

THE WEB FRAMEWORK OF YOUR DREAMS.

Designed for developers by Giant Squid.

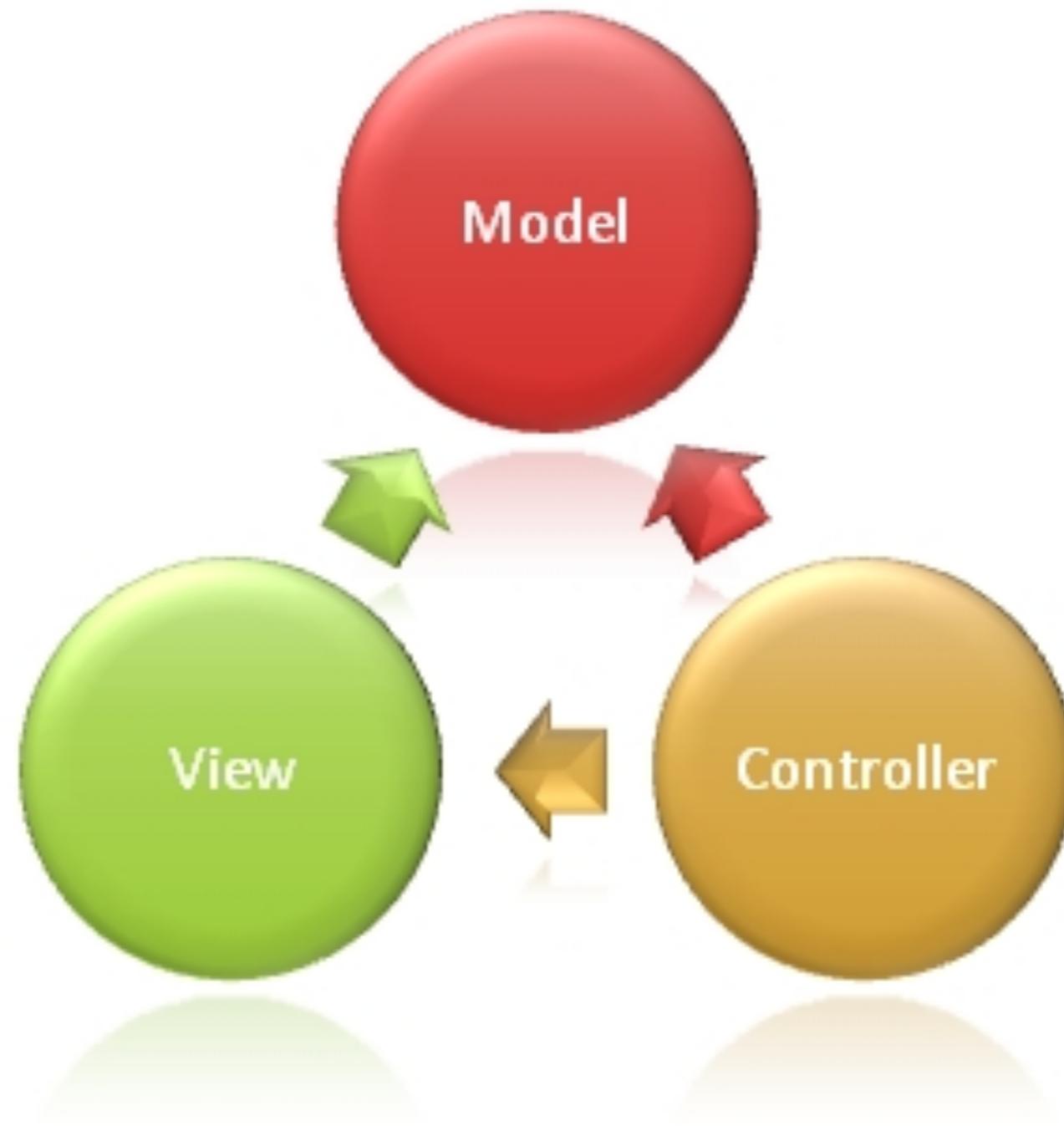
Get started

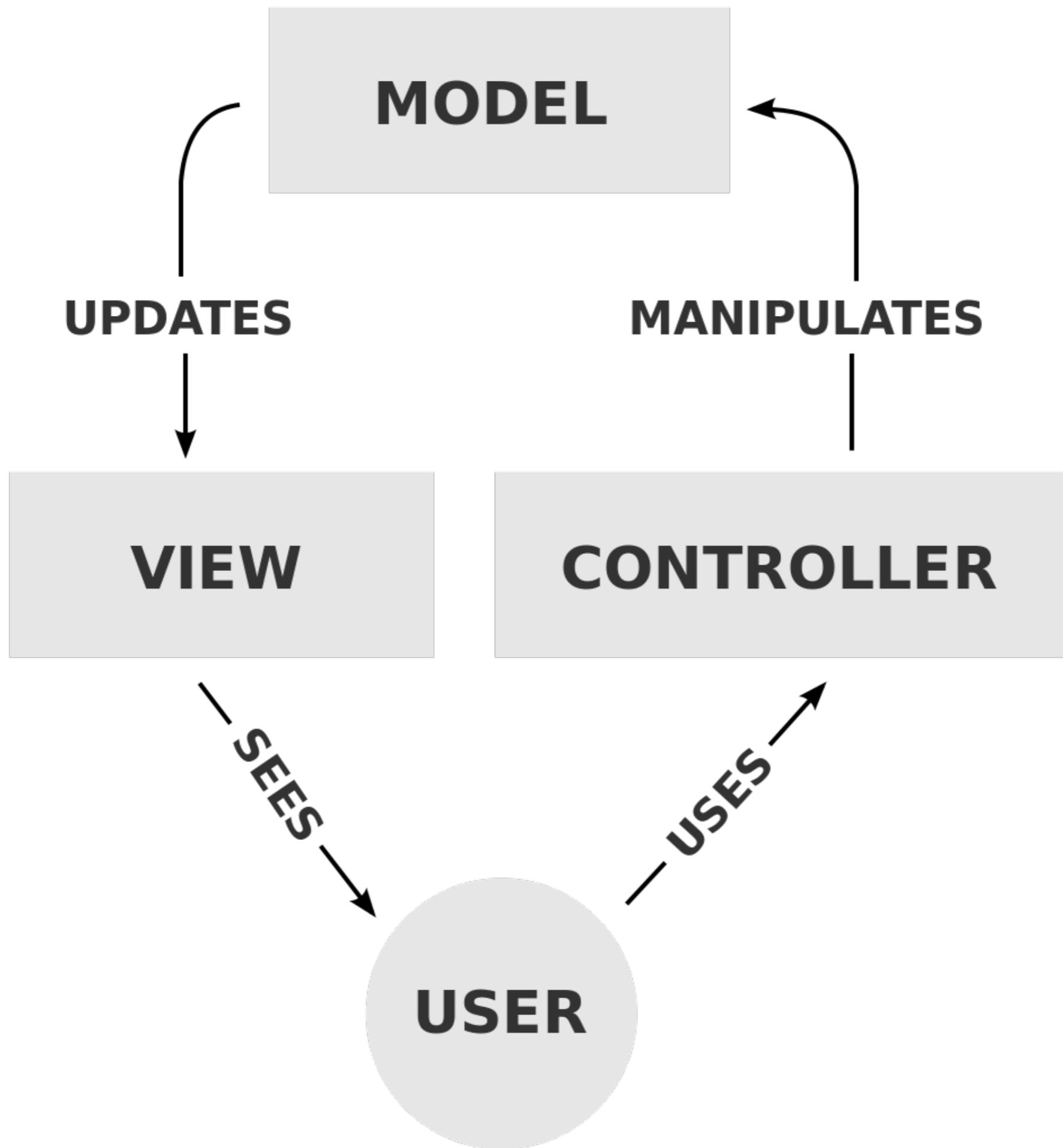


MVC way to introduce sails.js - Caesar Chi

<http://sailsjs.org/#!>

Web, MVC







Rails like framework, but extends node.js feature



