

What's new with Nitrogen Web Framework



Rusty Klophaus

Follow @rklophaus

rklophaus@gmail.com

"Nitrogen is by far one of the most fascinating web frameworks I've ever seen."

- Marc Held

"Future = Erlang + Nitrogen"

- Tom Lackner

"I've done more quicker with Erlang and Nitrogen than I've done with any other framework I've used, and I've used a lot... Rails, Catalyst, Mason, Django, Zope..."

- Jeremy Wall

"I find your use of records inspirational."

- Joel Reymont

Thanks!

Russell Brown

Tom McNulty

Jon Gretar

Torbjorn "Tobbe" Tornkvist

Marius A. Eriksen

Robert Schonberger

Chris Williams

Laurent Ostiz

Benjamin Nortier

Dave Peticolas

Dan Bravender

Martin Scholl

Michael Mullis

Jay Doane

Ville Koivula

Joel Reymont

David Svanlund



Focus



Focus on Nitrogen's strengths.

**Let somebody else
handle the rest.**

S minutes

on Nitrogen's strengths

Bring Web 2.0 to Erlang.
Keep it **magical**

*Note to self:
Do the tasklist.*

Web **Applications**
Not Web **Sites**

Elements
Spit out HTML.

and

Actions
Do everything else.

Bring Web 2.0 to Erlang.

Keep it **magical**

*Note to self:
Deno the tasklist.*



My Business Plan

- Step 1 - Write an Erlang application [Delete](#)
- Step 2 - ? [Delete](#)
- Step 3 - Profit [Delete](#)

[Use default business plan](#)

Web Applications

Not Web Sites

Elements

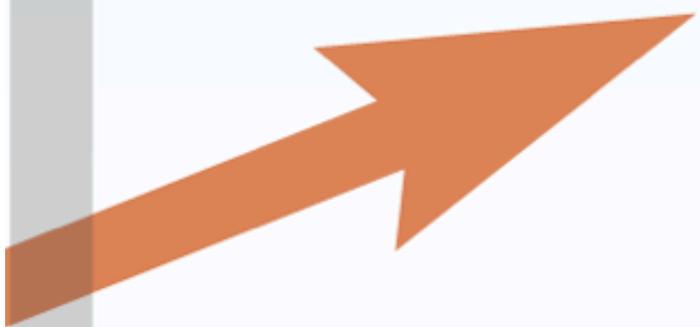
Spit out HTML.

and

Actions

Do everything else.

code samples!



* sample1.erl * sample2.erl * sample3.erl

```
1 main() ->
2   [
3     #panel { id=mypanel },
4     #button { text="Click" }
5   ].
6
7 event(_) -> ok.
```

```
1 main() ->
2   [
3     #panel { id=mypanel },
4     #button { text="Click", postback=clicked }
5   ].
6
7 event(clicked) ->|
8   wf:update(mypanel, [
9     #span { text="The button was clicked" }
10  ]).
11
```

```
1 main() ->
2   [
3     #panel { id=mypanel },
4     #button { text="Click", postback=clicked }
5   ].
6
7 event(clicked) ->
8   wf:update(mypanel, [
9     #span { text="The button was clicked" }
10  ]),
11
12  wf:wire(mypanel, [
13    #effect { effect=highlight }
14  ]).
```

What else?

Comet/Async Events

Run Erlang code upon Javascript events

Meta-Lang for Creating Pages

Drag-and-Drop

Maintain state between postbacks

Templates

Tag events with arbitrary data

Sorting

Data Binding

Ajax Inserts and Updates

Validation

Focus on Nitrogen's strengths.

**Let somebody else
handle the rest.**

Let somebody else
handle the rest.

logging

REST API

config settings

process registry

The Rest

security

MVC

caching

sessions

routing

How?

