

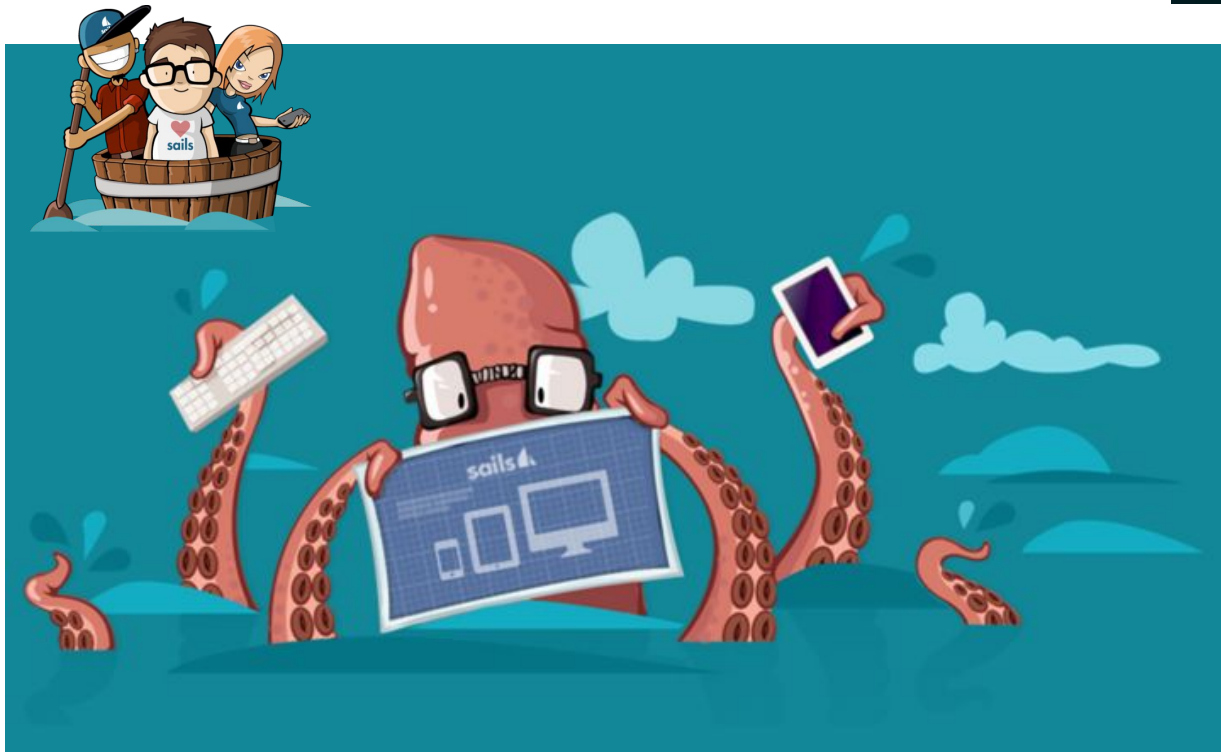
快快乐樂上手 Sails.js

Realtime MVC Framework for NodeJS

marty.shu@gmail.com

Realtime MVC Framework

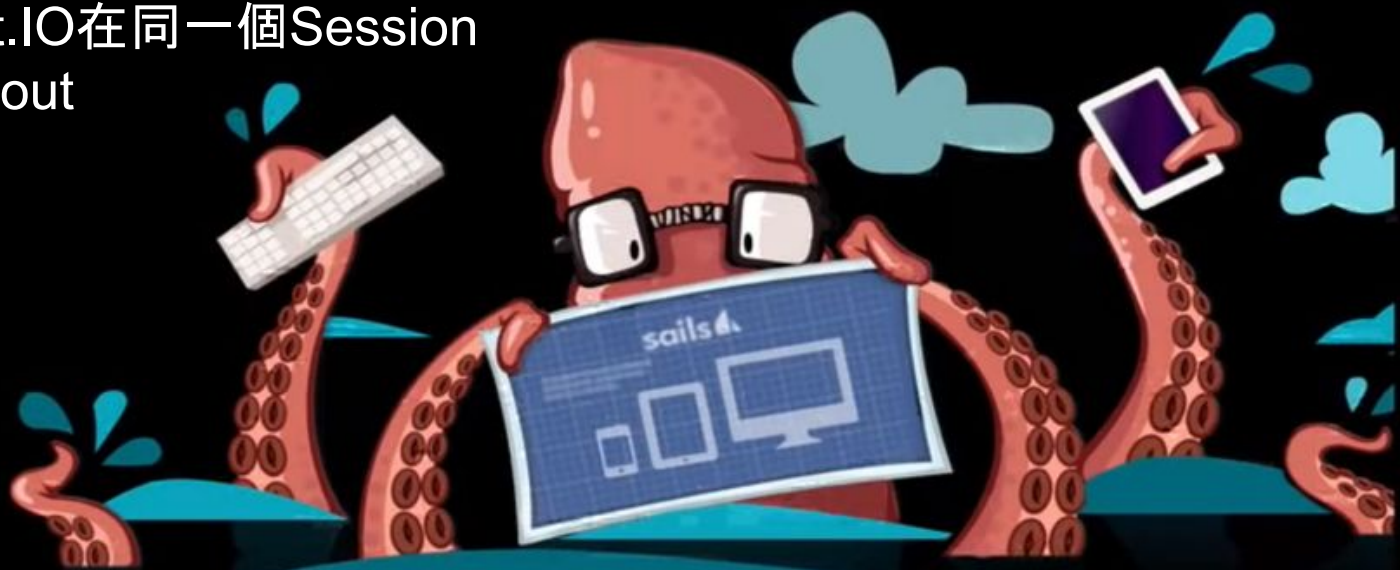
sails 



- 用JavaScript 開發後端
- 抄RoR MVC Framework
- 提供 ORM
- 加上WebSocket
- More.. [Link](#)

FULL STACK NODE.JS

1. 提供 ORM (Waterline)
2. 用 RESTful 方式調用 API
3. 用 SocketIO, 先天就 realtime
4. 預設就提供角色授權、參數設定
5. Express+Socket.IO在同一個Session
6. 幫忙處理 UI Layout



Get started

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
```

Create model

1

Get your API going.

Run `sails generate foo`. This will create a model `Foo` and controller `FooController`

2

Lift your app.