<http://sailsjs.org/>

Sails.js

- Auto Routing
- Express base
- Scaffolding
- Socket.io support
- Restful blueprint

Support
and
Enhance

express
3.0.0
web application
framework for
node

express

web application framework for node

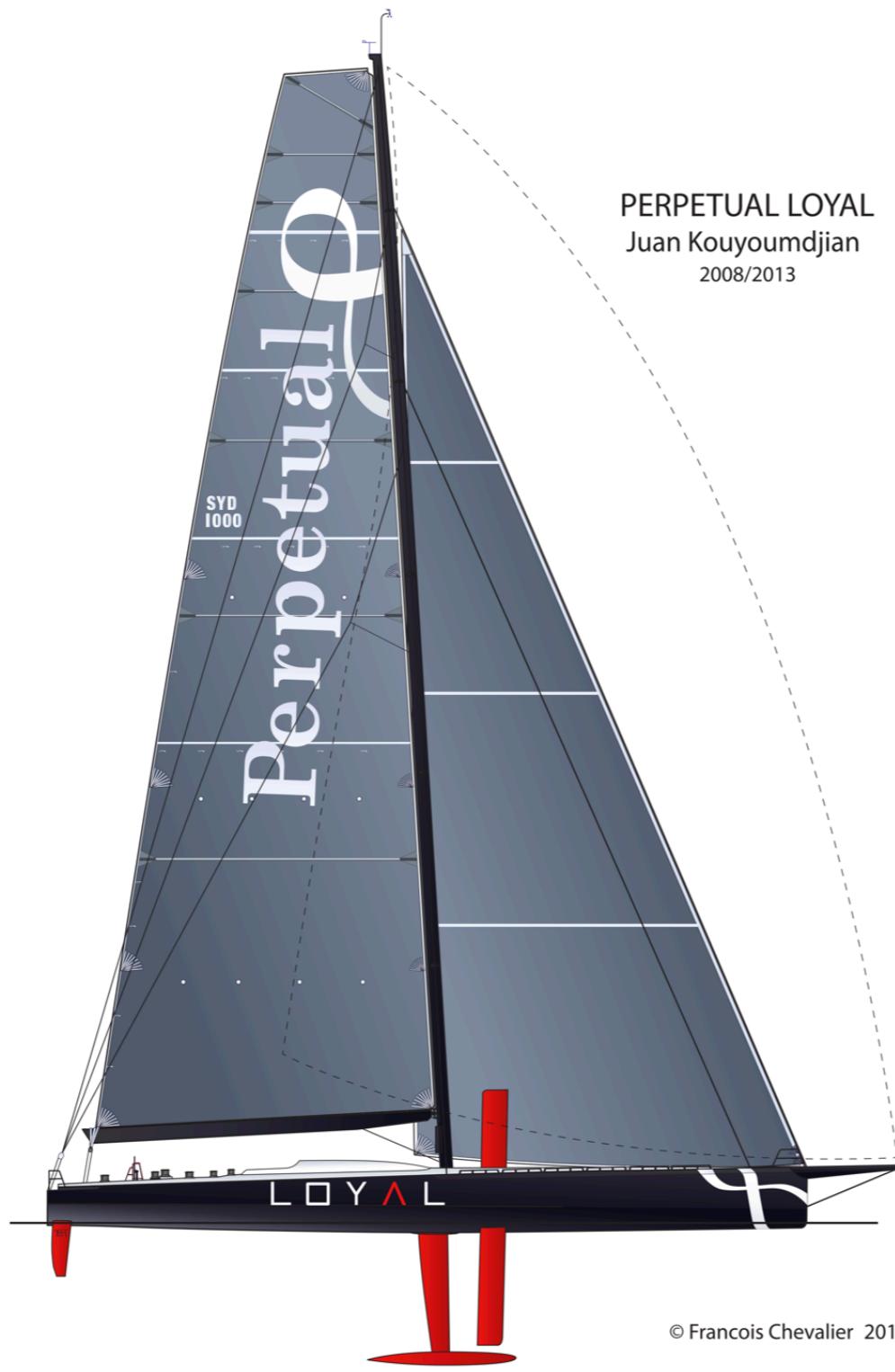
Dependents
(1000)

