

RESTing with Sails.js

Tom Marrs



About Me ...

O'REILLY



GREAT INDIAN
DEVELOPER
SUMMIT



What's The Point?

**Simplify RESTful API Development with
Sails.js**

Agenda

Sails.js Overview	1
--------------------------	----------

Getting Started with Sails	2
---------------------------------------	----------

REST APIs with Sails	3
---------------------------------	----------

Your Takeaway

Another RESTful API Tool for Node.js

We're Not Covering

Front-End Web Apps

Deep Node

Deep Express

Sails vs Hapi vs ...

Examples and Slides

<https://github.com/tmarrs/presentations/tree/master/BoulderJS/2016/RESTing-with-Sails>

Where Are We?

Sails.js Overview	1
Getting Started with Sails	2
REST APIs with Sails	3

What is Sails.js?

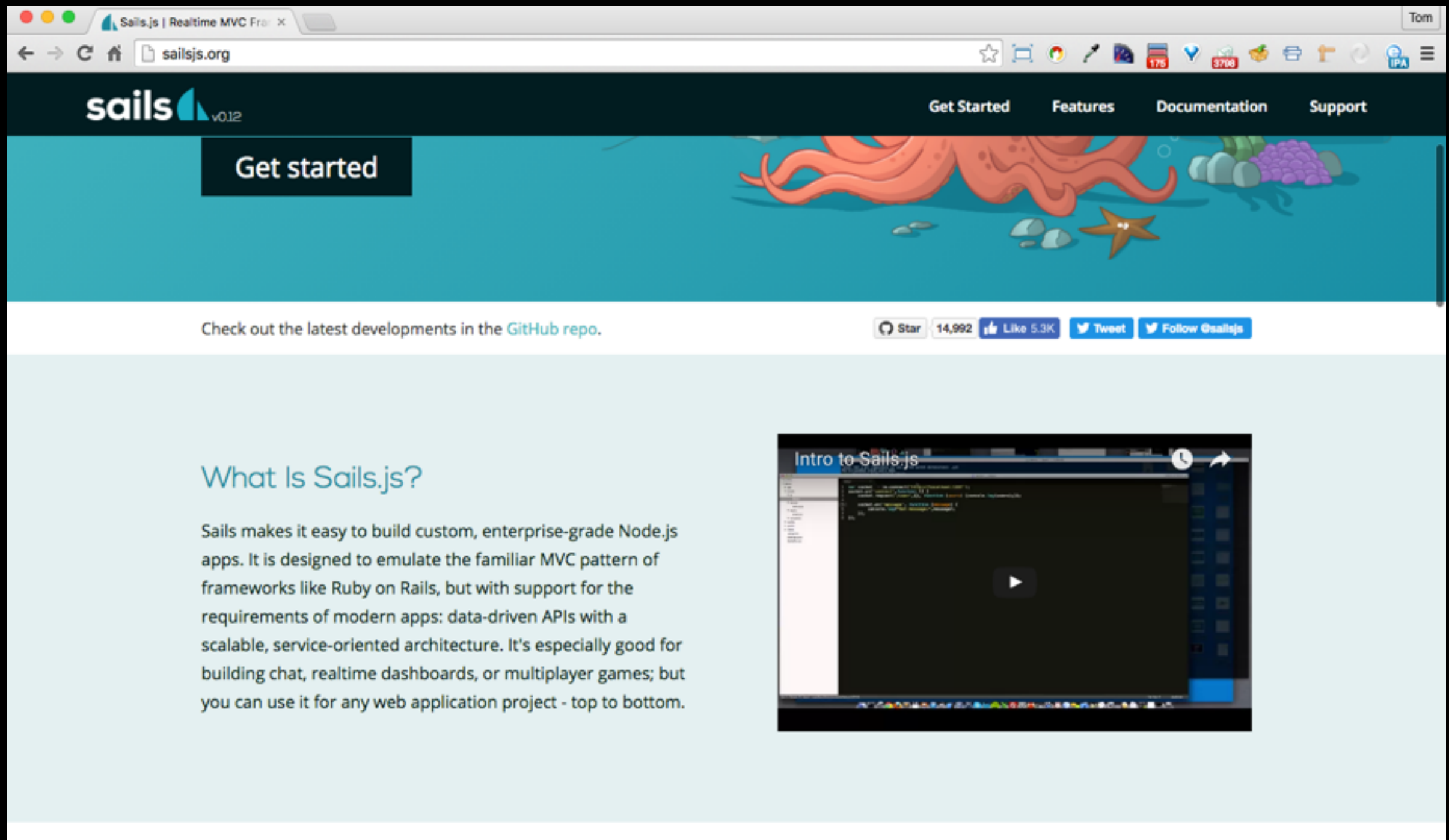
MVC Framework built on Node and Express

RESTful API Scaffolding

The Journey ...