Run `sails lift` to start up your app. If you visit `http://localhost:1337/foo` in your browser, you'll see a socket.io-compatible REST API was generated for your 'Foo' model.

3

Dive in and start building.

From here, you can modify your models, create custom controller methods as Express middleware, and create custom routes (routes are set up in `config/routes.js`). Visit [the Sails website](#) for more information on next steps.

Next Steps

- ❑ Build a single page app
- ↔ Build a traditional webapp
- 💬 Build a realtime app
- Build an API

routes.js

Guides

New to Node.js?

What is Sails?

Models

Controllers

Assets

Asset Management

Routes

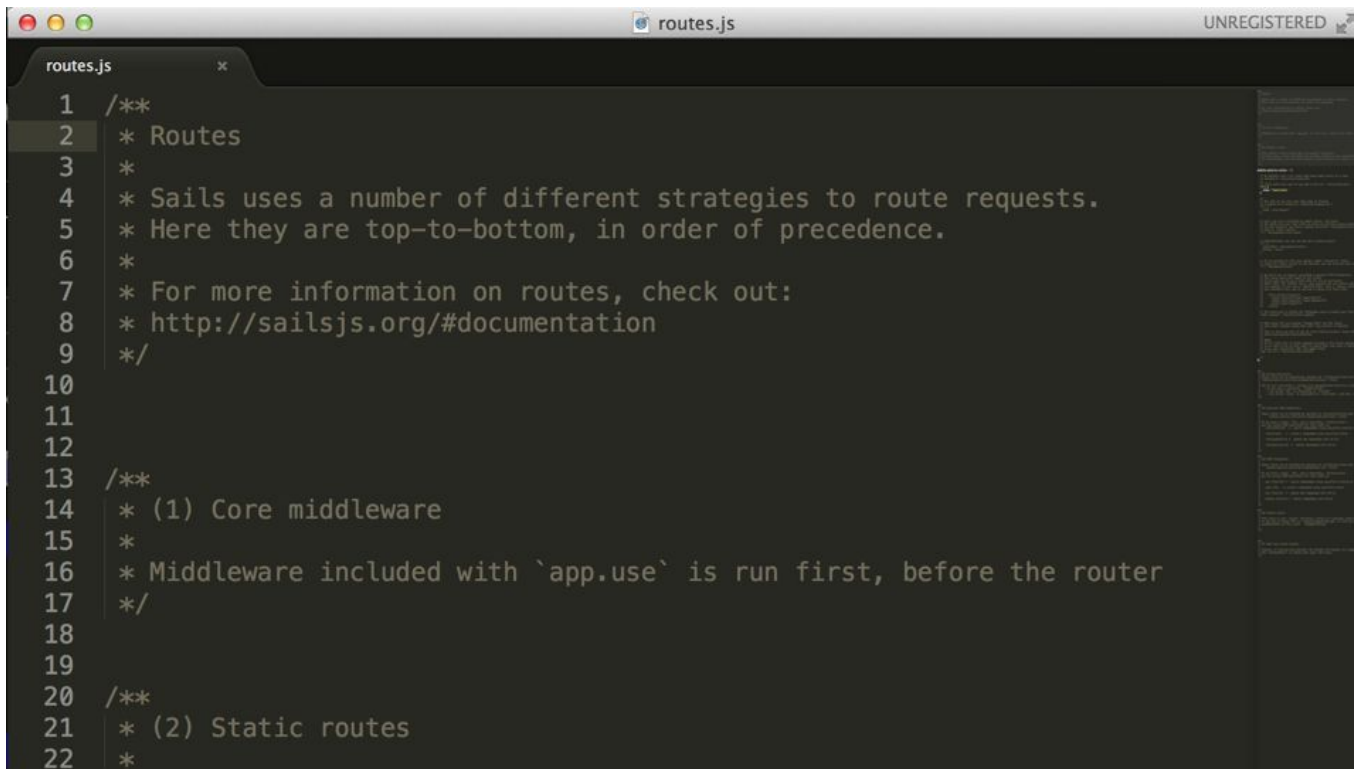
Policies

Views

Custom Adapters

Deployment

Migration Guide