claude, derby-auth, frakture-objects, fast-api, blab,
gitdeploy, cadigan, gaia-screenshot, cloudq,
express-logger, angular-doc, doa, consoloid-
server, crafity-webserver, express-restify-
mongoose, creatary, chateau, coffee-service, breti-
ics665-expressjs, amid,
and 980 more



Blueprint default support, you do not key in CRUD

<https://github.com/hankejh/blueprint>



PERPETUAL LOYAL
Juan Kouyoumdjian
2008/2013

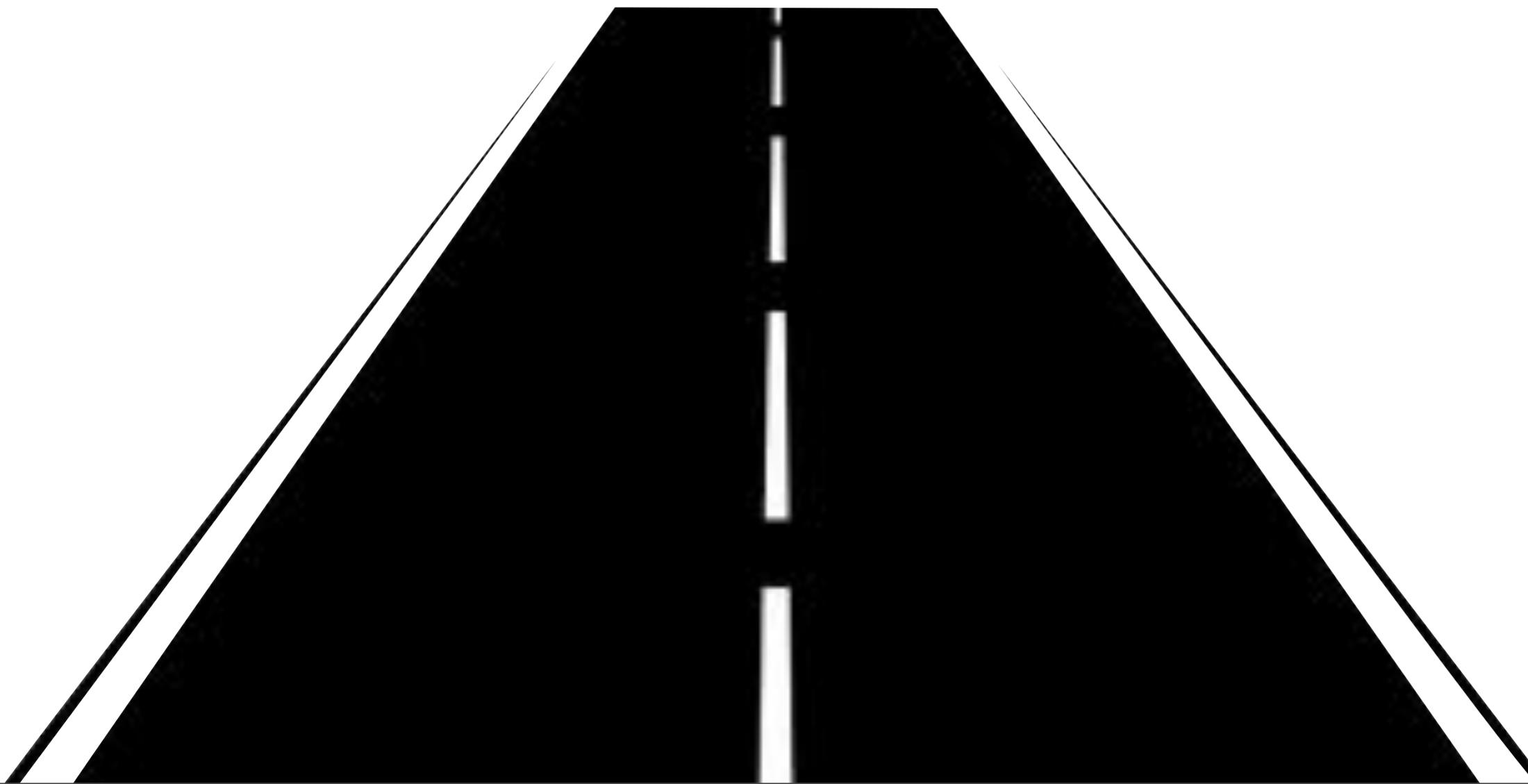
sails 

The word "sails" is written in a bold, dark blue sans-serif font. To the right of the text is a graphic element consisting of two overlapping shapes: a teal semi-circle on top and a teal triangle below it, both pointing upwards and to the right.