sailsjs.org



The screenshot shows the homepage of the Sails.js website. The browser's address bar displays 'sailsjs.org'. The website's header includes the 'sails' logo with 'v0.12' and navigation links for 'Get Started', 'Features', 'Documentation', and 'Support'. A large teal banner features a cartoon octopus and a starfish, with a 'Get started' button. Below the banner, a message encourages users to check out the latest developments in the GitHub repo, accompanied by social media sharing buttons for GitHub (14,992 stars), Facebook (5.3K likes), and Twitter. The main content area is titled 'What Is Sails.js?' and describes the framework's capabilities. To the right, a video player titled 'Intro to Sails.js' is shown.

sails v0.12

Get Started Features Documentation Support

Get started

Check out the latest developments in the [GitHub repo](#).

Star 14,992 Like 5.3K Tweet Follow @sailsjs

What Is Sails.js?

Sails makes it easy to build custom, enterprise-grade Node.js apps. It is designed to emulate the familiar MVC pattern of frameworks like Ruby on Rails, but with support for the requirements of modern apps: data-driven APIs with a scalable, service-oriented architecture. It's especially good for building chat, realtime dashboards, or multiplayer games; but you can use it for any web application project - top to bottom.

Intro to Sails.js

Sails on GitHub

The screenshot shows the GitHub repository for Sails.js, maintained by balderdashy. The repository is titled "balderdashy / sails" and has 824 watches, 15,004 stars, and 1,560 forks. It is a Realtime MVC Framework for Node.js, with a link to the official website (http://sailsjs.org). The repository statistics show 5,738 commits, 30 branches, 184 releases, and 209 contributors. The latest commit is by mikermcneil, titled "Fixes tests by switching back to using == in RPS tests for consistenc...", committed 5 days ago. The file list includes .github, bin, errors, lib, test, .editorconfig, .gitignore, .jshintrc, .npmignore, .npmrc, .travis.yml, and 0.11-migration-guide.md.

balderdashy/sails: Realtime MVC Framework for Node.js <http://sailsjs.org>

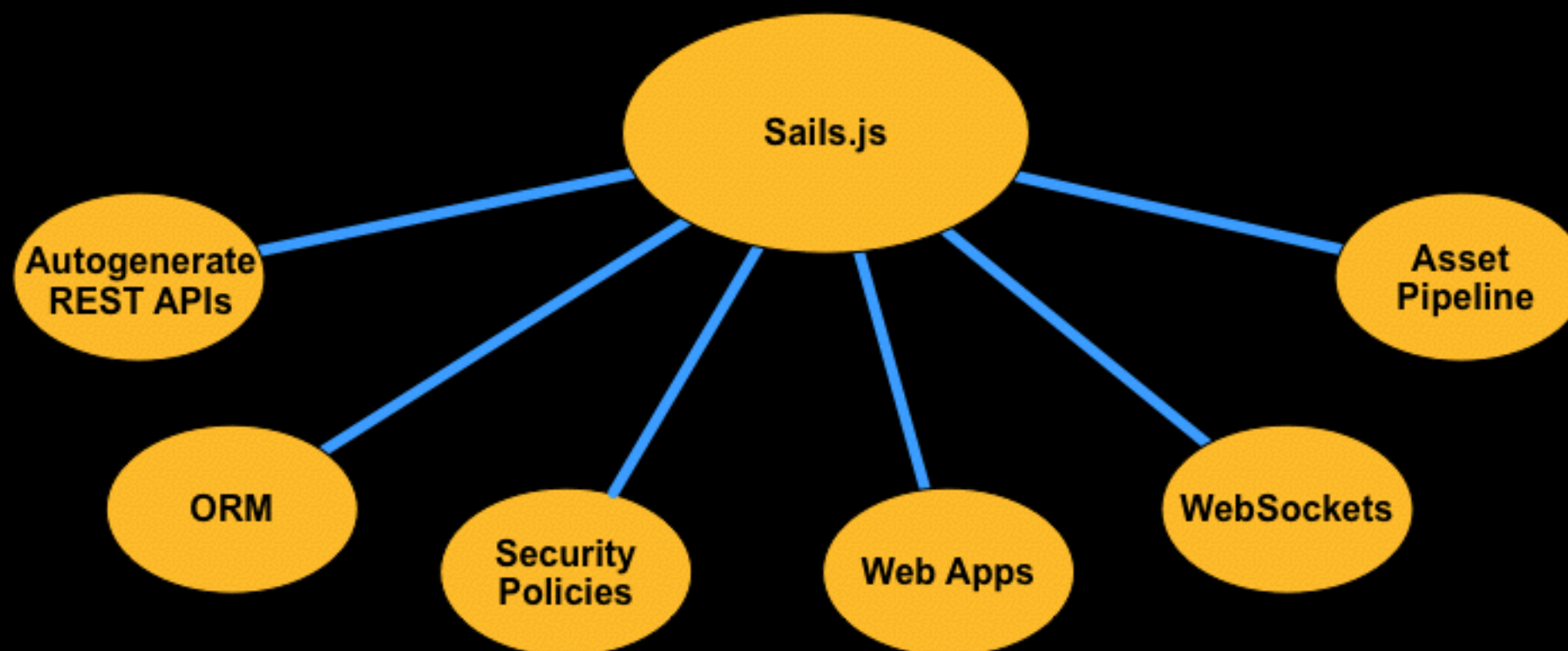
5,738 commits 30 branches 184 releases 209 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

mikermcneil Fixes tests by switching back to using == in RPS tests for consistenc... Latest commit 2fb022a 5 days ago

