

TDD Your Javascript with Backbone.js

Mike Jansen
@mjansen401

Intro

Goals

An Immediate Digression on Javascript





Backbone.js

Jeremy Ashkenas



Structure

Model

```
var Task = Backbone.Model.extend({  
  defaults: {  
    'complete': false  
  }  
});
```

Model

```
var task = new Task({  
  description: "Set up lightning rod on  
  top of laboratory"  
});  
task.get("description");  
  
> "set up lightning rod on top of  
laboratory"
```

Collection

```
var Tasks = Backbone.Collection.extend({  
  model: Todo.models.Task,  
  url: "tasks"  
});
```

Collection

```
var tasks = new Tasks();  
tasks.create(  
{  
  description: "Dig up some brains at the  
  cemetery"  
});
```

Pick and Choose

Router

```
var TodoRouter = new
Backbone.Router.extend({
  initialize: function(options) {
    this.tasks = new Tasks(options.tasks)
    this.tasksView = new
    TasksView({collection: this.tasks});
  },
  ...
});
```

Router

...

```
routes: {  
  "": "loadTasks",  
  "tasks/:id/edit": "editTask"  
},
```


Router

...

```
editTask: function(id) {  
  var view = new TaskEditView({id: id,  
    collection: this.tasks});  
  view.render();  
}  
});
```

Router

```
Todo.router = new TodoRouter(options);  
Backbone.history.start();
```

Router



www.drfrankenstein.com/todo#tasks/34/edit

So...



**What about the
DOM?**

View

**“...more convention than they
are code”**

View

```
var TaskView = Backbone.View.extend({  
  tagName: 'div',  
  className: 'task-list',  
  
  ...  
});
```


View

```
events: {  
  'keyup #new-task' : 'createOnEnter'  
},
```

View

```
createOnEnter : function (event) {  
    ...  
    this.collection.create({ description:  
    inputValue });  
    ...  
},
```

View

```
render : function () {  
  this.collection.each(this.addTask);  
  return this;  
},
```

View

```
addTask: function (task) {  
  var view = new  
    Todo.views.TaskView({ model: task });  
  var taskEl = view.render().el;  
  $( '.tasks' ).prepend(taskEl);  
},
```

Structure



Now what?



**How do you test
drive client side
Javascript?**

How do you write clean Javascript?

Scenarios

Scenario One

Scenario: Web app with no Javascript.

Feature: Show additional fields if user wants to add additional emergency contacts.

Scenario One

On document.ready...

- A) Bind a function to click event, find hidden fields and show them.
- B) Create "App" namespace, initialize and bind event to a "showFields" function.

Scenario Two

Hooray, you namespaced your Javascript!

Scenario Two

Hooray, you namespaced your Javascript!

But you didn't test it.

Scenario Two

Scenario: You've added Jasmine to your project and open up AppSpec.js

You have a jQuery selector that is finding and showing a class.

You ponder fixtures.

Scenario Two

Do you:

A) Copy the full html from your app into a fixture.html file, then load in your spec

B) Set a fixture that looks like:

```
'<button id="add" type="button">Add  
  Contact</button><div id="extra_fields"  
  style="display:none;">some fields</div>'
```

Scenario Three

Scenario: You have your fixture and are ready to test.

Scenario Three

Do you:

A) Use `$('.add').click();` to trigger the event, which makes the fixture element visible

B) 1: Spy on the bind function, 2: call `showFields` and check the fixture afterwards

Scenario Four

Scenario: Add client side validation for SSN. If not 9 digits, show error message right away.

Scenario Four

Do you:

- A) Use `$('.ssn').trigger('change')` to trigger the event, which leads to display of error
- B) 1: Spy on the bind function, 2: call `showFields` and check the fixture afterwards

**Why not just trigger
events?**

With option A:

- 1) Verifying binding of event**
- 2) Checking logic of valid vs invalid**
- 3) Checking for correct message**
- 4) Traversing the DOM (twice)!**

Scenario Five

Fed up with your multiple responsibility Javascript, you attend an amazing session on Backbone.js in McHenry County.

Scenario Five

You refactor your Javascript to use
Backbone.js

Scenario Five

Now:

- 1) Verify binding of event
- 2) Check logic of valid vs invalid
- 3) Check for correct message
- 4) Traversing the DOM

Scenario Five

Scenario: You want to refactor the Javascript that shows hidden fields in your HTML.

Scenario Five

Do you refactor to:

- A) Backbone View event binding and function that unhides the View's el
- B) Backbone View event binding and function that renders a JST

A note on jQuery

jQuery

“If there’s a DOM method, there’s probably a problem with it somewhere, in some capacity.”

-John Resig, creator of jQuery

jQuery

Know why you're using jQuery

Summary

Backbone.js

It's a framework.

Clean Javascript

Namespace your Javascript

Clean Javascript

Minimize your fixtures

Clean Javascript

Separate your responsibilities/
unit test your code

Thanks!

<https://github.com/mjansen401/todo>

https://github.com/mjansen401/todo_local_storage

mike@8thlight.com
@mjansen401