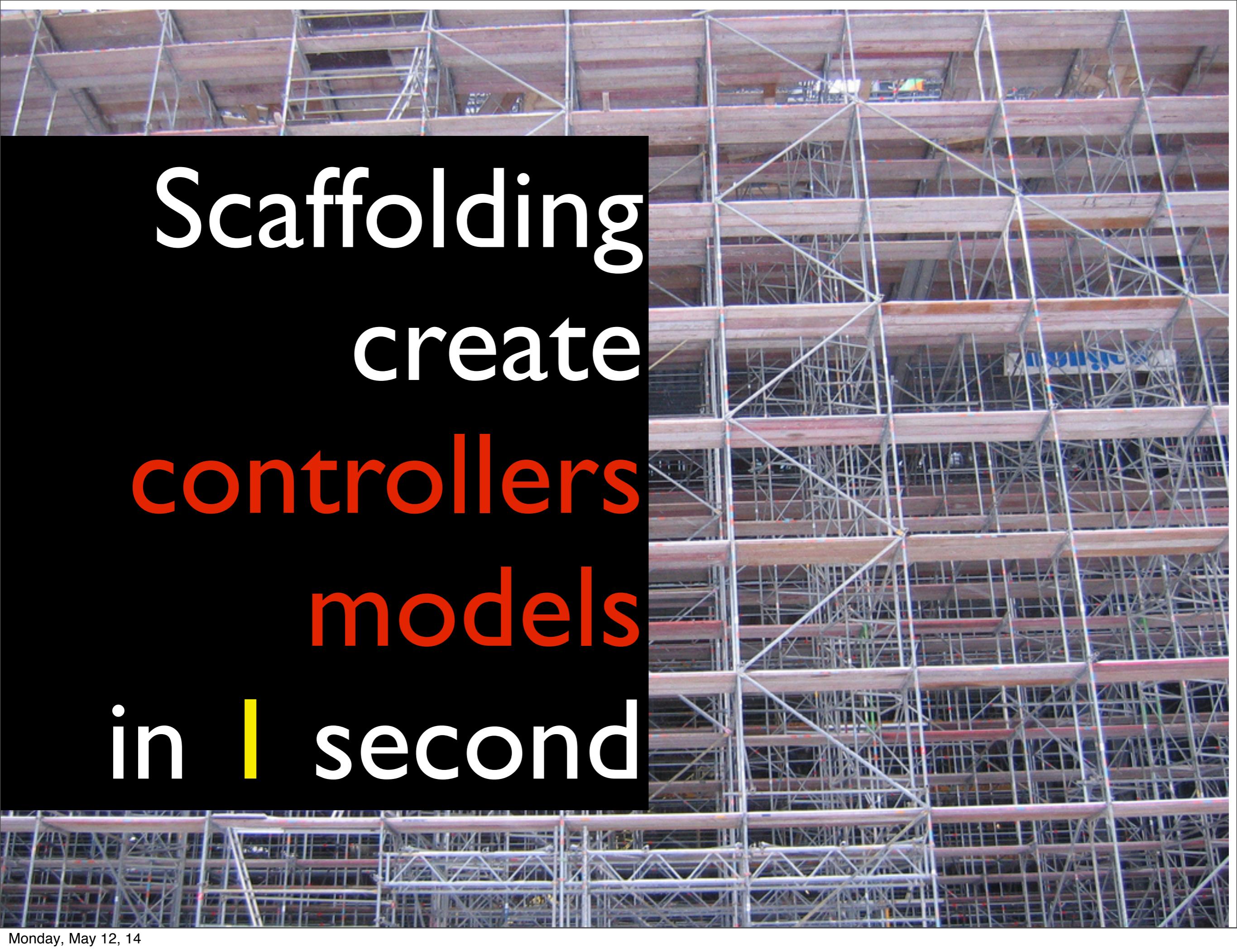
Model module, use Waterline

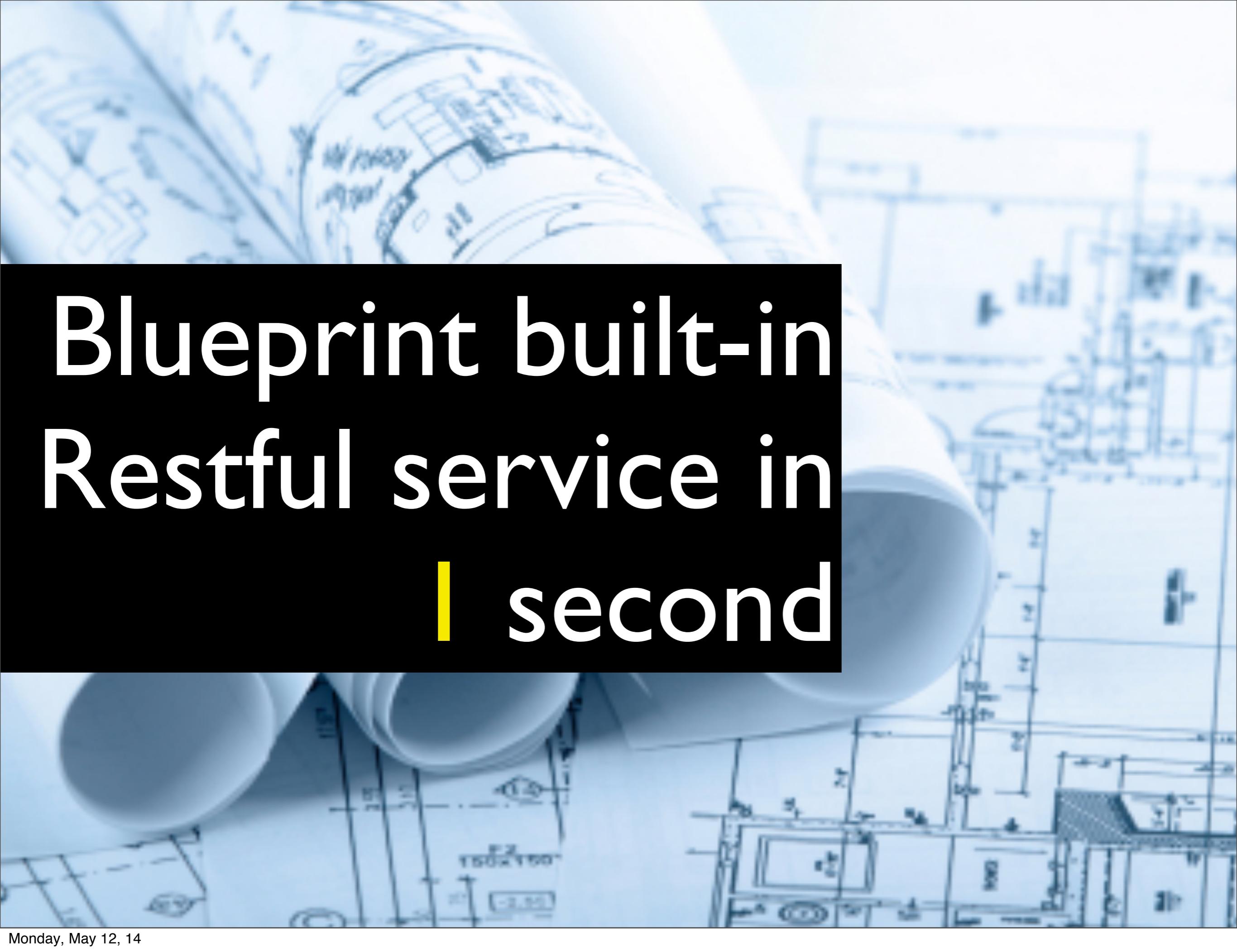
<https://github.com/balderdashy/waterline>

Route Setting
is
*ucking EASY



Scaffolding
create
controllers
models
in 1 second





Blueprint built-in
Restful service in
| second

How to start?

cli

npm install -g sails

Quick start

Installation

To install the latest stable release with the command-line tool:

```
sudo npm -g install sails
```

Creating a New Sails Project

Create a new app:

```
sails new testProject
```

Now lift the server:

```
cd testProject  
sails lift
```

run app

- cd project
- node app
- sails lift

Sails.js command line

cli

Sails generate **Controller**
Model action ...

Generate

cli

sails generate

controller
model

name

blueprints include

get /:controller/:id?

post /:controller

put /:controller/:id

delete /:controller/:id

:controller/find/:id?

:controller/create

:controller/update/:id

:controller/destroy/:id

Generate(cli)

- Model
- path:
 - /api/model
- Controller
- Path:
 - /api/controller

Generate(cli)

- Model
- path:
- Controller
- Path:

Model, Controller will be global object

Controller

<http://sailsjs.org/#!documentation/controllers>

cli

sails generate controller Comment create destroy tag like

```
var CommentController = {
  create: function(req, res) {

  },
  destroy: function(req, res) {

  },
  tag: function(req, res) {

  },
  like: function(req, res) {

  }
}

module.exports = CommentController;
```

cli

sails generate controller Comment create destroy tag like

```
var CommentController = {
  create: function(req, res) {
    // Request, Response
  },
  destroy: function(req, res) {
    // Request, Response
  },
  tag: function(req, res) {
    // Request, Response
  },
  like: function(req, res) {
    // Request, Response
  }
}

module.exports = CommentController;
```