.github	Update ISSUE_TEMPLATE	4 months ago
bin	Switch to chalk	4 months ago
errors	Switch to chalk	4 months ago
lib	Use `.read()` to get response body instead of accessing internal buff...	12 days ago
test	Fixes tests by switching back to using == in RPS tests for consistenc...	5 days ago
.editorconfig	Improve default serverError handler in builtin router. Also, I'll pro...	2 years ago
.gitignore	Added globby thing for logfiles related to perf timing tests to gitig...	3 months ago
.jshintrc	Standardize jshintrc for consistency w/ our deps including the new wl...	5 months ago
.npmignore	added all md files to the npmignore (README is an automatic exception...	3 years ago
.npmrc	Checking npm-Version on npm-install	2 years ago
.travis.yml	run test suite on node v6	a month ago
0.11-migration-guide.md	Update 0.11-migration-guide.md	6 months ago

What Does Sails.js Provide?



Where Are We?

Sails.js Overview	1
Get Started with Sails	2
REST APIs with Sails	3

Install Sails

```
npm -g install sails
```

```
json-at-work => sails -v  
0.12.3  
_
```

Your First Sails Project

```
sails generate new FirstSailsProject
```


Starting Sails ...

```
1. node
json-at-work => sails lift

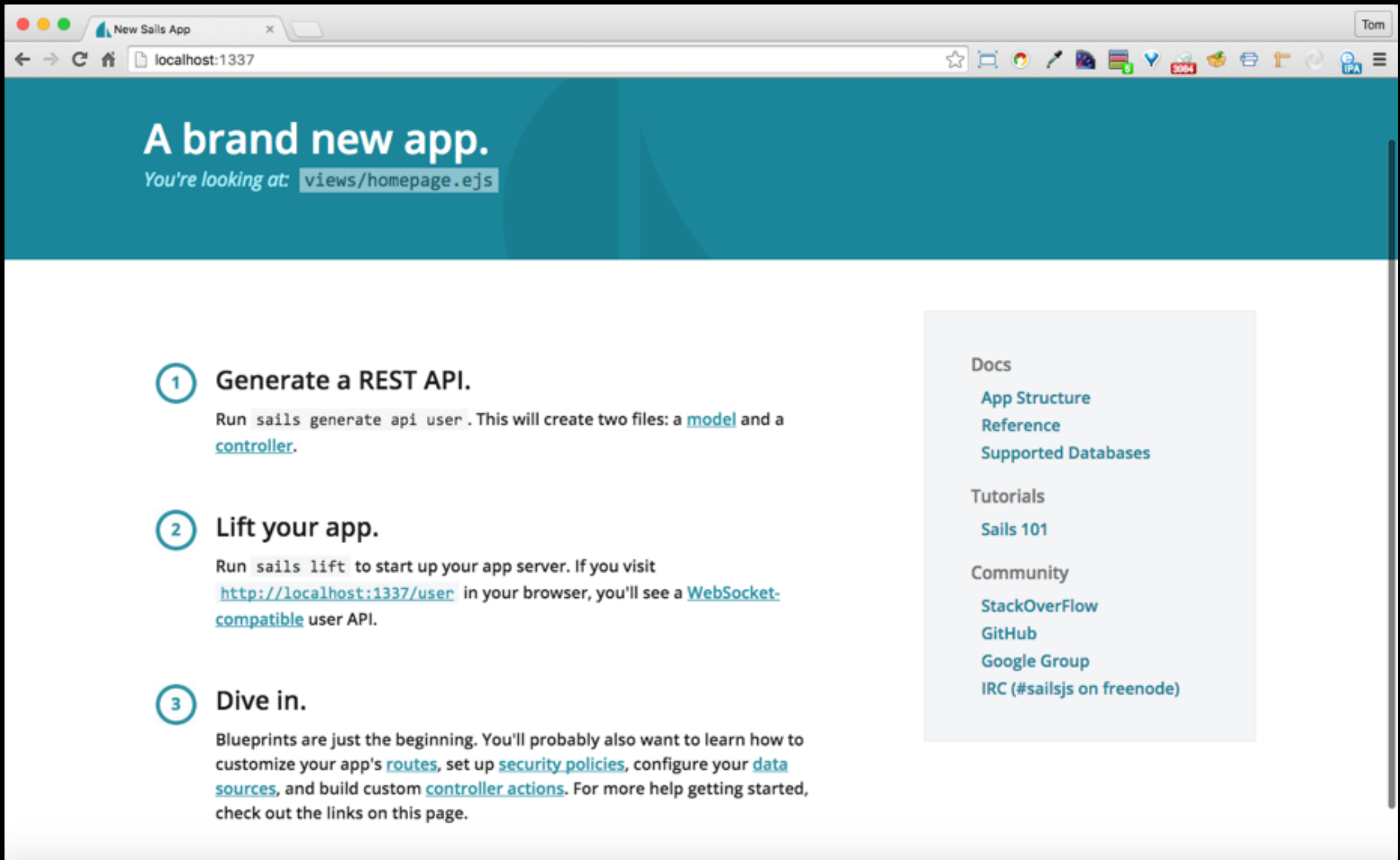
info: Starting app...

info:
info:      .-...-.
info:
info:  Sails      <|      .-...-.
info:  v0.12.3
info:
info:      / \
info:     /  \
info:    /    \
info:   /      \
info:  /        \
info: /          \
info:/          \
info:  \          /
info:   \        /
info:    \      /
info:     \    /
info:      \  /
info:       \/
info:      ---
info:     ---
info:
info: Server lifted in `/Users/tmarrs/projects/resting-with-sails/FirstSailsProject`
info: To see your app, visit http://localhost:1337
info: To shut down Sails, press <CTRL> + C at any time.

debug: -----
debug: :: Wed Jul 20 2016 15:40:19 GMT-0600 (MDT)

debug: Environment : development
debug: Port         : 1337
debug: -----
```

