Web development in Lua

Introducing Sailor, an MVC web framework

Etiene Dalcol
@etiene_d
Sailor!
sailorproject.org
Google Summer of Code

LabLua
Lua Ladies

lualadies.org
Lua Space
lua.space
Web dev in Lua
Sailor
The future
Web dev in Lua

Sailor

The future
Average across all speed tests

(less is better)

Pretty fast

Super fast!

Servers

- Apache: mod_lua
- Nginx: OpenResty
Servers

- Apache: mod_lua
- Nginx: OpenResty
Servers

- Apache: mod_lua
- Nginx: OpenResty
- Xavante
- Others: Lighttpd, Lwan, Pegasus, Mongoose
Frameworks

Micro-frameworks
Lapis

MVC
Orbit

Event-driven
Luvit
TurboLua

Others
Ophal, Sputnik, LuaPress, Tir, Vanilla

http://lua.space/webdev/the-best-lua-web-frameworks
Web dev in Lua
Sailor
The future
What exactly is Sailor?

• It’s an MVC web framework
• Completely written in Lua
• Compatible with Apache (mod_lua), Nginx (OpenResty), Xavante, Mongoose, Lighttpd and Lwan
• Compatible with Linux, Windows and Mac
• Compatible with different databases
• MIT License
• v0.5 (Pluto)
• Planning next release to be a 1.0!
What (else) is cool about Sailor?

- Routing and friendly URLs
- Session, cookies, include, redirect...
- Lua Pages parsing
- Mail sending
- Simple Object Relational-Mapping
- Validation (valua)
- Basic login and authentication modules
- Form generation
- Themes (Bootstrap integration out of the box)
- App generator (Linux and Mac only)
- Model and CRUD generator
- Automated tests
What (else) is cool about Sailor?

• Routing and friendly URLs
• Session, cookies, include, redirect...
• Lua Pages parsing
• Mail sending
• Simple Object Relational-Mapping
• Validation (valua)
• Basic login and authentication modules
• Form generation
• Themes (Bootstrap integration out of the box)
• App generator (Linux and Mac only)
• Model and CRUD generator
• Automated tests
• Lua at client
Not so great things

• It’s still in early development
• Things are changing fast
• It lacks features
• Documentation
How to get Sailor!

$ luarocks install sailor
$ sailor create 'My App' /var/www
$ cd /var/www/my_app
$ lua start-server.lua
Welcome to your first Sailor application

This is the default view file, you will find it under /views/main/index.lp. It is rendered by a controller, you will find it under /controllers/main.lua. This layout is provided by Twitter Bootstrap, an open source front-end framework. You can use this layout as is, modify it, or get new layouts and place them under /layouts/. Further documentation of Sailor MVC can be found here.
App structure

/conf
/controllers
/models
/pub
/runtime
/tests
/themes
/views
```lua
-- /controllers/site.lua

local site = {}

function site.index(page)
    local msg = "Hello World"
    page:render('index', { msg = msg })
end

function site.notindex(page)
    page.theme = nil
    page:write("I'm different!")
end

return site
```
A message from the server:

```lua
page:print(msg)
```

The message again:

```html
msg
```

<!-- syntactic sugar: same thing as above -->
A message from the server: Hello World
The message again: Hello World

I'm different!
Example!

```lua
-- Code here runs on the server
альн ограниченный доступ

-- Same as above

-- Runs at the client

-- Runs at the server and the client

-- Runs at the server and the client

another_msg = "Another message"

-- Runs at the client

window:alert(another_msg)
```
The page at localhost:8080 says:
Another message
local user = {}
local v = require "valua" -- validation module

user.attributes = {
  { id = "safe" },
  { name = v:new().not_empty().len(6,50) }
}
user.db = {
  key = 'id',
  table = 'users'
}
user.relations = {
  posts = { -- u.posts
    relation = "HAS_MANY", model = "post", attribute = "author_id"
  }
}
return user
local user = {}
local v = require "valua" -- validation module

user.attributes = {
    { id = "safe" },
    { name = v:new().not_empty().len(6,50) }
}

user.db = {
    key = 'id',
    table = 'users'
}

user.relations = {
    posts = { -- u.posts
        relation = "HAS_MANY", model = "post", attribute = "author_id"
    }
}

return user
local site = {};

function site.index(page)
    local User = sailor.model('user')
    local u = User:new()
    u.name = 'Arnold'
    local msg
    if u:save() then
        msg = 'Success'
    else
        msg = table.unpack(u.errors)
    end
    local users = User:find_all()
    page:render('index', { msg = msg, users = users })
end

return site
Web dev in Lua

Sailor

The future
Rails Girls
Summer of Code
bit.ly/luawebdev
Sailor! A Lua MVC framework.

Signup to our mail list to be updated on our releases or if you need support

Google Group / Mail List

sailorproject.org
github.com/sailorproject
dalcol@etiene.net
@etiene_d
Sailor! A Lua MVC framework.

Signup to our mail list to be updated on our releases or if you need support.

Google Group / Mail List

sailorproject.org
github.com/sailorproject
dalcol@etiene.net
@etiene_d
gitter.im/sailorproject/sailor
Sailor! A Lua MVC framework.

Signup to our mail list to be updated on our releases or if you need support

Google Group / Mail List

- **Support transactions**
  - #40 opened on May 17 by Etienne

- **Integrate sailor with other Lua based web servers such as pegasus**
  - #39 opened on May 2 by Etienne

- **Setting up a password to use the autogenerated module**
  - #38 opened on May 2 by Etienne

- **Provide user model with app**
  - #37 opened on May 2 by Etienne

- **Verify why page:inspect works differently on Apache and Xavante**
  - #36 opened on May 1 by Etienne

sailorproject.org
github.com/sailorproject
dalcol@etiene.net
@etiene_d