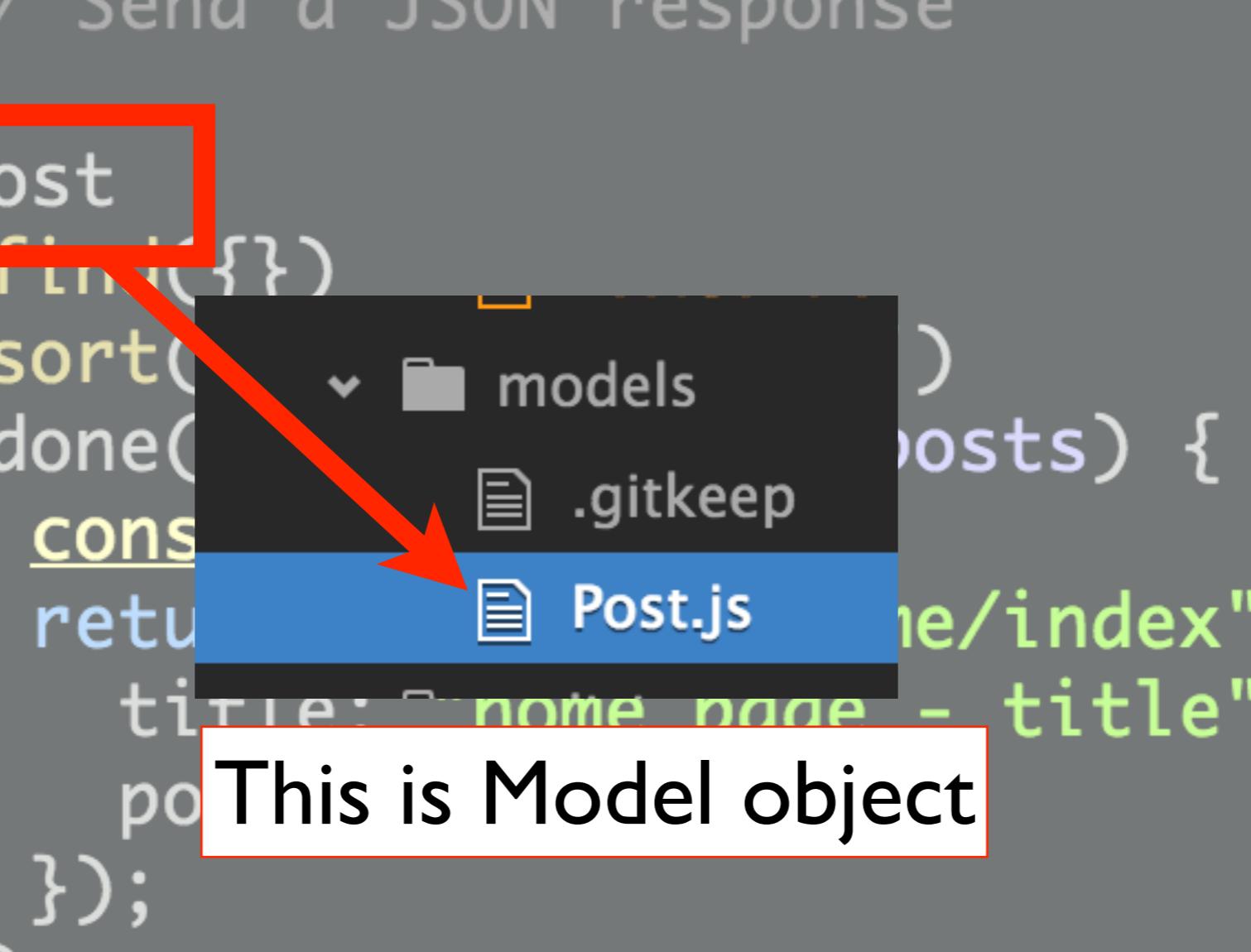
Request, Response

CommentController.js

```
list: function (req, res) {  
  // Send a JSON response  
  
  Post  
    .find({})  
    .sort('updatedAt DESC')  
    .done(function (err, posts) {  
      console.log(posts);  
      return res.view("home/index", {  
        title: "home page - title",  
        posts: posts  
      });  
    });  
},
```

CommentController.js

```
list: function (req, res) {  
    // Send a JSON response  
  
    Post  
.find({})  
.sort()  
.done(  
    const  
return  
    title: "name name - title",  
    post  
});  
});  
},
```



This is Model object

This is Model object

Controller

- `/api/controllers/`
- <http://sailsjs.org/#!documentation/controllers>

View

View

- View -> as a render function
- Path:
 - /view/...
- Support
 - ejs (default)
 - jade

View

- View -> as a render function
- Path:

ejs (default)