```
1 /**
2  * Routes
3  *
4  * Sails uses a number of different strategies to route requests.
5  * Here they are top-to-bottom, in order of precedence.
6  *
7  * For more information on routes, check out:
8  * http://sailsjs.org/#documentation
9  */
10
11
12
13 /**
14  * (1) Core middleware
15  *
16  * Middleware included with `app.use` is run first, before the router
17  */
18
19
20 /**
21  * (2) Static routes
22  *
```

ORM - Waterline

An adapter based ORM for use with Node.js and the Sails.js Framework

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

Waterline

build **failing** npm module **0.9.11** dependencies **none**

Waterline is a brand new kind of storage and retrieval engine.

It provides a uniform API for accessing stuff from different kinds of databases, protocols, and 3rd party APIs.

That means you write the same code to get users, whether they live in MySQL, LDAP, MongoDB, or Facebook.

At the same time, Waterline aims to learn lessons and maintain the best features from both Rails' ActiveRecord and Grails' Hibernate ORMs.

ORM - Waterline

Database Support

Note: You are viewing the Sails.js v0.9.0 documentation. If you're looking for information on v0.8.x, please visit [here](#).

Sails uses **Waterline** as its ORM, and supports all databases which have waterline adapters:

✓ PostgreSQL - 0.9+ compatible - `sails-postgresql`

✓ MySQL - 0.9+ compatible - `sails-mysql`

✓ MongoDB - 0.9+ compatible - `sails-mongo`

✓ Memory - 0.9+ compatible - `sails-memory`

✓ Disk - 0.9+ compatible - `sails-disk`

CRUD Model

create

```
localhost:1337/foo/create?name=marty
```

```
{
  "name": "marty",
  "createdAt": "2014-02-12T06:53:28.041Z",
  "updatedAt": "2014-02-12T06:53:28.041Z",
  "id": 1
}
```

destroy

```
localhost:1337/foo/destroy/13
```

```
{
  "name": "xxxx",
  "createdAt": "2014-02-14T13:12:36.641Z",
  "updatedAt": "2014-02-14T13:12:36.641Z",
  "id": 13
}
```

update

```
localhost:1337/foo/update/2?email=marty.shu@gmail.com
```

```
{
  "name": "marty1",
  "createdAt": "2014-02-12T06:54:01.453Z",
  "updatedAt": "2014-02-12T06:57:31.639Z",
  "id": 2,
  "email": "marty.shu@gmail.com"
}
```

Query Model

localhost:1337/foo