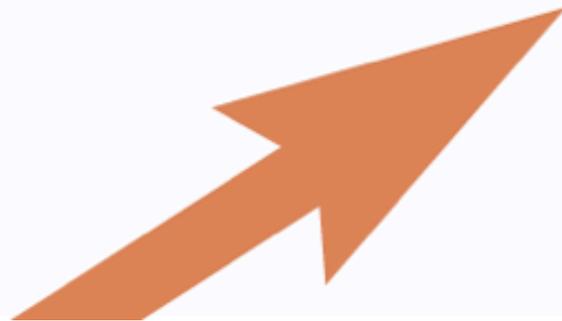
Run Nitrogen
inside of an
existing Erlang
Web Framework

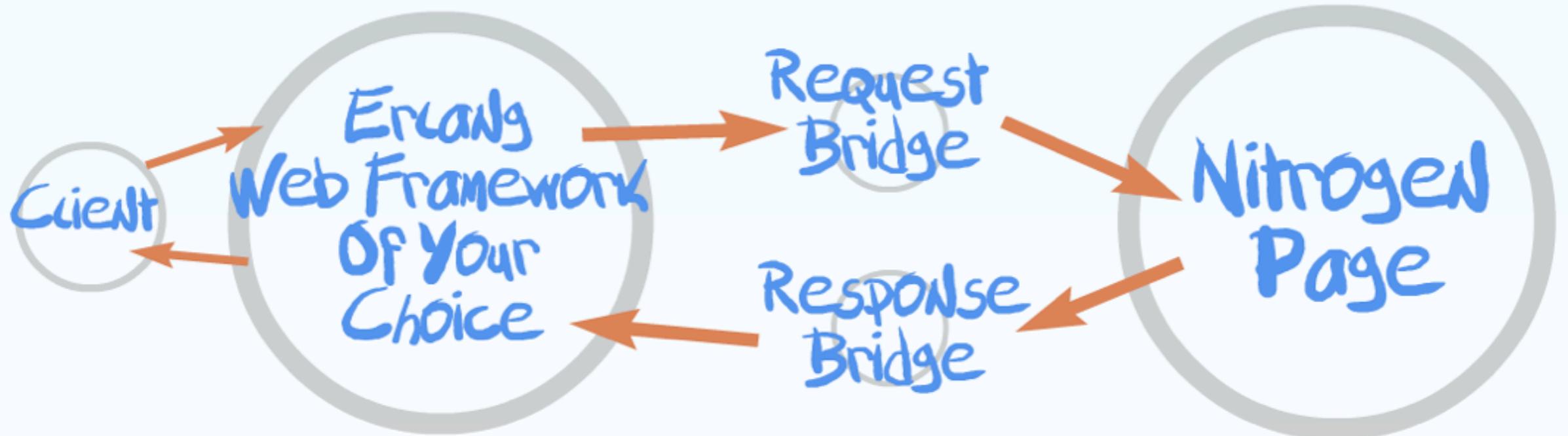
Create a formal
way for others to
extend the guts of
Nitrogen.

Run Nitrogen
inside of an
existing Erlang
Web Framework

introducing simple_bridge

- http request/response interface
- adapters for Inets, Mochiweb, Yaws, Webmachine
- coming soon: Erlang Web, ErlyWeb, Beep Beep





Note to self:
Show these five folks the
simple_bridge interface

* simple_bridge_request.erl

```
1 % Simple Bridge
2 % Copyright (c) 2008-2009 Rusty Klophaus
3 % See MIT-LICENSE for licensing information.
4
5 -module (simple_bridge_request).
6 -export ([
7     make/2,
8     behaviour_info/1
9 ]).
10
11 make(Module, Req) ->
12     Req1 = Module:init(Req),
13     simple_bridge_request_wrapper:new(Module, Req1).
14
15 behaviour_info(callbacks) -> [
16     {init, 1},           % Should accept the request value passed by the http server.
17
18     {request_method, 1}, % GET, POST, etc.
19     {path, 1},          % The path. (http://server.com/<PATH>?querystring)
20
21     {headers, 1},       % Return a proplist of headers, key and value are strings.
22     {cookies, 1},       % Return a proplist of cookies, key and value are strings.
23     {query_params, 1},  % Return a proplist of query parameters, key and value are strings.
24     {post_params, 1},   % Return a proplist of post parameters, key and value are strings.
25
26     {peer_ip, 1}        % The remote IP address
27 ];
28
29 behaviour_info(_) -> undefined.
30
31
32
```

* simple_bridge_request.erl * simple_bridge_response.erl

```
1 % Simple Bridge
2 % Copyright (c) 2008-2009 Rusty Klophaus
3 % See MIT-LICENSE for licensing information.
4
5 -module (simple_bridge_response).
6 -include ("simplebridge.hrl").
7 -export ([
8     make/2,
9     behaviour_info/1
10 ]).
11
12 make(Mod, Req) ->
13     simple_bridge_response_wrapper:new(Mod, Req, #response{}).
14
15 behaviour_info(callbacks) -> [
16     {build_response, 2}
17 ];
18
19 behaviour_info(_) -> ok.
```

Create a formal
way for others to
extend the guts of
Nitrogen.

introducing handlers

pluggable modules to interact with
resources used by Nitrogen

Note to self:
Now show the handler interface.