Is Sails Working?



New Sails App

localhost:1337

A brand new app.

You're looking at: `views/homepage.ejs`

- 1 Generate a REST API.**

Run `sails generate api user`. This will create two files: a [model](#) and a [controller](#).
- 2 Lift your app.**

Run `sails lift` to start up your app server. If you visit <http://localhost:1337/user> in your browser, you'll see a [WebSocket-compatible](#) user API.
- 3 Dive in.**

Blueprints are just the beginning. You'll probably also want to learn how to customize your app's [routes](#), set up [security policies](#), configure your [data sources](#), and build custom [controller actions](#). For more help getting started, check out the links on this page.

Docs

- [App Structure](#)
- [Reference](#)
- [Supported Databases](#)

Tutorials

- [Sails 101](#)

Community

- [StackOverFlow](#)
- [GitHub](#)
- [Google Group](#)
- [IRC \(#sailsjs on freenode\)](#)

Stop Sails

Ctrl-C

Are We Done Yet? No ...

```
json-at-work => ls -alFt
total 56
drwxr-xr-x  16 tmarrs  staff  544 Jul 20 15:40 ./
drwxr-xr-x   3 tmarrs  staff  102 Jul 20 15:40 .tmp/
drwxr-xr-x 219 tmarrs  staff 7446 Jul 20 15:38 node_modules/
drwxr-xr-x   8 tmarrs  staff  272 Jul 20 15:37 assets/
drwxr-xr-x   6 tmarrs  staff  204 Jul 20 15:37 tasks/
drwxr-xr-x   4 tmarrs  staff  136 Jul 20 15:37 ../
-rw-r--r--   1 tmarrs  staff  166 Jul 20 15:37 .editorconfig
-rw-r--r--   1 tmarrs  staff 2758 Jul 20 15:37 .gitignore
-rw-r--r--   1 tmarrs  staff   43 Jul 20 15:37 .sailsrc
-rw-r--r--   1 tmarrs  staff 2265 Jul 20 15:37 Gruntfile.js
-rw-r--r--   1 tmarrs  staff   63 Jul 20 15:37 README.md
drwxr-xr-x   7 tmarrs  staff  238 Jul 20 15:37 api/
-rw-r--r--   1 tmarrs  staff 1802 Jul 20 15:37 app.js
drwxr-xr-x  20 tmarrs  staff  680 Jul 20 15:37 config/
-rw-r--r--   1 tmarrs  staff  929 Jul 20 15:37 package.json
drwxr-xr-x   7 tmarrs  staff  238 Jul 20 15:37 views/
```

Where Are We?

