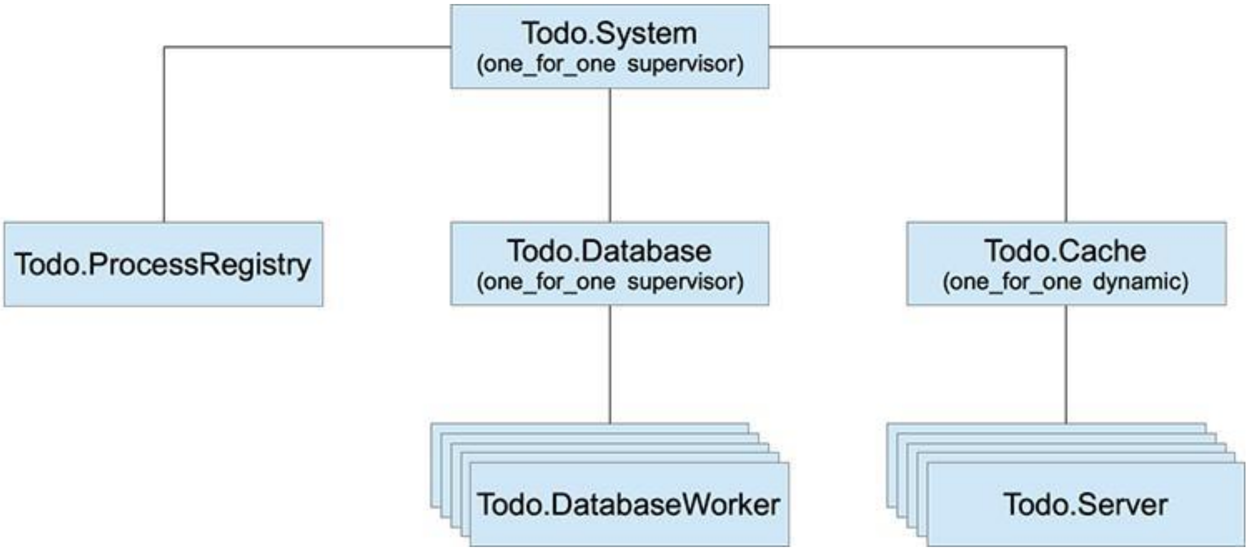
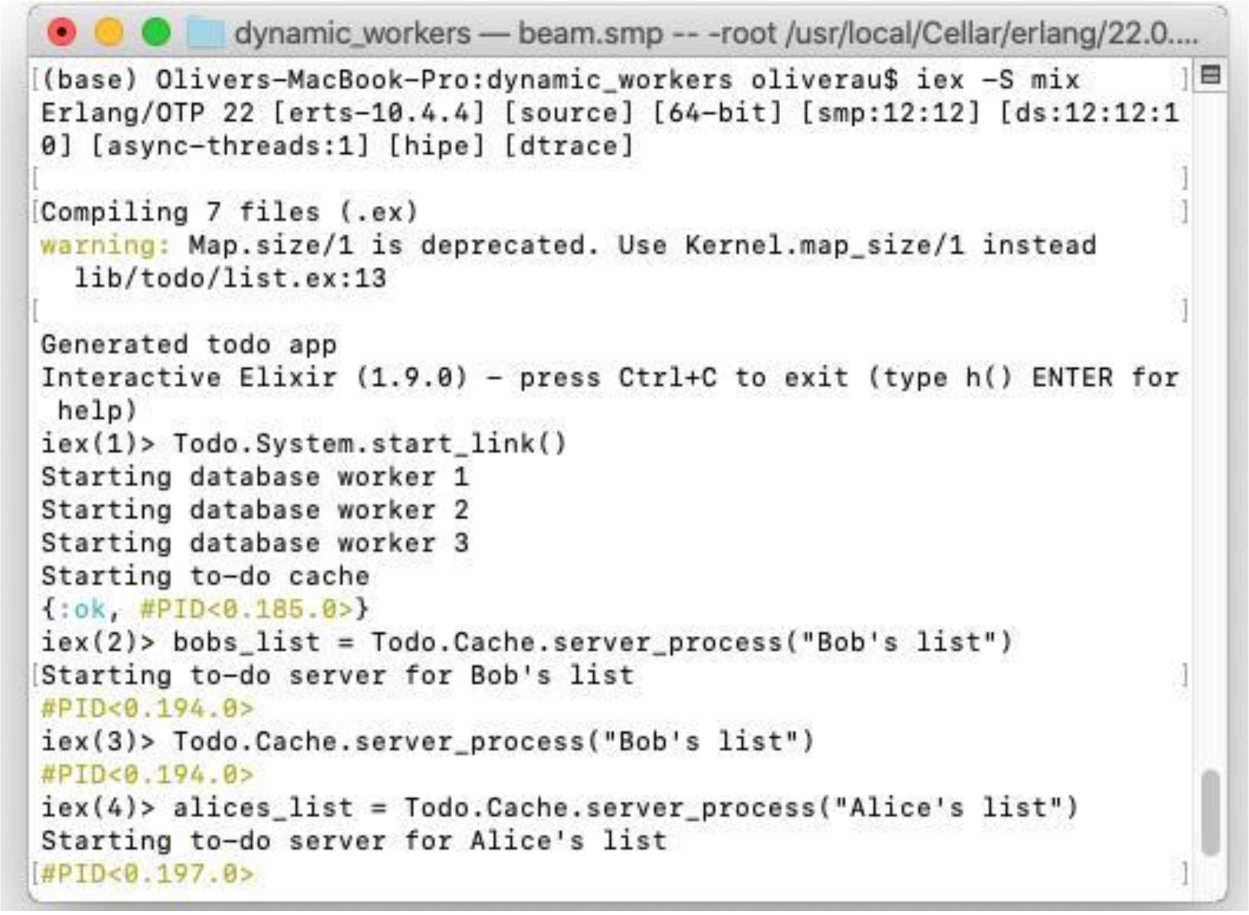
