

```

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

```

Interface

```

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

```

Implementation

```

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
54  end

```

```

1  defmodule NameServer5 do
2    @moduledoc """
3    Very simple name server supporting transactions, using GenServer:
4    """
5    use GenServer
6

```

Interface

```

7  ## Interface -----
8  def start(name) do
9    {:ok, pid} = GenServer.start(__MODULE__, :ok, name: name)
10   :ok
11 end
12
13 def add(serverName, name, place) do
14   GenServer.cast(serverName, {:add, name, place})
15 end
16
17 def find(serverName, name) do
18   GenServer.call(serverName, {:find, name})
19 end
20

```

Registered name of the server, cleared on exit.

Term passed to init() callback

Callback module (This module)

```

21  ## Implementation -----
22  @impl true
23  def init(_) do
24    {:ok, %{}}
25  end
26
27  @impl true
28  def handle_cast({:add, name, place}, _from, state) do
29    newState = Map.put(state, name, place)
30    {:noreply, newState}
31  end
32
33  @impl true
34  def handle_call({:find, name}, _from, state) do
35    {:reply, state[name], state}
36  end
37  end
38

```

Implementation

Cast: does not send a reply. The calling process does not have to wait.

Call: does send a reply.