Sails.js Overview **1**

**Get Started with
Sails** **2**

**REST APIs with
Sails** **3**

Sails API Project

```
sails generate new MyFirstApi \  
  --no-front-end
```

Sails API Directory Structure

```
json-at-work => ls -alFt
total 48
drwxr-xr-x 33 tmarrs staff 1122 Jul 20 16:09 node_modules/
drwxr-xr-x 11 tmarrs staff  374 Jul 20 16:09 ./
-rw-r--r--  1 tmarrs staff  166 Jul 20 16:09 .editorconfig
-rw-r--r--  1 tmarrs staff 2758 Jul 20 16:09 .gitignore
-rw-r--r--  1 tmarrs staff   80 Jul 20 16:09 .sailsrc
-rw-r--r--  1 tmarrs staff   56 Jul 20 16:09 README.md
drwxr-xr-x  7 tmarrs staff  238 Jul 20 16:09 api/
-rw-r--r--  1 tmarrs staff 1802 Jul 20 16:09 app.js
drwxr-xr-x 20 tmarrs staff  680 Jul 20 16:09 config/
-rw-r--r--  1 tmarrs staff  484 Jul 20 16:09 package.json
drwxr-xr-x  5 tmarrs staff  170 Jul 20 16:09 ../
```

Starting Sails - Just a Bit Whiny

```
json-at-work => sails lift
```

```
info: Starting app...
```

```
info: Could not find module: ejs in path: /Users/tmarrs/projects/resting-with-sails/MyFirstApi/node_modules
error: Ignoring attempt to bind route (`/`) to unknown view: `homepage`
```

```
info:
```

```
info:
```

```
.....
```

```
info:
```

```
info: Sails
```

```
info: v0.12.3
```

```
info:
```

```
info:
```

```
info:
```

```
info:
```

```
info:
```

```
info:
```

```
info:
```

```
info:
```

```
info: Server lifted in `/Users/tmarrs/projects/resting-with-sails/MyFirstApi`
```

```
info: To see your app, visit http://localhost:1337
```

```
info: To shut down Sails, press <CTRL> + C at any time.
```

```
debug: -----
```

```
debug: :: Wed Jul 20 2016 16:15:13 GMT-0600 (MDT)
```

```
debug: Environment : development
```

```
debug: Port : 1337
```

```
debug: -----
```

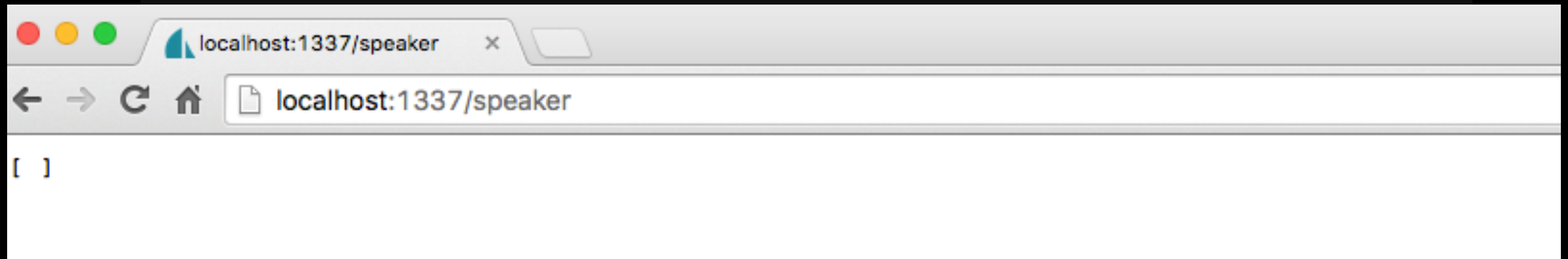

Generate an API - Finally

```
json-at-work => sails generate api speaker  
info: Created a new api!
```

api/models/Speaker.js - Model

```
1  /**
2  * Speaker.js
3  *
4  * @description :: TODO: You might write a short summary of how this model
5  * @docs         :: http://sailsjs.org/documentation/concepts/models-and-
6  */
7
8  module.exports = {
9
10   attributes: {
11     fullName: {
12       type: 'string',
13       required: true
14     },
15     age: {
16       type: 'integer',
17       required: true
18     },
19     registered: {
20       type: 'boolean',
21       required: true
22     }
23   }
24 ;
```

Empty Speaker API



Waterline ORM

The screenshot shows a web browser window displaying the Sails.js documentation page for Waterline ORM. The browser's address bar shows the URL `sailsjs.org/documentation/concepts/models-and-orm`. The page has a dark teal header with the Sails.js logo and navigation links: "Get Started", "Features", "Documentation" (which is underlined), and "Support". Below the header is a teal sidebar with the word "Concepts" and three sub-links: "Anatomy", "Reference", and "Concepts" (which is highlighted with a dark teal background). The main content area has a white background. On the left side of the main area is a vertical list of navigation links: "Concepts", "Assets", "Blueprints", "Configuration", "Controllers", "Custom Responses", "Deployment", "Extending Sails", "File Uploads", "Globals", "Internationalization", "Logging", "Middleware", "Models and ORM" (which is highlighted with a blue background and a right-pointing arrow), "Associations", "Attributes", "Models", "Lifecycle callbacks", and "Validations". The main content area features a large heading "Waterline: SQL/noSQL Data Mapper (ORM/ODM)" followed by a paragraph explaining that Sails.js includes Waterline, a datastore-agnostic tool that simplifies database interaction. Below this is a section titled "Database Agnosticism" with a paragraph explaining how Waterline abstracts database-specific details, allowing for easy switching between databases like PostgreSQL, Oracle, MySQL, MongoDB, and Redis. The page ends with a partial paragraph about Waterline's query syntax.

