What happens if a method call crashes?

- The whole server will crash! Not good

Lister: "I'm going to use my brains for the first time in my life."
Kryten: "Considering the circumstances, sir, do you really believe that's wise?"

```elixir
 1 ∨ defmodule NameServer2 do
 2 ∨   @moduledoc """
 3     Very simple name server supporting transactions:
 4     - In the event of a crash, the caller is sent a message.
 5     - All other processes using the server will not be affected.
 6     """
 7 >   def start(name) do▭
10     end
11
12 >   def rpc(name, request) do▭
18     end
19
20 ∨   def loop(name, state) do
21 ∨     receive do
22 ∨       {from, request} ->
23 ∨         try do
24           {response, newState} = handleRequest(request, state)
25           send(from, {name, response})
            loop(name, newState)
          rescue
28 ∨         thrown_value ->
29             send(from, {:exception, "#{inspect(thrown_value)}"})
30             loop(name, state)
31           end
32       end
33     end
34
35 >   defp handleRequest({:add, name, place}, state) do▭
38     end
39
40 >   defp handleRequest({:find, name}, state) do▭
42     end
43
44 ∨   defp handleRequest({:crash, name}, state) do
45       1 / 0
46       {state[name], state}
47     end
48 end
```

Interface

Functions called by the process using the server.

Server

Runs in the server Process.

Implementation

Functions called by the server process to implement the server logic.