Handlers

log

process_registry

cache

session

state

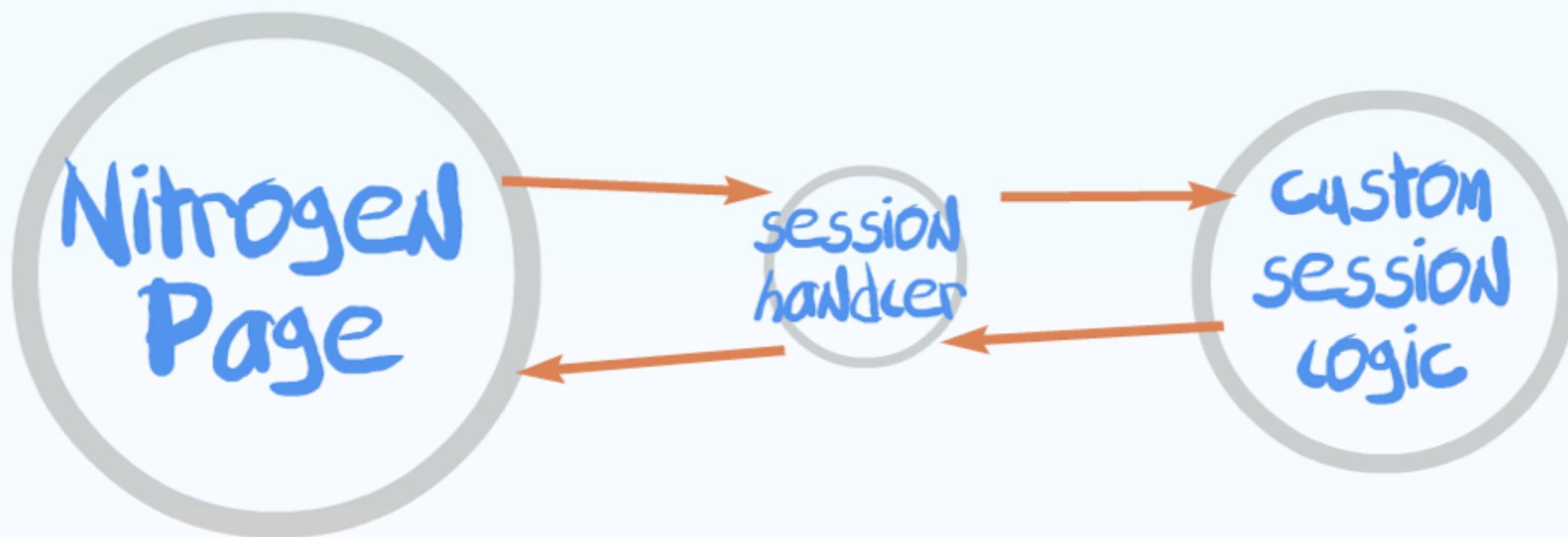
identity

role

route

security

output



* session_handler.erl

```
1 % Nitrogen Web Framework for Erlang
2 % Copyright (c) 2008-2009 Rusty Klophaus
3 % See MIT-LICENSE for licensing information.
4
5 %
6 % The session_handler provides a place to store values on the server
7 % between requests.
8 %
9 % An application can define a custom session handler to control
10 % how Nitrogen manages session values.
11
12 -module (session_handler).
13 -export ([
14     behaviour_info/1, get_value/2, get_value/3, set_value/3, clear_value/2, clear_all/1
15 ]).
16
17 % Example Session Handler Interface
18 behaviour_info(callbacks) -> [
19     {init, 2},
20     {finish, 2},
21     {get_value, 4},
22     {set_value, 4},
23     {clear_value, 3},
24     {clear_all, 2}
25 ];
26 behaviour_info(_) -> undefined.
27
```

Focus on Nitrogen's strengths.

**Let somebody else
handle the rest.**

Thanks!

<http://www.nitrogenproject.com>

Twitter: @rklophaus

Email: rklophaus@gmail.com

What's new with
Nitrogen
Web Framework