Models and ORM | Sails.js

sailsjs.org/documentation/concepts/models-and-orm

sails v0.12

Get Started Features Documentation Support

Concepts Anatomy Reference **Concepts**

Concepts

- Assets
- Blueprints
- Configuration
- Controllers
- Custom Responses
- Deployment
- Extending Sails
- File Uploads
- Globals
- Internationalization
- Logging
- Middleware
- Models and ORM**
- Associations
- Attributes
- Models
- Lifecycle callbacks
- Validations

Waterline: SQL/noSQL Data Mapper (ORM/ODM)

Sails comes installed with a powerful [ORM/ODM](#) called [Waterline](#), a datastore-agnostic tool that dramatically simplifies interaction with one or more [databases](#). It provides an abstraction layer on top of the underlying database, allowing you to easily query and manipulate your data *without* writing vendor-specific integration code.

Database Agnosticism

In schemaful databases like [Postgres](#), [Oracle](#), and [MySQL](#), models are represented by tables. In [MongoDB](#), they're represented by Mongo "collections". In [Redis](#), they're represented using key/value pairs. Each database has its own distinct query dialect, and in some cases even requires installing and compiling a specific native module to connect to the server. This involves a fair amount of overhead, and garners an unsettling level of [vendor lock-in](#) to a specific database; e.g. if your app uses a bunch of SQL queries, it will be very hard to switch to Mongo later, or Redis, and vice versa.

Waterline query syntax floats above all that, focusing on business logic like creating new records, fetching/searching existing records, updating records, or destroying records. No matter what database you're contacting, the usage is *exactly the same*. Furthermore, Waterline allows you to `.populate()` associations between models, *even if* the data for each model lives in a different database. That means you can switch your app's models from Mongo, to Postgres, to MySQL, to Redis, and back again -

Waterline ORM - Validations

Validations | Sails.js Docum X

sailsjs.org/documentation/concepts/models-and-orm/validations

sails v0.12

Get StartedFeaturesDocumentationSupport

Query Language

Model Settings

Policies

Programmatic Usage

Realtime

Routes


Security

Services

Testing

Upgrading

Views



We wrote a book!

Get early access to the book with promotion code: **mcneildoc**

Get the Book

Built-in Data Types

Every attribute definition must have a built-in data type (or *typeclass*) specified. This is used for logical validation and coercion of results and criteria.

Data Type	Usage	Description
String	type: 'string'	Any string (tolerates null).
Number	type: 'number'	Any number (tolerates null)
Boolean	type: 'boolean'	true or false (also tolerates null)
JSON	type: 'json'	Any JSON-serializable value, including numbers, booleans, strings, arrays, dictionaries, and null.
Array	type: 'array'	Any array consisting solely of JSON-serializable contents.

Different databases vary slightly in the way they handle edge cases and special values such as `Infinity`, `null`, strings of varying lengths, etc. Sails' ORM (Waterline) and its adapters perform loose validation to ensure that the values provided in criteria dictionaries and as values to `.create()` or `.update()` match the expected typeclass.

Speaker API - POST a Record

The screenshot displays a REST client interface with the following details:

- URL:** `http://localhost:1337/speak`
- Method:** `POST`
- Environment:** No environment
- Body Tab:** Selected, showing a JSON payload:

```
{  "fullName": "Lars Larsen",  "age": 29,  "registered": true}
```
- Request Headers:** `Content-Type: application/json`
- Response Status:** `201 Created`
- Response Time:** `61 ms`
- Response Body (Pretty):**

```
{  "fullName": "Lars Larsen",  "age": 29,  "registered": true,  "createdAt": "2016-07-20T22:45:47.495Z",  "updatedAt": "2016-07-20T22:45:47.495Z",  "id": 1}
```

Speaker API - GET all

The screenshot shows a web browser's developer tools interface. At the top, the method is set to 'GET' and the URL is 'http://localhost:1337/speaker'. The 'Send' button is highlighted in blue. Below the URL bar, the 'Body' tab is selected, showing the response data. The status is '200 OK' and the time taken is '20 ms'. The response body is displayed in 'JSON' format, showing a single object with the following properties:

```
[
  {
    "fullName": "Lars Larsen",
    "age": 29,
    "registered": true,
    "createdAt": "2016-07-23T22:20:23.435Z",
    "updatedAt": "2016-07-23T22:20:23.435Z",
    "id": 3
  }
]
```

Speaker API - DELETE

The screenshot shows a REST client interface with the following components:

