GenServer Callbacks

- **init(args) (interface - start(), start_link())**
  - {:ok, state *[,timeout]*}
  - {:stop, reason}
- **handle_call(request, from, state) (interface - call())**
  - {:reply, reply, new_state *[, timeout]*}
  - {:stop, reason *[,reply]*, new_state}
- **handle_cast(request, state) (interface - cast())**
  - {:noreply, new_state *[, timeout]*}
  - {:stop, reason, new_state}
- **handle_info(msg, state) (async - message or timeout)**
  - msg = term or :timeout
  - {:noreply, new_state *[,timeout]*}
  - {:stop, reason, new_state}
- **terminate(reason, state) (Callback return = {:stop …..} or interface stop call)**

Oh No! Not Dwayne Dibley

Further Information

- Examples: https://github.com/DwayneDibley
- Docs: https://hexdocs.pm/elixir/GenServer.html
- Docs: http://erlang.org/doc/man/gen_server.html
- Programming Erlang (Joe Armstrong)
- https://www.youtube.com/watch?v=QBIWMgPZo2Q