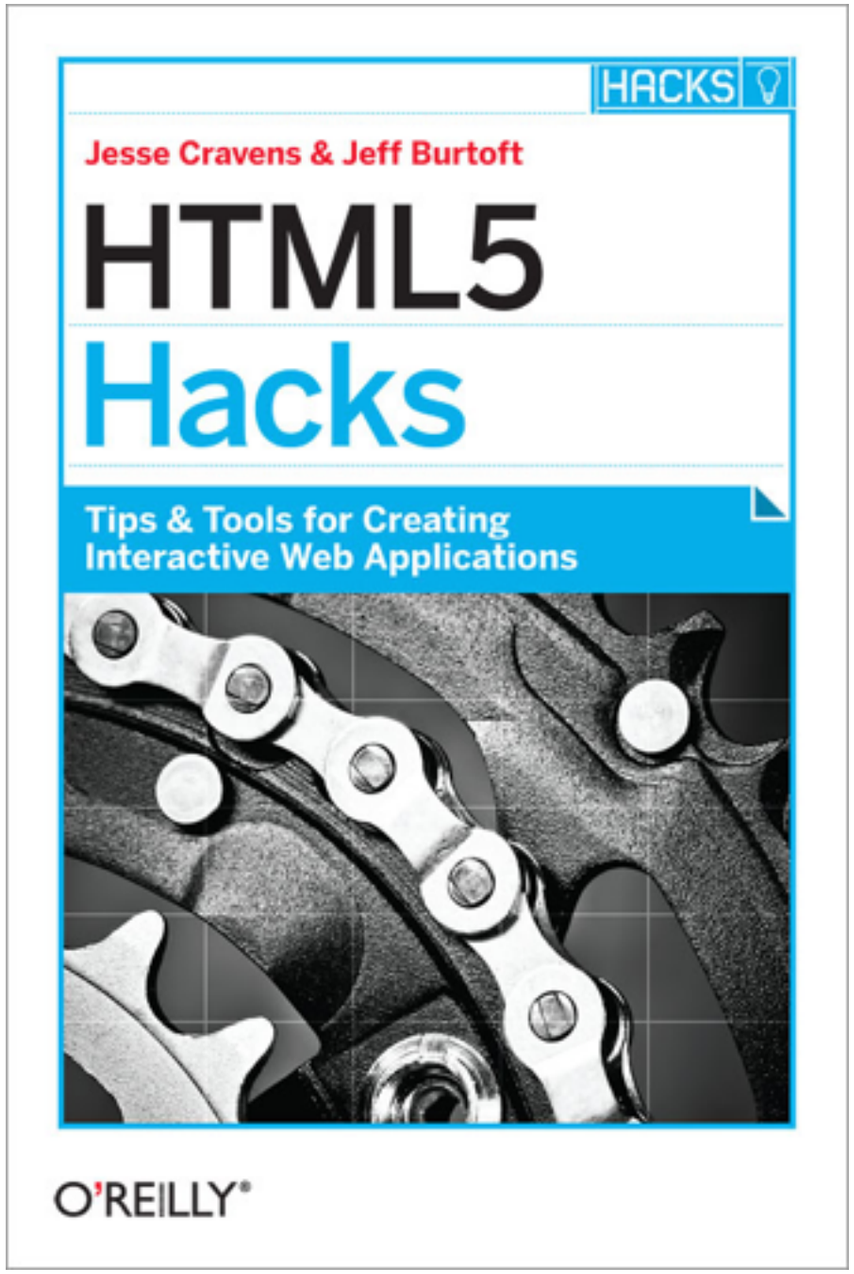