```
[
  {
    "name": "marty",
    "createdAt": "2014-02-12T06:53:28.041Z",
    "updatedAt": "2014-02-12T06:53:28.041Z",
    "id": 1
  },
  {
    "name": "marty1",
    "createdAt": "2014-02-12T06:54:01.453Z",
    "updatedAt": "2014-02-12T06:54:01.453Z",
    "id": 2
  },
  {
    "name": "marty2",
    "createdAt": "2014-02-12T06:54:04.596Z",
    "updatedAt": "2014-02-12T06:54:04.596Z",
    "id": 3
  }
]
```

localhost:1337/foo/find?name=ZXC

```
[
  {
    "name": "ZXC",
    "createdAt": "2014-02-14T14:34:17.597Z",
    "updatedAt": "2014-02-14T14:34:17.597Z",
    "id": 32
  }
]
```

localhost:1337/foo?limit=1

```
[
  {
    "name": "batman",
    "createdAt": "2014-02-12T06:53:28.041Z",
    "updatedAt": "2014-02-12T06:56:25.869Z",
    "id": 1
  }
]
```

Query Modifiers

In order to use a `contains` modifier, you would do the following.

```
where: {  
  name: {  
    contains: 'James'  
  }  
}
```

Modifier: or

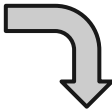
In order to use an `or` modifier, you would do the following.

```
where: {  
  or: [{name: 'James'}, {name: 'Mike'}]  
}
```

```
name: { '>': 'a' }
```

Query -> Create -> Query

```
localhost:1337/foo  
[]
```



```
localhost:1337/foo/create?name=marty  
  
{  
  "name": "marty",  
  "createdAt": "2014-02-14T16:26:28.942Z",  
  "updatedAt": "2014-02-14T16:26:28.942Z",  
  "id": 106  
}
```



```
localhost:1337/foo  
[  
  {  
    "name": "marty",  
    "createdAt": "2014-02-14T16:26:28.942Z",  
    "updatedAt": "2014-02-14T16:26:28.942Z",  
    "id": 106  
  }  
]
```

Socket IO

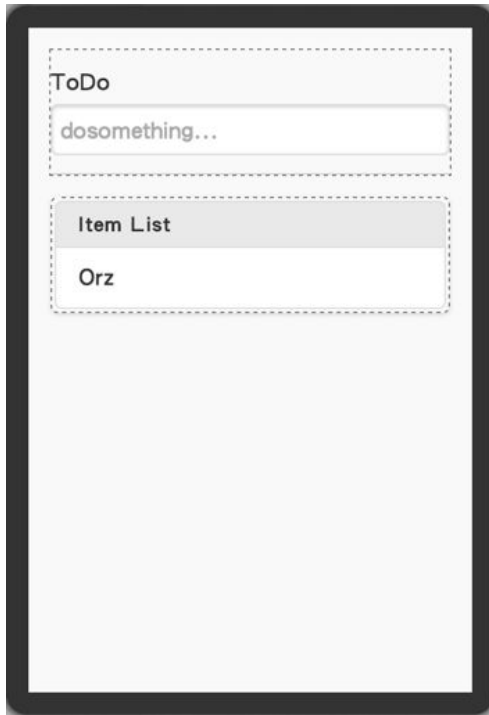
```
(function (io) {  
  // as soon as this file is loaded, connect automatically,  
  var socket = io.connect();  
  if (typeof console !== 'undefined') {  
    log('Connecting to Sails.js...');  
  }  
  
  socket.on('connect', function socketConnected() {  
    // Listen for Comet messages from Sails  
    socket.on('message', function messageReceived(message) {  
      ///////////////////////////////////////////////////////////////////  
      // Replace the following with your own custom logic  
      // to run when a new message arrives from the Sails.js  
      // server.  
      ///////////////////////////////////////////////////////////////////  
      log('New comet message received :: ', message);  
      ///////////////////////////////////////////////////////////////////  
    });  
  });  
});
```

connect...
on('connect')
on('message')

Socket IO