- Support
 - ejs (default)
 - jade

EJS

<% Embedded JavaScript %>

<%= An open source
JavaScript Template library %>

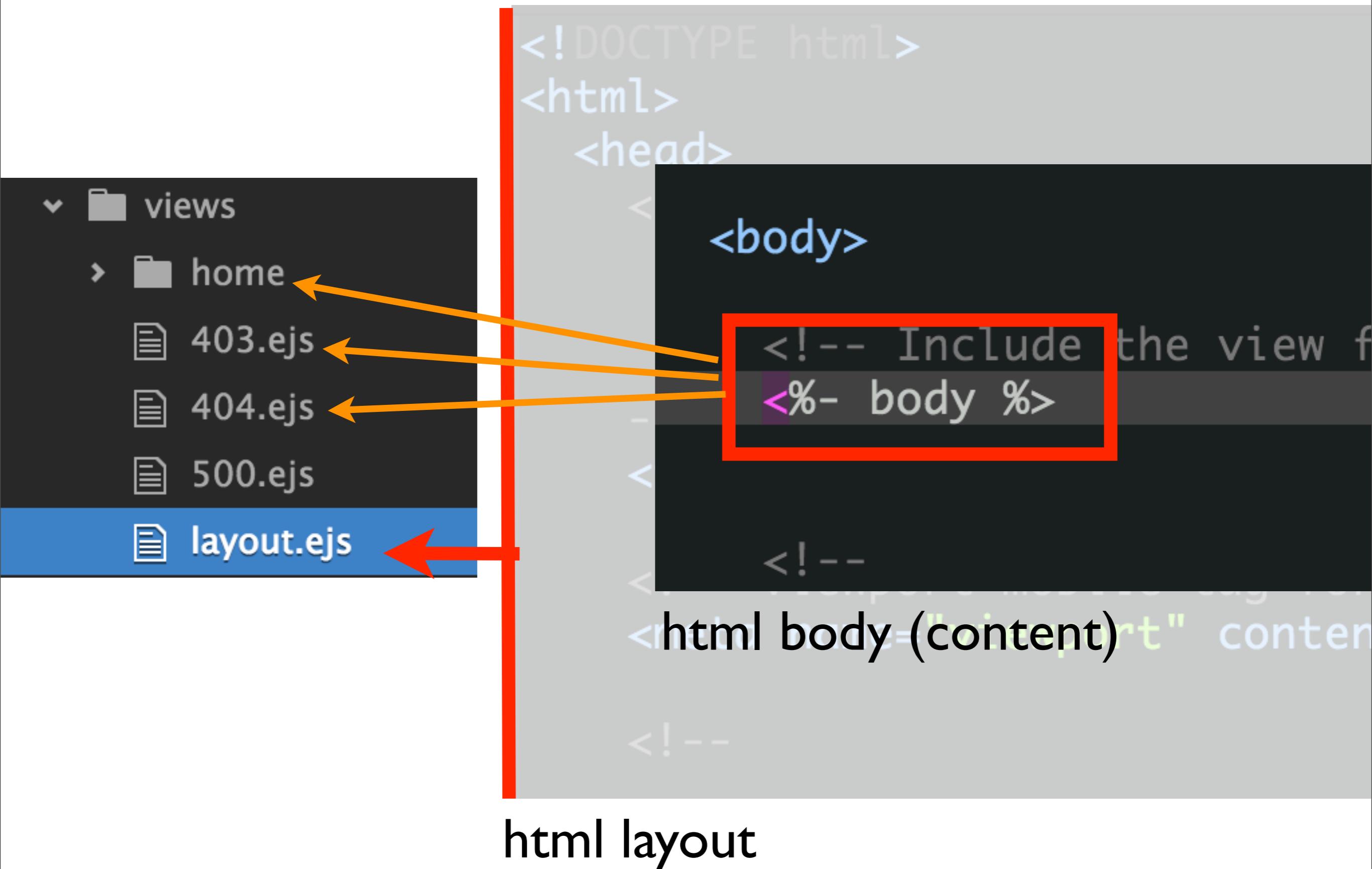
<http://embeddedjs.com/>

EJS

<% Embedded JavaScript %>

<%= An open source
JavaScript Template library %>

```
<ul>
<% for(var i=0; i<supplies.length; i++) {%
    <li><%= supplies[i] %></li>
<% } %
</ul>
```



controller

response.view()

view path

```
return res.view("home/index", {  
    title: "home page - title",  
    posts: posts  
});
```

controller

response.view()

parameters

```
return res.view("home/index", {  
    title: "home page - title",  
    posts: posts  
});
```

View Partials

/views/site/partials/footer.ejs

```
<footer>
  <span>
    <div class="float-right">
    </div>
    <div class="float-left">
      Balderdash &copy; 2013
    </div>
    <div class="cle
    </div>
  </span>
<footer>
```

/views/site/index.ejs

```
<%- partial('partials/header')%>
<%- partial('partials/page')%>
<%- partial('partials/footer') %>
```

Response view

- `res.view() = res.render();`
- `res.send({ some: 'json' });`
- `res.json({ user: 'tj' });`
- `res.redirect('/', 301);`

Route

<http://sailsjs.org/#!documentation/routes>

Route

- Route
- Path
 - /api/routes.js

```
module.exports.routes = {
  // To route the home page to the "index" action of the "home" controller:
  '/': {
    controller: 'home'
  },

  // Additional routes might look like:
  '/whateverYouWant': {
    controller: 'someController',
    action: 'someAction'
  },

  // If you want to set up a route only for a particular HTTP method/verb
  // (GET, POST, PUT, DELETE) you can specify the verb before the path:
  'post /signup': {
    controller: 'auth',
    action: 'signup'
  }
}
```

Route (default)

default follow blueprint

URL route

```
// URLs would also work
'/' : {
  controller: "post",
  action: "list"
},
```

Route (default)

default follow blueprint

```
// routes would also work
'/' . {
    controller: "post",
    action: "list"
},
```