- Method and URL:** DELETE `http://localhost:1337/speaker/1`
- Buttons:** Params, Send, Save
- Tabs:** Authorization, Headers, Body, Pre-request Script, Tests
- Authorization:** Type: No Auth
- Response Section:**
 - Tabs:** Body, Cookies, Headers (11), Tests
 - Status:** 200 OK
 - Time:** 89 ms
 - Format:** JSON
 - Content:**

```
1 {
2   "fullName": "Lars Larsen",
3   "age": 29,
4   "registered": true,
5   "createdAt": "2016-07-20T22:45:47.495Z",
6   "updatedAt": "2016-07-20T22:45:47.495Z",
7   "id": 1
8 }
```


Add Associations - Tag

```
1  /**
2   * Tag.js
3   *
4   * @description :: TODO: You might write a short summary of how th
5   * @docs        :: http://sailsjs.org/documentation/concepts/model
6   */
7
8  module.exports = {
9
10   attributes: {
11     name: {
12       type: 'string',
13       required: true
14     },
15     owner: {
16       model: 'speaker'
17     }
18   }
19 };
20
```

Add Associations - Speaker

```
2  * Speaker.js
3  *
4  * @description :: TODO: You might write a short summary of this module
5  * @docs        :: http://sailsjs.org/documentation/concepts/associations
6  */
7
8  module.exports = {
9
10   attributes: {
11     fullName: {
12       type: 'string',
13       required: true
14     },
15     age: {
16       type: 'integer',
17       required: true
18     },
19     registered: {
20       type: 'boolean',
21       required: true
22     },
23     tags: {
24       collection: 'tag',
25       via: 'owner'
26     }
27   }
28 }
```

Speaker API - Get Speaker/Tags

The screenshot shows a web browser interface for the Speaker API. The top bar displays the method **GET** and the URL `http://localhost:1337/speaker`. To the right are buttons for **Params**, **Send**, and **Save**. Below the top bar, the **Body** tab is selected, showing the response status **200 OK** and time **35 ms**. The response is displayed in **JSON** format, showing a list of tags and speaker information.

```
{
  "tags": [
    {
      "name": "REST",
      "createdAt": "2016-07-23T22:25:59.706Z",
      "updatedAt": "2016-07-23T22:28:39.179Z",
      "id": 5,
      "owner": 4
    },
    {
      "name": "JS",
      "createdAt": "2016-07-23T22:26:05.360Z",
      "updatedAt": "2016-07-23T22:28:39.181Z",
      "id": 6,
      "owner": 4
    },
    {
      "name": "Node",
      "createdAt": "2016-07-23T22:26:12.703Z",
      "updatedAt": "2016-07-23T22:28:39.182Z",
      "id": 7,
      "owner": 4
    }
  ],
  "fullName": "Fred Green",
  "age": 29,
  "registered": true,
  "createdAt": "2016-07-23T22:28:39.162Z",
  "updatedAt": "2016-07-23T22:28:39.169Z",
  "id": 4
}
```

Sails and APIs - What Else?

Custom Controllers

Security Policies

Access a Real DB

Agenda

Sails.js Overview	1
--------------------------	----------

Getting Started with Sails	2
---------------------------------------	----------

REST APIs with Sails	3
---------------------------------	----------

Going Deeper - Sails 101

The screenshot shows the GitHub organization page for 'Sails 101'. The browser's address bar displays 'https://github.com/sails101'. The organization's profile includes a logo of two overlapping teal triangles, the name 'Sails 101', and a description: 'Simple example apps, each demonstrating a specific concept.' Below this, there are links to 'http://sailsjs.org' and 'hello@balderdash.io'. The 'Repositories' tab is selected, showing a list of three repositories. The first repository is 'use-xml-body-parser', a JavaScript project with 0 stars and 3 forks, described as 'How do I use a custom body parser config to handle XML requests?' and updated on April 12. The second is 'file-uploads', also in JavaScript, with 36 stars and 12 forks, described as 'How do I upload files in Sails?' and updated on March 14. The third is 'jwt-login', in JavaScript, with 8 stars and 5 forks, described as 'How do I implement login with JWT (JSON Web Tokens)?' and updated on February 26. On the right side, the 'People' section lists two members: 'yedhukrishnan' (Yedhu Krishnan) and 'Zolmeister' (Zoli Kahan).

Sails 101

Simple example apps, each demonstrating a specific concept.

<http://sailsjs.org> hello@balderdash.io

Repositories

Filters Find a repository...

use-xml-body-parser JavaScript ★ 0 🍴 3

How do I use a custom body parser config to handle XML requests?

Updated on Apr 12

file-uploads JavaScript ★ 36 🍴 12

How do I upload files in Sails?

Updated on Mar 14

jwt-login JavaScript ★ 8 🍴 5

How do I implement login with JWT (JSON Web Tokens)?