```
var socket = io.connect('http://localhost:1337');
socket.on('connect',function(){
  socket.request('/foo',{},function(d){console.log(d)});
  socket.on('message',function(data){
    console.log(data);
  });
});
```

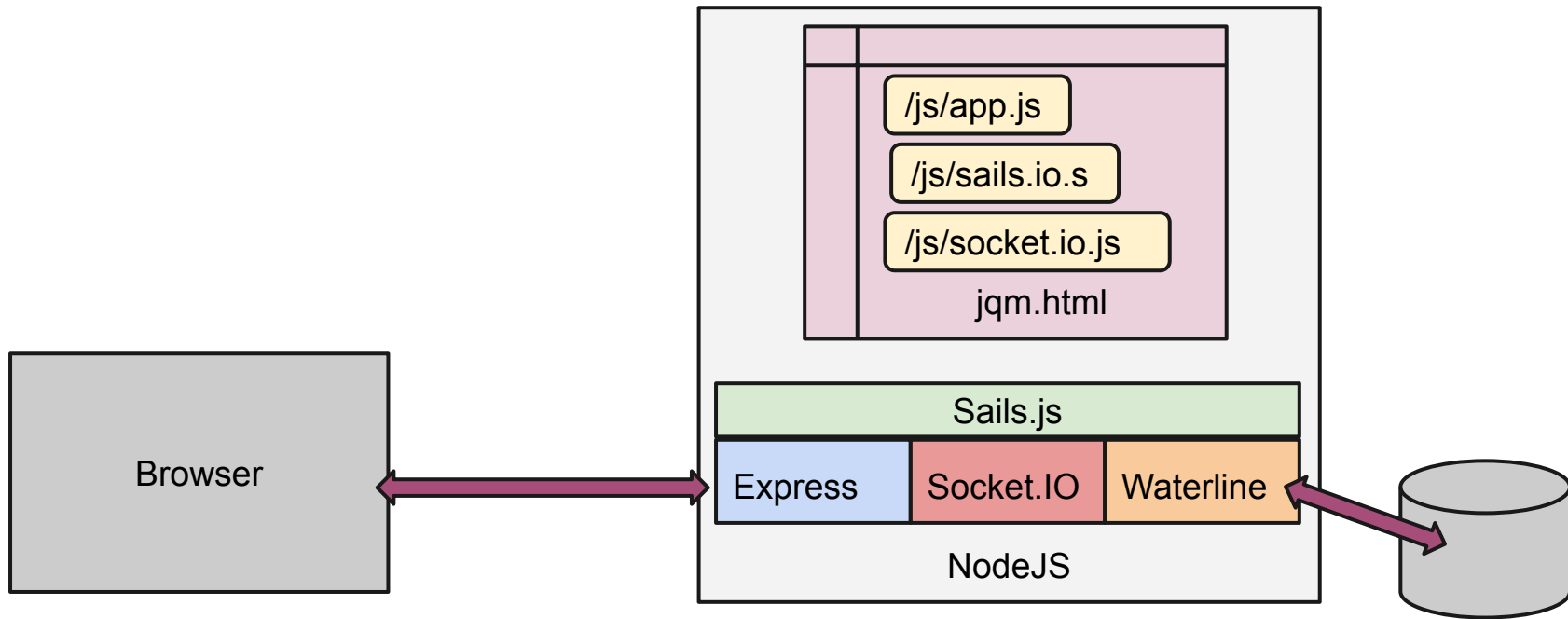
LiveDemo: ToDoList



The screenshot shows a mobile application interface with a dark border. At the top, there is a dashed-line box containing the text 'ToDo' above a text input field with the placeholder text 'dosomething...'. Below this is another dashed-line box containing the text 'Item List' above a list view. The list view contains a single item, 'Orz', which is highlighted with a light gray background.

```
1 <div id="home" data-role="page">
2   <div role="main" class="ui-content">
3     <div class="ui-field-contain">
4       <label for="item">ToDo</label>
5       <input type="text" name="" id="item" placeholder="dosomething...">
6     </div>
7     <ul id="list" data-role="listview" data-inset="true">
8       <li data-role="list-divider">Item List</li>
9       <li>Orz</li>
10    </ul>
11  </div>
12 </div>
```

arch.



Include socket.io , sails.io , app

