

Introduction to Sails.js

yungshin

About Me

- yungshin
- 大學化學系念不下去，碩班跳坑念資工
- III IDEAS
- 上班爬網站，下班寫網站
- 喜愛鑽研前端技術



投資新技術有賺有賠, 使用前請先詳閱
公開說明書(官方文件)

THE WEB FRAMEWORK OF YOUR DREAMS.

Designed for developers by Giant Squid.

Get started



Sails.js

- Web MVC Framework for Node.js
- Auto Routing
- Express based
- Scaffolding

Sails.js

- Socket.io support
- Restful blueprint
- Model module, ORM, use Waterline

MVC

- M - Model
- V - View
- C - Controller

MVC

- User see the views
- User use controllers
- Controllers manipulate models
- Models update views

Get Started

Get Started

- Install Command-line tools

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

Get Started

- Create a new app:

```
sails new testProject
```

- Now lift the server:

```
cd testProject  
sails lift
```

- See the default home page:

```
http://localhost:1337/
```

Sails.js Command Line

- Generate Model and Controller
 - `sails generate [Name]`
- Generate Controller
 - `sails generate controller [Name] [Action]...`
 - Ex: `sails generate controller post create find update destroy`
 - Path: `api/controller`

Sails.js Command Line

- Generate Model
 - `sails generate model [Name] [Attribute:Type]...`
 - Ex: `sails generate model person firstName:string lastName:string age:integer birthDate:date`
 - Path: `api/model`

Controller

Controller

- <http://sailsjs.org/#!documentation/controllers>
- Think of controllers as being the middleman between your model and your views.

```
sails generate controller comment create destroy tag like
```

Controller

```
module.exports = {
  create: function (req, res) {

    // Send a JSON response
    return res.json({
      hello: 'world'
    });
  },
  destroy: function (req, res) {

    // Send a JSON response
    return res.json({
      hello: 'world'
    });
  },
  tag: function (req, res) {
```


Controller

- generate routes would be the following: (Action blueprints)
 - /comment/create
 - /comment/destroy
 - /comment/tag
 - /comment/like

Controller

- REST blueprints
- You can disable it in “config/controller.js”
 - find(id) -> GET /:controller/:id
 - create() -> POST /:controller
 - update(id) -> PUT /:controller/:id
 - destroy(id) -> DELETE /:controller/:id

Request and Response

- If you need to dive deeper, check out the [express guide](#).

```
module.exports = {
  create: function (req, res) {

    // Send a JSON response
    return res.json({
      hello: 'world'
    });
  },
  destroy: function (req, res) {

    // Send a JSON response
    return res.json({
      hello: 'world'
    });
  },
  tag: function (req, res) {
```

Request and Response

- `req.param()`
 - `/:controller/:action/:foo`
 - `var foo = req.param("foo");`
- `req.body.val`
 - form data, query string

Request and Response

- `res.view([view, options[, fn]])`
- `res.send(body|status[, headers|status[, status]])`
- `res.json(obj[, headers|status[, status]])`
- `res.redirect(url[, status])`

Routes

Routes

- <http://sailsjs.org/#!/documentation/routes>
- routes urls to controllers/actions
- Path: “config/routes.js”

Routes

- static routes

```
/post: {  
  controller: 'PostController',  
  action: 'findAll'  
}
```

- controller name
- controller action (function)

Routes

- route point to a view

```
‘/’: {  
  view: ‘home/index’  
}
```

- the followings point to the “view/home/index.ejs”

Routes

- set routes for particular http verbs

```
'POST /signup': {  
  controller: controller_name,  
  action: controller_action  
}
```

Model

Model

- <http://sailsjs.org/#!documentation/models>
- Path: api/models
- ORM -> Waterline
- multiple adapters for databases

Model

```
module.exports = {  
  attributes: {  
    firstName: 'STRING',  
    lastName: 'STRING',  
    age: {  
      type: 'INTEGER',  
      max: 150,  
      required: true  
    }  
  }  
};
```

Model

attribute name

```
module.exports = {
  attributes: {
    firstName: 'STRING',
    lastName: 'STRING',
    age: {
      type: 'INTEGER',
      max: 150,
      required: true
    }
  }
};
```

Model

attribute name

attribute type

```
module.exports = {  
  attributes: {  
    firstName: 'STRING',  
    lastName: 'STRING',  
    age: {  
      type: 'INTEGER',  
      max: 150,  
      required: true  
    }  
  }  
};
```

Model

attribute name

attribute type

validation

```
module.exports = {  
  attributes: {  
    firstName: 'STRING',  
    lastName: 'STRING',  
    age: {  
      type: 'INTEGER',  
      max: 150,  
      required: true  
    }  
  }  
};
```


Model

```
Person.create({
  firstName: 'foo',
  lastName: 'bar',
  age: 18
}).done(function(err, post) {
  if(err) {
    /* ... */
  }

  /* ... */
});
```

View

View

- <http://sailsjs.org/#!documentation/views>
- render pages
- Path: views/...
- Supports:
 - ejs (default) (<http://embeddedjs.com/>)
 - jade (<http://jade-lang.com/>)
 - ...

View

- render views in controller
- `response.view()`

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


Asset Management

Asset Management

- Path: assets/...
 - css, js, images
- Grunt.js
 - less, coffeescript
 - concat, cssmin, uglify, ...

Pros & Cons

- Pros
 - Use command line to generate controllers and models
 - Developing routes / Restful API is fast
 - other good things: policy...
- Cons
 - Documentation (especially waterline)
 - No support for associations
 - Devs have not been very responsive lately

Conclusion

- Sails.js is good mvc framework for Node.js
- Building routes / Restful API is fast

?

Thank you!

E-mail: yungshin.chien@gmail.com

twitter: @yungshin_chien