A BEGINNERS GUIDE TO ERLANG

ABOUT ME

‣ John-Paul Bader

‣ http://twitter.com/hukl

‣ http://github.com/hukl

‣ http://smyck.net

Wooga

http://www.slideshare.net/wooga/presentations

Show, don't tell: Seeing is believing

Lean back and enjoy the shows, but be careful: Going back to your former treadmill after being exposed to these movies can prove exceedingly painful. Don't say we didn't try to warn you.

```
def create
  @post = Post.new(params[:post])
  if @post.save
    flash[:notice] = 'Post was successfully
    redirect_to :action => 'list'
  else
    render :action => 'new'
  end
end

def edit
  @post = Post.find(params[:id])
end

def update
  @post = Post.find(params[:id])
  if @post.update_attributes(params[:post])
```

Creating a weblog in 15 minutes

Creating a weblog in 15 minutes

In 15 minutes, we go from scratch to complete weblog engine: with comments and an administrative interface. But since the actual application only took 58 lines to complete, we also have time left over to do unit testing, examine the logs, and play around with the domain model.

BOOKS

‣ Programming Erlang

‣ Erlang Programming

‣ OTP in Action

‣ Learn you some Erlang for great Good

BOOKS

‣ Programming Erlang

‣ Erlang Programming

‣ OTP in Action

‣ Learn you some Erlang for great Good

USE REBAR!

‣ Build Tool for Erlang

‣ Dependency Management
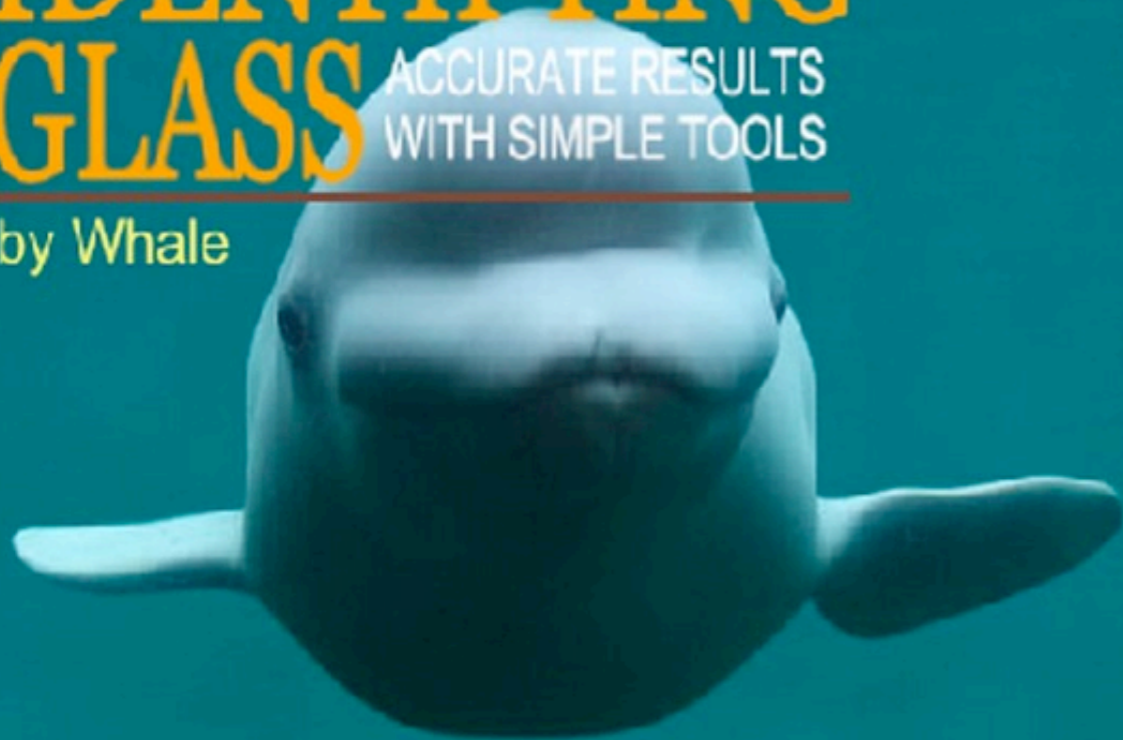
‣ https://github.com/basho/rebar

OTP RELEASES

OTP RELEASES

‣ Skip if you own / operate the infrastructure where the app is running

‣ Skip if you deploy frequently

‣ Use if your customer owns / operates the infrastructure

‣ Use if you ship the app to the customer and updates are rare and huge

TESTS

- ‣ eunit
- ‣ common test
- ‣ etest
  etest_http

ETEST

```erlang
-module(my_api_test).
-compile(export_all).
-include_lib("etest/include/etest.hrl").
-include_lib("etest_http/include/etest_http.hrl").

before_suite() -> application_start(my_api).
after_suite()  -> application_stop(my_api).

test_hello_world() ->
    Response = ?perform_get("http://localhost:3000/hello"),
    ?assert_status(200, Response),
    ?assert_body_contains("Hello", Response),
    ?assert_body("Hello World", Response).
```

Terminal

zsh   vim

SHARE

```erlang
1 -module (my_fancy_test).
2 -compile (export_all).
3
4 % etest macros
5 -include_lib ("etest/include/etest.hrl").
6 % etest_http macros
7 -include_lib ("etest_http/include/etest_http.hrl").
8
9
10 test_basic_operation() ->
11     ?assert_equal(1, 1).
12
13
14 test_hello_world() ->
15     Response = ?perform_get("http://localhost:3000/hello/world"),
16     ?assert_status(200, Response),
17     ?assert_body_contains("Hello", Response),
18     ?assert_body("Hello World", Response).
```

http://vimeo.com/hukl/10-minutes-erlang

10:56

"test/my_fancy_test.ert" 18L, 445C written

HD

How to create a HTTP API with Erlang in 10 Minutes

SINGLE RESPONSIBILITY PROCESSES

STRINGS & TIME

‣ Text Manipulation: Do it somewhere else

‣ Time Operations: Gregorian / Half-Baked

STRINGS & TIME

```
(Time.now - 3600).sunday?
=> true


> 3.months.ago.end_of_week.midnight
=> Sun, 18 Nov 2012 00:00:00 CET +01:00


"Hello World is overrated".gsub(/o/) do |match|
  match += "ooo"
end
=> "Helloooo Woooorld is oooooverrated"
```

SYNTAX ...

SYNTAX …

‣ Not as bad as you initially think

‣ Do not blindly follow emacs mode dogma

‣ Do not hide more than necessary

‣ Try to structure your code in the most readable way

QUESTIONS?

LICK HERE!

●

(you may be one
of the lucky 25)