```
← → ↻ view-source:localhost:1337
137 <!--
138
139 Client-side Javascript
140 =====
141
142 You can import any js files in your `assets` directory as you would
143 normally: using <script></script> tags.
144
145 Here's an example of importing a few dependencies, in order:
146 -->
147 <!-- Bring in the socket.io client -->
148 <script type="text/javascript" src="/js/socket.io.js"></script>
149 <!-- then beef it up with some convenience logic for talking to Sails.js -->
150 <script type="text/javascript" src="/js/sails.io.js"></script>
151 <!-- listen on socket.io for incoming messages -->
152 <script type="text/javascript" src="/js/app.js"></script>
153
```

```
<script type="text/javascript" src="/js/socket.io.js"></script>
<script type="text/javascript" src="/js/sails.io.js"></script>
<script type="text/javascript" src="/js/app.js"></script>
```

js/app.js

```
app.js x
1 ▾ /**
2   * app.js
3   *
4   * This file contains some conventional defaults for working with Socket.io
5   * It is designed to get you up and running fast, but is by no means anything
6   *
7   * Feel free to change none, some, or ALL of this file to fit your needs!
8   */
9
10
11 ▾ (function (io) {
12   var socket = io.connect();
13   log('Connecting to Sails.js...');
14
15 ▾ socket.on('connect', function socketConnected() {
```

update ItemList

```
socket.on('connect', function socketConnected() {  
  socket.request('/foo', function(itemList){  
    $(itemList).each(function(idx, item){  
      $('#list').append('<li>'+item.name+'</li>');  
    });  
    $('#list').listview('refresh');  
  });  
});
```

Item List

marty

marty2

marty2

取得資料，並塞到listview後進行刷新

update broadcast

```
25 // Listen for Comet messages from Sails
26 socket.on('message', function messageReceived(msg) {
27   socket.request('/foo', function(itemList){
28     var title = $('#list :first-child')[0].outerHTML;
29     $('#list').html(title);
30     $(itemList).each(function(idx, item){
31       $('#list').append('<li>'+item.name+'</li>');
32     });
33     $('#list').listview('refresh');
34   });
35 });
```



localhost:1337/foo/create?name=okok

```
{
  "name": "okok",
  "createdAt": "2014-02-14T15:28:04.833Z",
  "updatedAt": "2014-02-14T15:28:04.833Z",
  "id": 74
}
```

remove Item