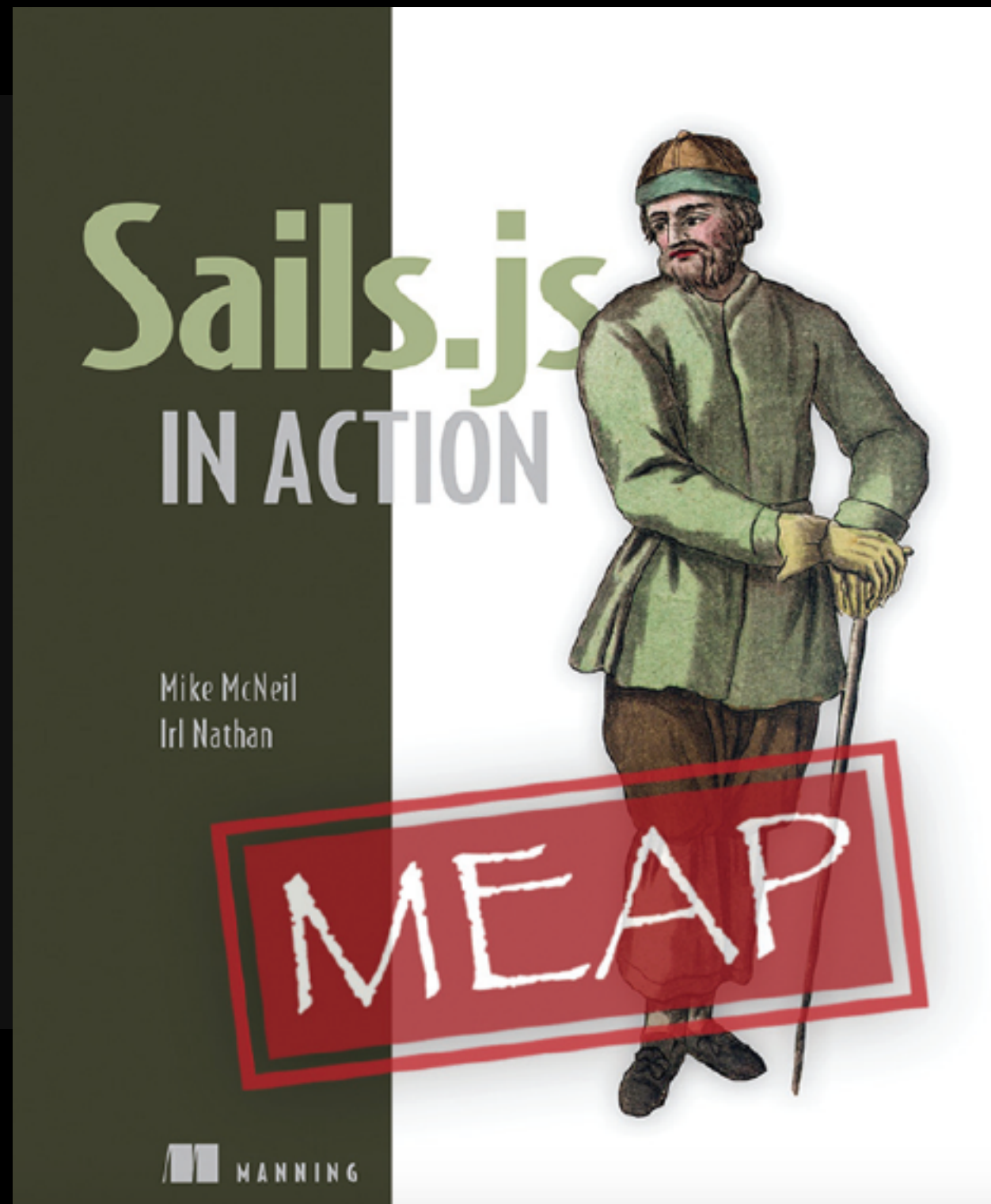
Updated on Feb 26

People 2 >

yedhukrishnan
Yedhu Krishnan

Zolmeister
Zoli Kahan

Going Deeper - Sails.js in Action



What's The Point?

**Simplify RESTful API Development with
Sails.js**

Questions?

Tom Marrs

thomasamarrs@comcast.net



sails 

Sails Resources

Sails.js Site - <http://sailsjs.org>

Sails.js GitHub Repo - <https://github.com/balderdashy/sails>

Rest API Development with Sails.js - <http://www.jlmonteagudo.com/2014/06/rest-api-development-with-sails-js/>

Sails 101 - <https://devdactic.com/rapid-development-with-sailsjs/>

Rapid REST API Development with Sails.js - <https://github.com/sails101>

Sails.js in Action - <https://www.manning.com/books/sails-js-in-action>

Handlebars Resources

Handlebars Site - <http://handlebarsjs.com/>

JS Online Template Tester - <http://rowno.github.io/architect/>