defrecord Person, first\_name: nil, last\_name: "Dudington" do

def name record do # huh ?

"#{record.first\_name} #{record.last\_name}"

end

end

defrecord User, name: "José", age: 25

guy = Person.new first\_name: "Guy"

guy.name

defmodule ListServer do

@moduledoc """

This module provides an easy to use ListServer, useful for keeping

lists of things.

"""

use GenServer.Behaviour

alias Foo.Bar

### Public API

@doc """

Starts and links a new ListServer, returning {:ok, pid}

## Example

iex> {:ok, pid} = ListServer.start\_link

"""

def start\_link do

:gen\_server.start\_link({:local, :list}, \_\_MODULE\_\_, [], [])

end

### GenServer API

def init(list) do

{:ok, list}

end

# Clear the list

def handle\_cast :clear, list do

{:noreply, []}

end

end

{:ok, pid} = ListServer.start\_link

pid <- {:foo, "bar"}

greeter = fn(x) -> IO.puts "hey #{x}" end

greeter.("Bob")