Controller name

Controller action (function)

view point to route

URL route

View route

```
  '/': {  
    view: 'user/signup'  
  }
```

Route(default)

```
# Backbone Conventions
GET  :  /:controller          => findAll()
GET  :  /:controller/read/:id  => find(id)
POST :  /:controller/create    => create()
POST :  /:controller/create/:id => create(id)
PUT  :  /:controller/update/:id => update(id)
DELETE:  /:controller/destroy/:id => destroy(id)

# You can also explicitly state the action
GET  :  /:controller/find      => findAll()
GET  :  /:controller/find/:id   => find(id)
POST :  /:controller/create    => create(id)
PUT  :  /:controller/update/:id => update(id)
DELETE:  /:controller/destroy/:id => destroy(id)
```

Route and action

config/Route.js

```
'/' : {  
  controller: "post",  
  action: "list"  
},  
  
'/' : {  
  view: 'user/signup'  
}  
  
"GET /:controller/:action/:id?": {},
```

Model

Model

- file path -> api/models
- Capital naming
- Set schema and rule.
- ORM like
- you can use multi-adapter

example, api/models/Post.js

```
module.exports = {  
  
  attributes: {  
    title: {  
      type: 'string',  
      required: true  
    },  
    content: {  
      type: 'string',  
      required: true  
    }  
  }  
};
```

example, api/models/Post.js

```
module.exports = {  
  
  attributes: {  
    title: {  
      type: 'string',  
      required: true  
    },  
    content: {  
      type: 'string',  
      required: true  
    }  
  };
```

defined
key
&
attribute

example, controller call Post model

```
// send a JSON response
Post.create({
  title: title,
  content: content
}).done(function (err, post) {
  if (err) {
    return res.error();
  }

  req.flash('info', 'info: Create');
  res.redirect("/");
});
```

example, controller call Post model

model name



```
model name → Post.create({
  title: title,
  content: content
}).done(function (err, post) {
  if (err) {
    return res.error();
  }

  req.flash('info', 'info: Creat');
  res.redirect("/");
});
```

example, controller call Post model

model
behavior

```
// Some code here - response
Post.create({
  title: title,
  content: content
}).done(function (err, post) {
  if (err) {
    return res.error();
  }

  req.flash('info', 'info: Creat');
  res.redirect("/");
});
```

example, controller call Post model

new data

```
// send a JSON response
Post.create({
  title: title,
  content: content
}).done(function (err, post) {
  if (err) {
    return res.error();
  }

  req.flash('info', 'info: Creat');
  res.redirect("/");
});
```

example, controller call Post model

success
callback
&
response

```
// send a JSON response
Post.create({
  title: title,
  content: content
}).done(function (err, post) {
  if (err) {
    return res.error();
  }

  req.flash('info', 'info: Create');
  res.redirect('/');
});
```

Model document

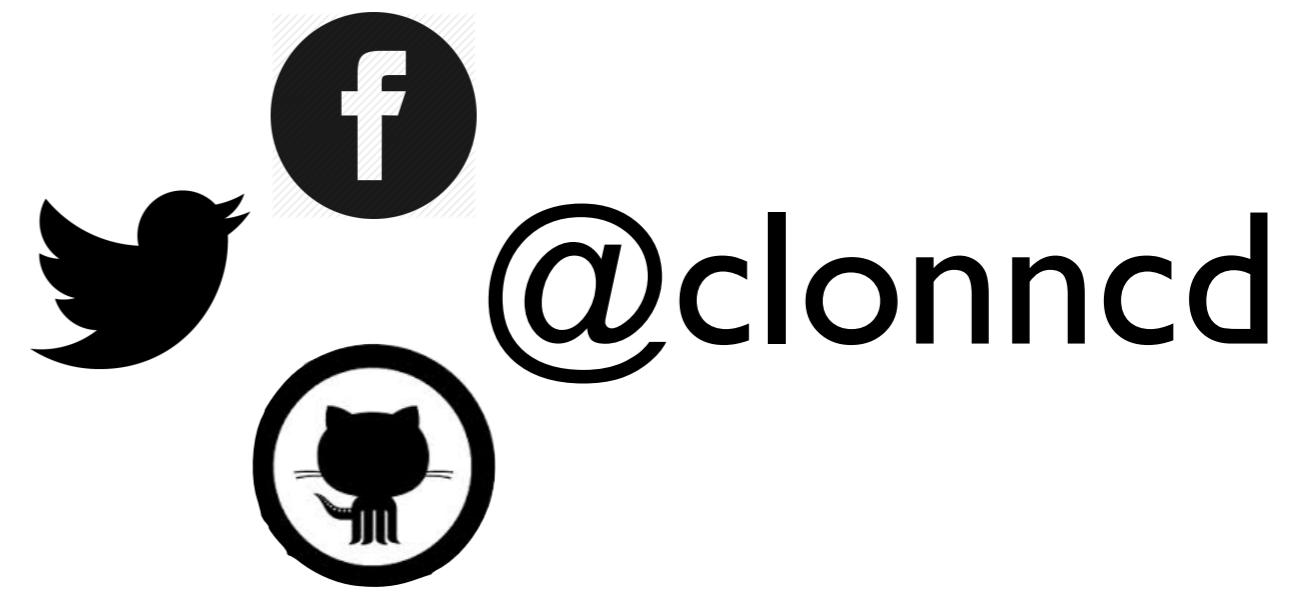
<http://sailsjs.org/#/documentation/models>



Controller, Model, Service, will be Global object

MVC way example

<https://github.com/nodejs-tw/example-week3>



Sails.js

- sails.js version is 0.9.x
- the model part and service, we will talk in another chapter.
- keep coding, be passions.