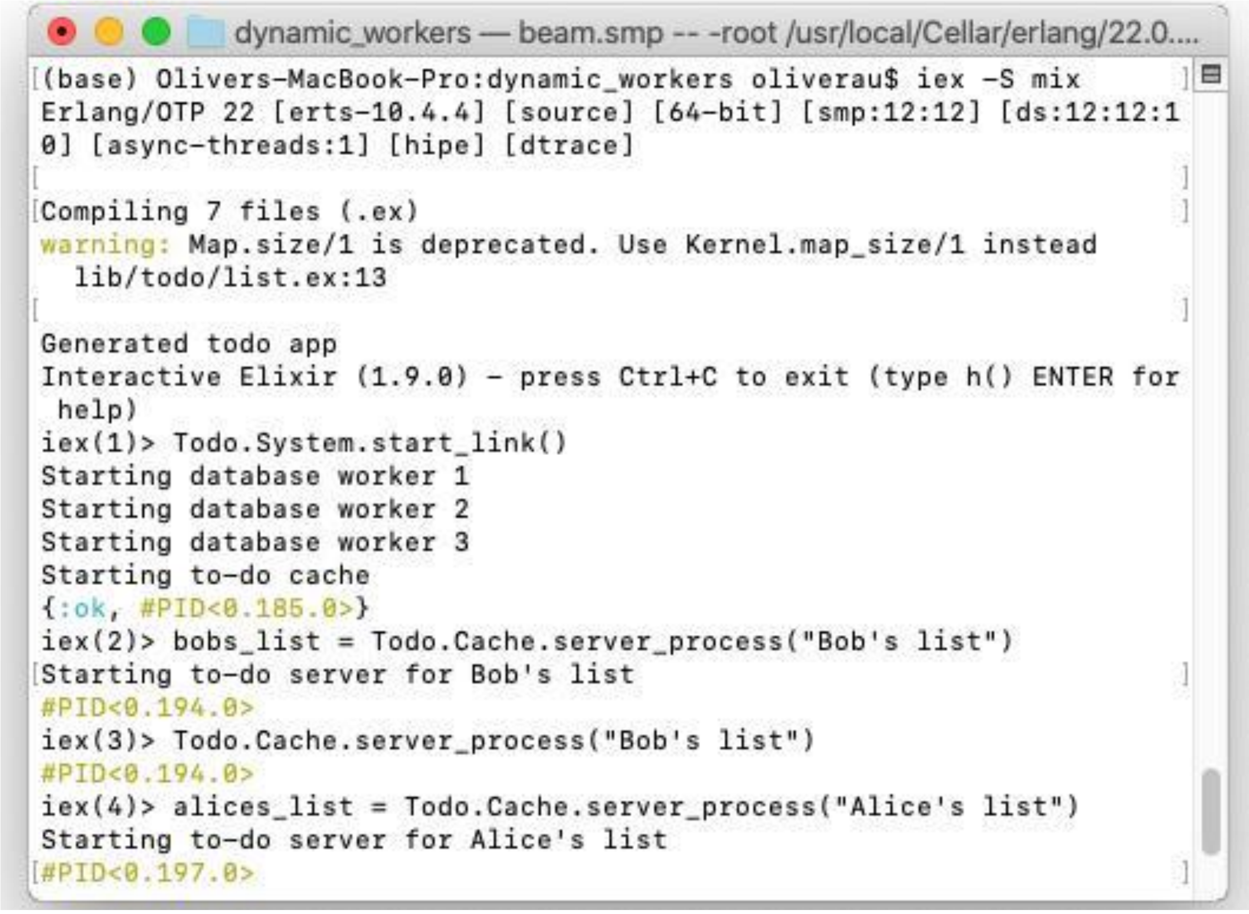
System architecture