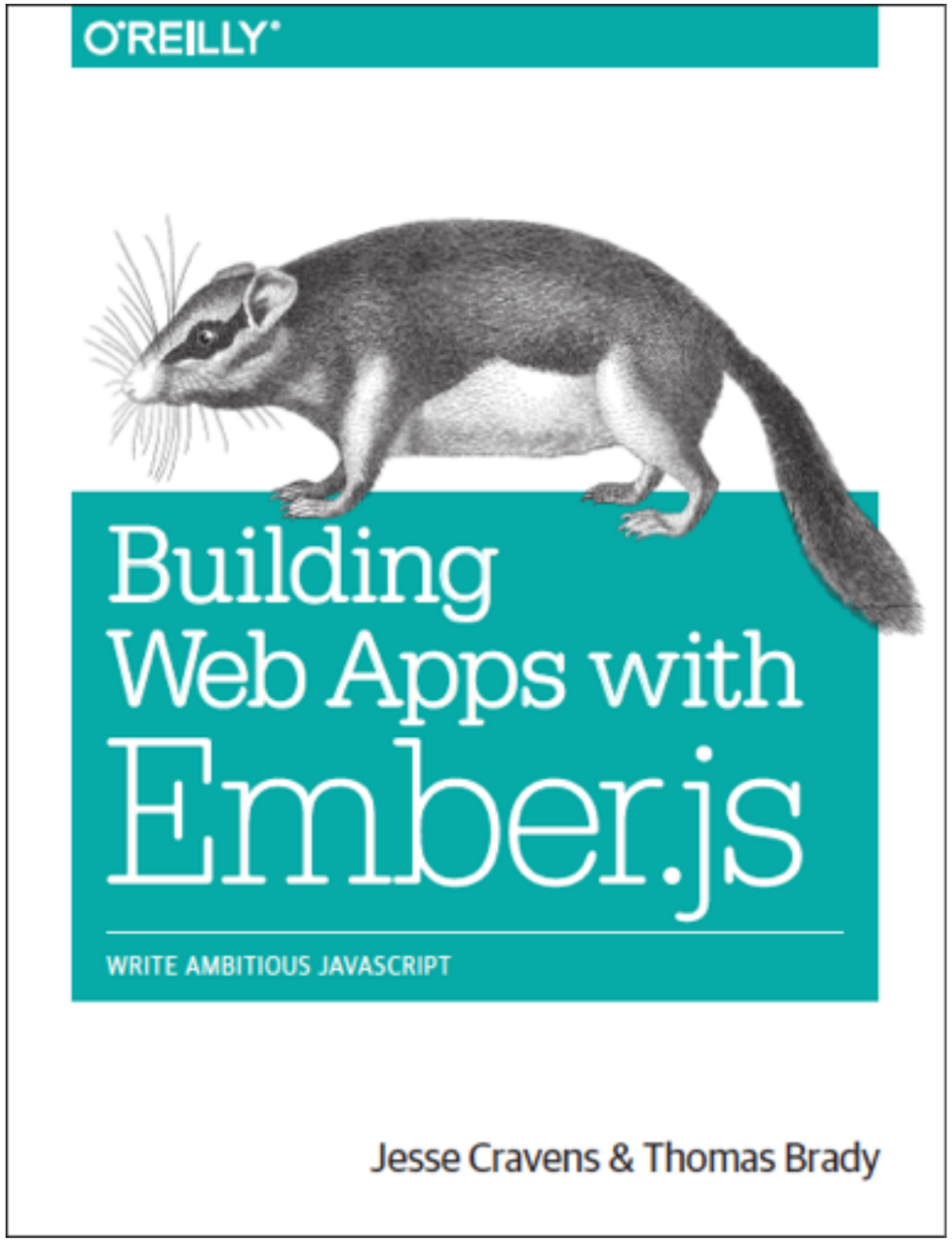
BUILDING WEB APPS WITH EMBER.JS

