

NameServer1 Test

- Create the server
- Add a name and place
- Retrieve the place by name

```
1 ~ defmodule GenServerSlidesTest do
2   use ExUnit.Case
3   doctest NameServer1
4
5 ~   test "1 - start the server" do
6     assert NameServer1.start(:my_server) == :ok
7     assert NameServer1.rpc(:my_server, {:add, :dwayne, "Red Dwarf"}) == :ok
8     assert NameServer1.rpc(:my_server, {:find, :dwayne}) == "Red Dwarf"
9   end
10 end
```

The client and server parts.

Interface

Functions called by the process using the server.

Server

The server.

Implementation

Functions called by the server to implement the server logic.

```
1 defmodule NameServer1 do
2   @moduledoc """
3   Very simple name server supporting two methods:
4   add: Add a name and a place.
5   find: Given a name, return the place or nil
6   """
7
8   def start(name) do
9     Process.register(spawn(NameServer1, :loop, [name, %{}]), name)
10    :ok
11  end
12
13  def rpc(name, request) do
14    send(name, {self(), request})
15
16    receive do
17      {_name, response} -> response
18    end
19  end
20
21  def loop(name, state) do
22    receive do
23      {from, request} ->
24        {response, newState} = handleRequest(request, state)
25        send(from, {name, response})
26        loop(name, newState)
27    end
28  end
29
30  defp handleRequest({:add, name, place}, state) do
31    newState = Map.put(state, name, place)
32    {:ok, newState}
33  end
34
35  defp handleRequest({:find, name}, state) do
36    {state[name], state}
37  end
38 end
39
```