Original Test Page


dynamic_workers — beam.smp -- -root /usr/local/Cellar/erlang/22.0....

```
(base) Olivers-MacBook-Pro:dynamic_workers oliverau$ iex -S mix
Erlang/OTP 22 [erts-10.4.4] [source] [64-bit] [smp:12:12] [ds:12:12:1
0] [async-threads:1] [hipe] [dtrace]

Compiling 7 files (.ex)
warning: Map.size/1 is deprecated. Use Kernel.map_size/1 instead
  lib/todo/list.ex:13

Generated todo app
Interactive Elixir (1.9.0) - press Ctrl+C to exit (type h() ENTER for
 help)
iex(1)> Todo.System.start_link()
Starting database worker 1
Starting database worker 2
Starting database worker 3
Starting to-do cache
{:ok, #PID<0.185.0>}
iex(2)> bobs_list = Todo.Cache.server_process("Bob's list")
Starting to-do server for Bob's list
#PID<0.194.0>
iex(3)> Todo.Cache.server_process("Bob's list")
#PID<0.194.0>
iex(4)> alices_list = Todo.Cache.server_process("Alice's list")
Starting to-do server for Alice's list
#PID<0.197.0>
```


dynamic_workers — beam.smp -- -root /usr/local/Cellar/erlang/22.0....