UX DEVELOPMENT at frog

@jdcravens
github.com/jessecravens
jessecravens.com
principal web engineer | frog Austin

O'REILLY MEDIA

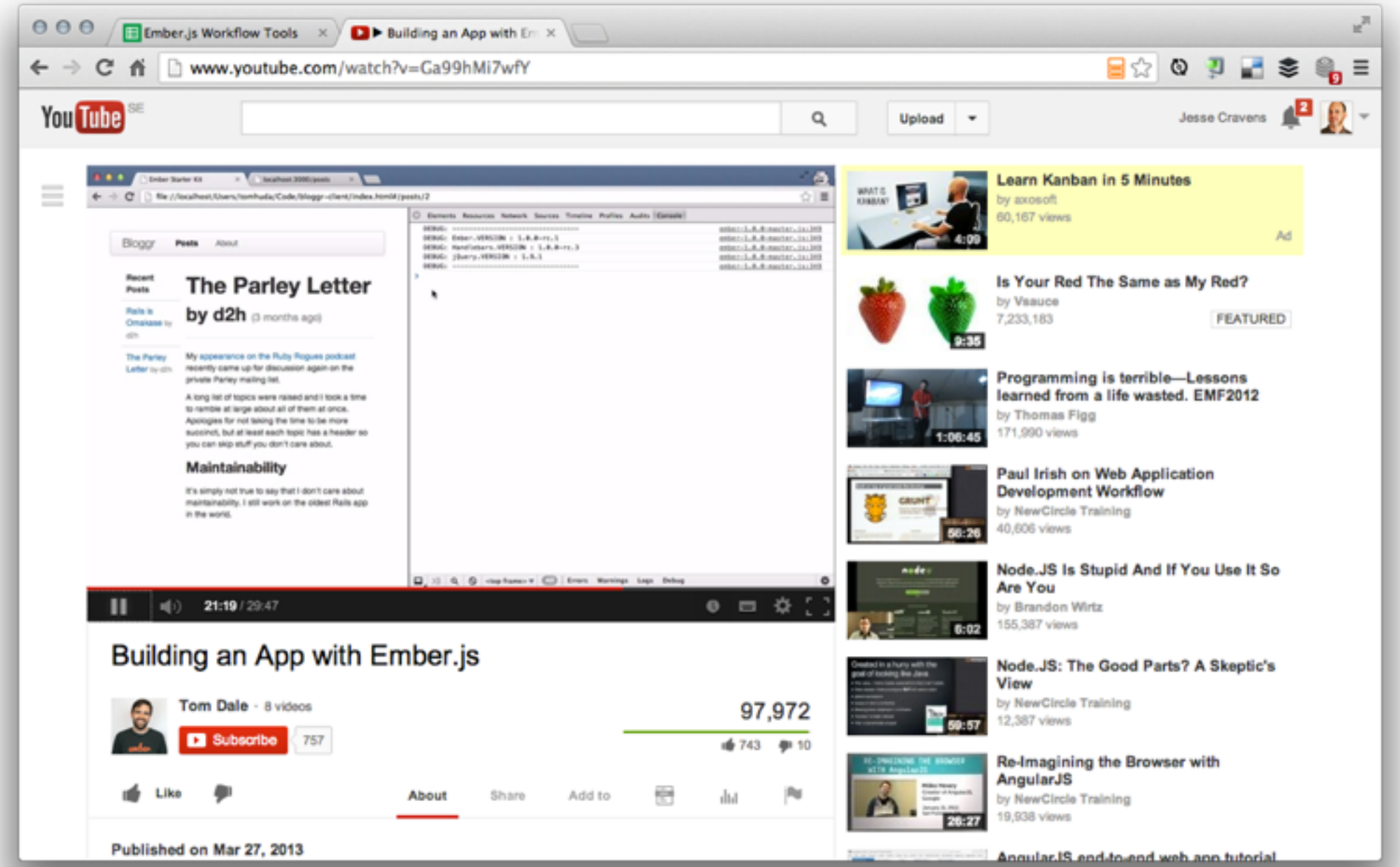




GETTING STARTED

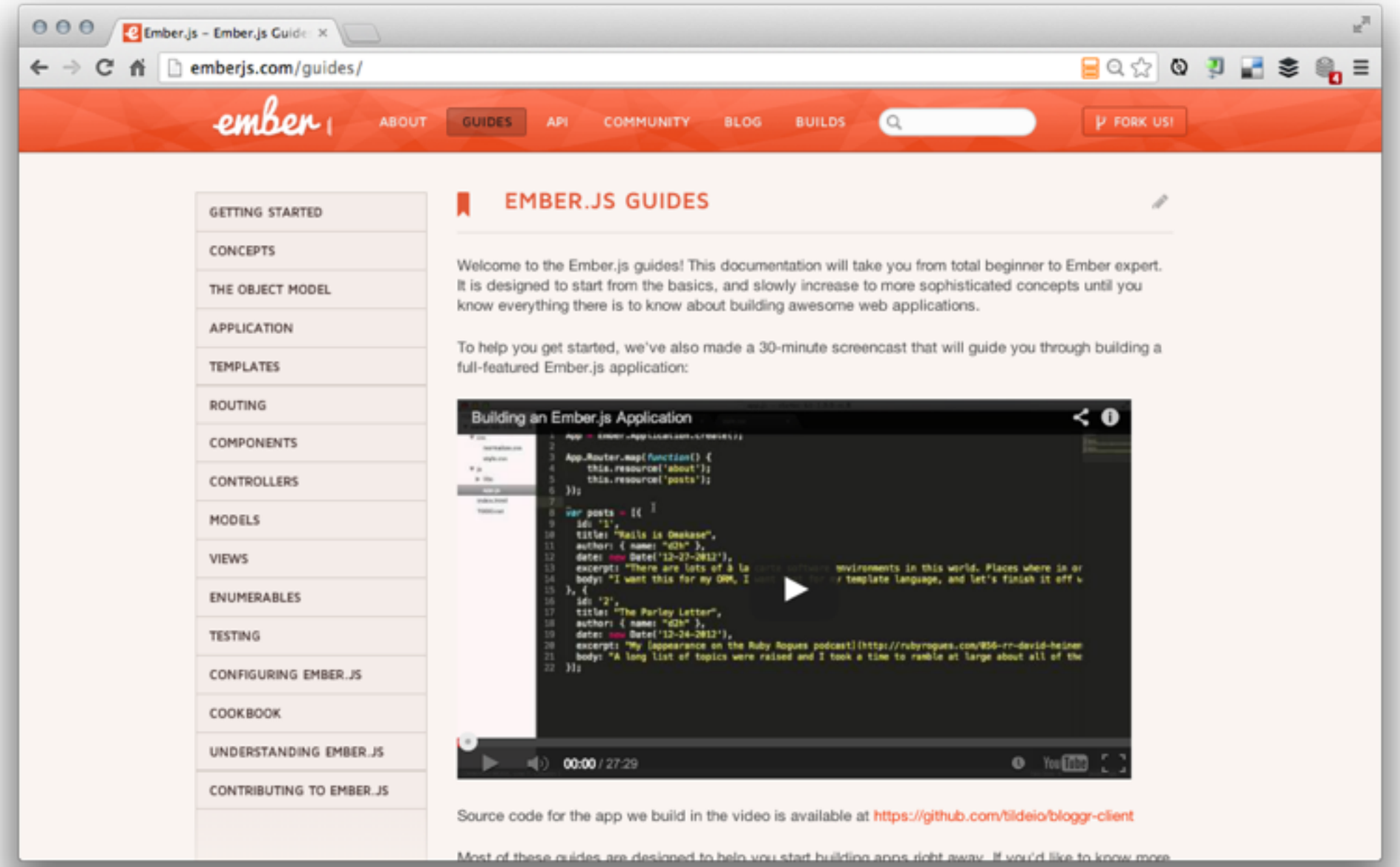
