

The Final Step!

Port the code to use GenServer



```

1 defmodule NameServer4 do
2   @moduledoc """
3     Very simple name server supporting transactions, using GenServer:
4     """

```

```

5   use GenServer

```

```

6
7   ## Interface -----
8   # def start(name) do
9   #   Process.register(spawn(NameServer3, :loop, [name, %{}]), name)
10  #   :ok
11  # end
12  def start(name) do
13    {:ok, pid} = GenServer.start(__MODULE__, :ok, name: name)
14    :ok
15  end
16
17  # def add(serverName, name, place) do
18  #   rpc(serverName, {:add, name, place})
19  # end
20  def add(serverName, name, place) do
21    GenServer.call(serverName, {:add, name, place})
22  end
23
24  # def find(serverName, name) do
25  #   rpc(serverName, {:find, name})
26  # end
27  def find(serverName, name) do
28    GenServer.call(serverName, {:find, name})
29  end
30

```

Interface

```

31  ## Implementation -----

```

```

32  @impl true
33  def init(_) do
34    {:ok, %{}}
35  end

```

```

36
37  # defp handleRequest({:add, name, place}, state) do
38  #   newState = Map.put(state, name, place)
39  #   {:ok, newState}
40  # end
41  @impl true
42  def handle_call({:add, name, place}, _from, state) do
43    newState = Map.put(state, name, place)
44    {:reply, :ok, newState}
45  end
46
47  # defp handleRequest({:find, name}, state) do
48  #   {state[name], state}
49  # end
50  @impl true
51  def handle_call({:find, name}, _from, state) do
52    {:reply, state[name], state}
53  end

```

Implementation

```

54 end

```