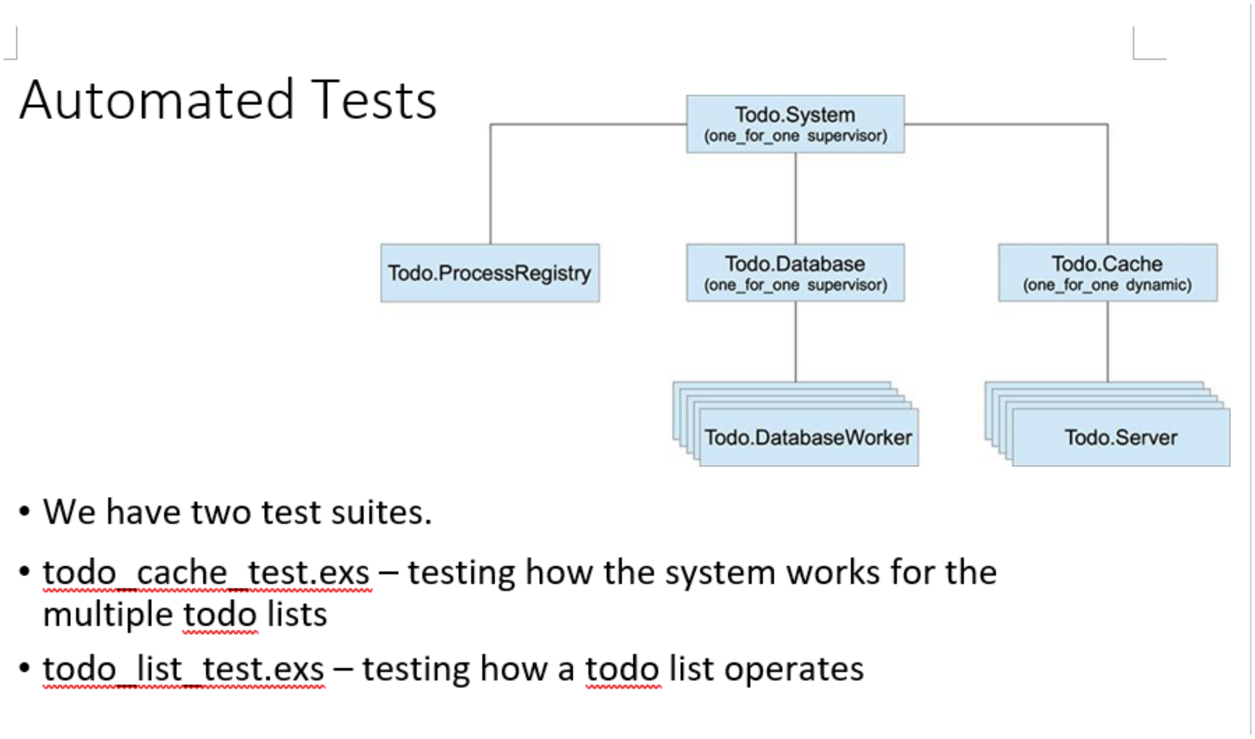
```
(base) Olivers-MacBook-Pro:dynamic_workers oliverau$ iex -S mix
Erlang/OTP 22 [erts-10.4.4] [source] [64-bit] [smp:12:12] [ds:12:12:1
0] [async-threads:1] [hipe] [dtrace]

Compiling 7 files (.ex)
warning: Map.size/1 is deprecated. Use Kernel.map_size/1 instead
  lib/todo/list.ex:13

Generated todo app
Interactive Elixir (1.9.0) - press Ctrl+C to exit (type h() ENTER for
 help)
iex(1)> Todo.System.start_link()
Starting database worker 1
Starting database worker 2
Starting database worker 3
Starting to-do cache
{:ok, #PID<0.185.0>}
iex(2)> bobs_list = Todo.Cache.server_process("Bob's list")
Starting to-do server for Bob's list
#PID<0.194.0>
iex(3)> Todo.Cache.server_process("Bob's list")
#PID<0.194.0>
iex(4)> alices_list = Todo.Cache.server_process("Alice's list")
Starting to-do server for Alice's list
#PID<0.197.0>
```

1. We mimic the crashing of Bob’s process.

2. If we ask for Bob’s list again, a new process was created so a new ID was returned.

3. If we ask for Alice’s list, no new process was created so the previous ID was returned.

4. Processes handling different todo lists are independent.



Automated Tests

- We have two test suites.
- todo_cache_test.exs – testing how the system works for the multiple todo lists
- todo_list_test.exs – testing how a todo list operates

Functionality Suggestions

1. Create a nice report summary (daily or weekly)

2. Add time in addition to date

3. Track time spent as well as planning

4. User authentication

5. Sharing todo list (read and/or write)

A. Choose your modifications wisely; some may take more effort than others.

B. You can do more than one change if you have time.

Efficiency or capacity

- Efficiency means doing it faster
- Capacity means the ability to handle more workload or data
- Sometimes they can be achieved by having more processes working in parallel
- Sometimes they can be achieved by revising the algorithms used

Fault Tolerance

- The sample project already possesses some fault tolerance as demonstrated in the killing of a todo server.
- Can you test the fault tolerance to other kind of processes?
- If it is not fault tolerant, can you improve it?
- Is there limitation in the existing fault tolerance? For example, a limit of 3 failures in 5 minutes. Can you find the limitation and relax it a little.

Automated Testing

- This is the area that you can enhance most easily because the sample test code is not comprehensive.
- For example, it tested the fault tolerance of a crashed todo server in a minimal way. It did not test the fault tolerance of a crashed database worker.

Originality

- If your modification is different from classmates' work, you are considered original.
- Will good idea be similar any way? Even if functionality is the same, you design the API more elegant than others.

Submission Requirements

1. All changes

A. must have an objective and explanation of how it works especially if the code is difficult to understand.

B. must be demonstrated through either screens of running at the prompt line or automated tests

C. must be identified by the line numbers being replaced with file name

2. Must my modifications be an improvement?

A. Preferably your modification is a clear improvement.

B. If you cannot find an improvement in an area, at least make some changes and explain the effect of your changes.