```
49 //remove Item
50 $('#list').on('click',function(e){
51     var text = $(e.target).html();
52     socket.request('/foo/find?name='+text,function(items){
53         socket.request('/foo/destroy/'+items[0].id);
54     });
55 });
```

/foo/find?name=\$val

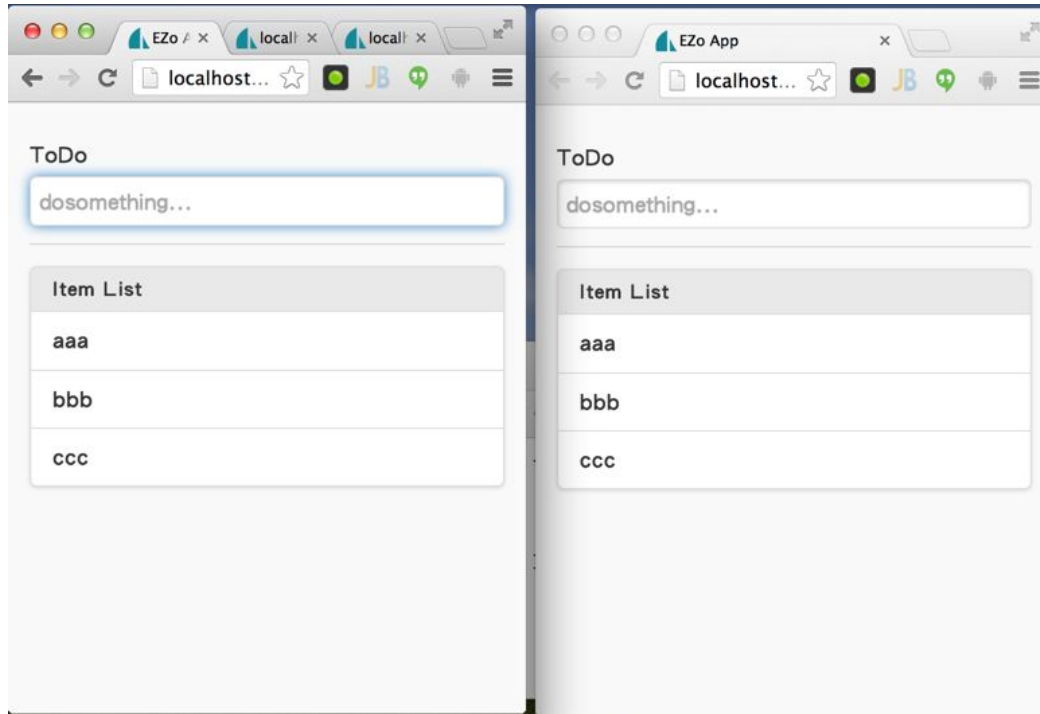
/foo/destroy/\$id

add Item

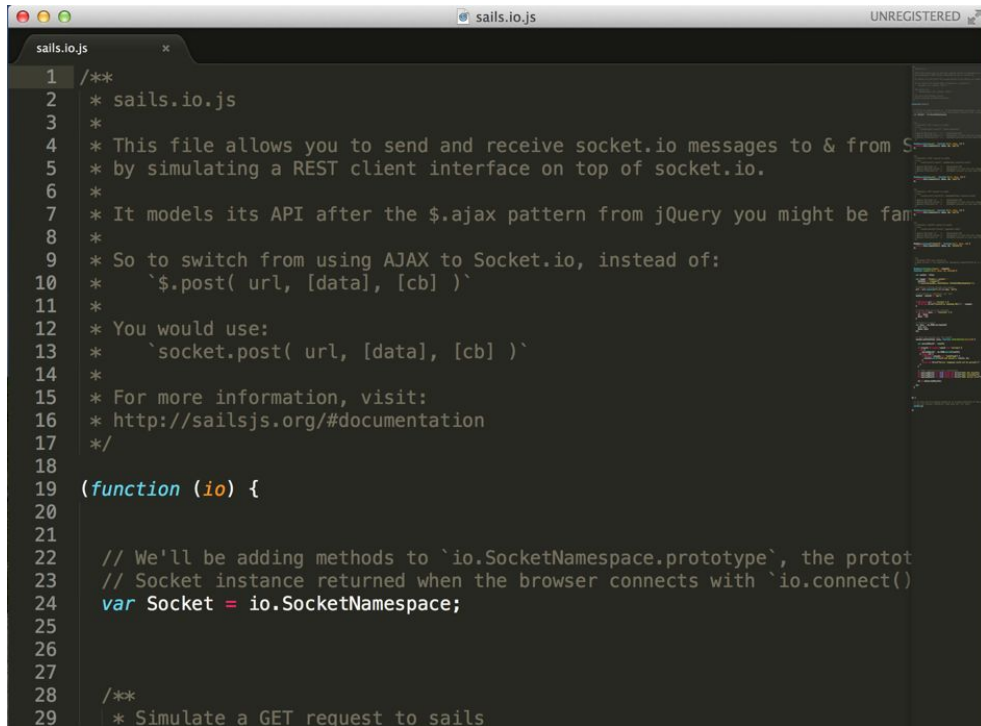
```
56 //add Item
57 $('#item').on('keypress',function(e){
58     if(e.which==13){
59         socket.request('/foo/create?name='+ $(this).val());
60         $(this).val('');
61     }
62 });
```

/foo/create?name=\$text

sync ToDoList



/js/sails.io.js



```
1 /**
2  * sails.io.js
3  *
4  * This file allows you to send and receive socket.io messages to & from S
5  * by simulating a REST client interface on top of socket.io.
6  *
7  * It models its API after the $.ajax pattern from jQuery you might be fam
8  *
9  * So to switch from using AJAX to Socket.io, instead of:
10 *   `$.post( url, [data], [cb] )`
11 *
12 * You would use:
13 *   `socket.post( url, [data], [cb] )`
14 *
15 * For more information, visit:
16 * http://sailsjs.org/#documentation
17 */
18
19 (function (io) {
20
21
22   // We'll be adding methods to `io.SocketNamespace.prototype`, the protot
23   // Socket instance returned when the browser connects with `io.connect()
24   var Socket = io.SocketNamespace;
25
26
27
28   /**
29   * Simulate a GET request to sails
```


FAQ - <http://goo.gl/aqb9gZ>

自己玩

step1: git clone <https://github.com/martyshu/jqm-sails-todolist.git>

step2: cd jqm-sails-todolist

step3: npm update

step4: open link: <http://localhost:1337/jqm.html>