use?

Batman Begins



The Machinist




```
1 Registry.keys(
2   Actor.Registry,
3   GenServer.whereis(
4     Actor.via_tuple(%{
5       name: "Christian Bale",
6       age: 50,
7       role: "Bruce Wayne"
8     })
9   )
10 )
11 ["Christian Bale - Bruce Wayne"]
```

```
1 Registry.lookup(  
2   Actor.Registry,  
3   Actor.via_tuple(%{  
4     name: "Christian Bale",  
5     age: 50,  
6     role: "Bruce Wayne"  
7   } )  
8 )  
9 [{#PID<0.189.0>, :actor}]
```



```
1 Registry.lookup(
2   Actor.Registry,
3   Actor_via_tuple(%{
4     name: "Christian Bale",
5     age: 50,
6     role: "Bruce Wayne"
7   })
8 )
9 []
```

Thank you!

<https://unsplash.com/photos/kzQ6gbTR-Fg>

References:

- Registry docs:

<https://hexdocs.pm/elixir/master/Registry.html>

- Elixir — Process Registries:

<https://medium.com/@StevenLeiva1/elixir-process-registries-a27f813d94e3>

References:

- Multiplayer Go with Elixir's Registry, PubSub and dynamic supervisors:
<https://blog.appsignal.com/2019/08/13/elixir-alchemy-multiplayer-go-with-registry-pubsub-and-dynamic-supervisors.html>
- How to start processes with dynamic names in Elixir:
<https://thoughtbot.com/blog/how-to-start-processes-with-dynamic-names-in-elixir>

References:

- Using the Registry in Elixir 1.4:
<https://medium.com/elixirlabs/registry-in-elixir-1-4-0-d6750fb5aeb>
- Process registry in Elixir: a practical example:
<https://www.brianstorti.com/process-registry-in-elixir/>

References:

- Registry in Elixir 1.4 - Youtube Video:
[https://www.youtube.com/watch?
v=VG8bBnOYj2g](https://www.youtube.com/watch?v=VG8bBnOYj2g)
- Elixircasts - 109 - DynamicSupervisor and
Registry:
[https://elixircasts.io/dynamicsupervisor-and-
registry](https://elixircasts.io/dynamicsupervisor-and-registry)

References:

- Process pools with Elixir's Registry:
<https://andrealopardi.com/posts/process-pools-with-elixirs-registry/>
- Darth Vader Image - Unsplash:
<https://unsplash.com/photos/GV2JHwiL21U>

References:

- The Batman Image - Wikipedia:
https://en.wikipedia.org/wiki/Batman_Begins
- The Machinist Image - Wikipedia:
https://en.wikipedia.org/wiki/The_Machinist

Thank you!

Herminio Torres

- email: herminiocesar@gmail.com
- github: [@herminiotorres](https://github.com/herminiotorres)
- twitter: [@herminiotorres](https://twitter.com/herminiotorres)