chapter 1

BLOGGR-CLIENT



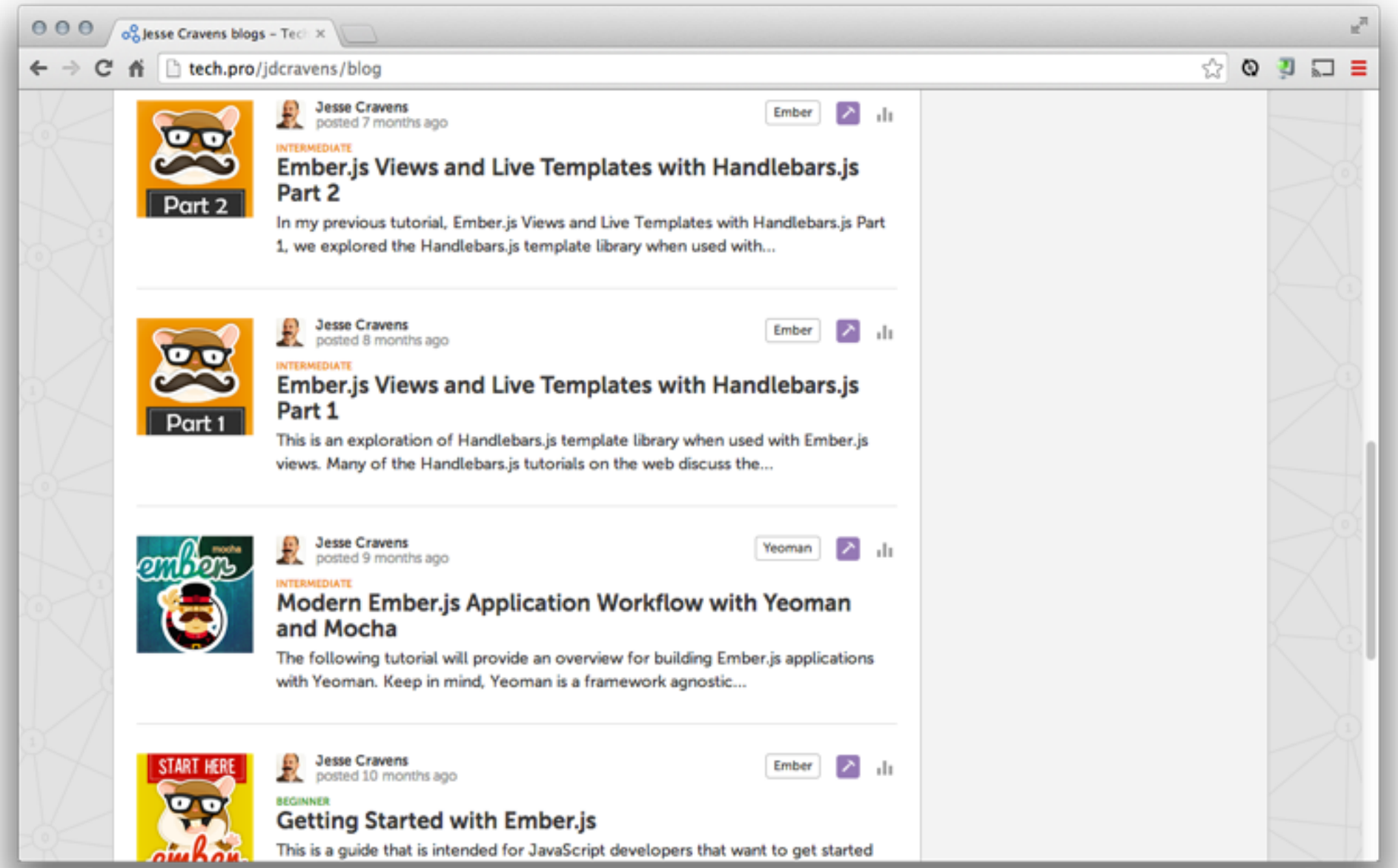
EMBER GUIDES

TODO APP

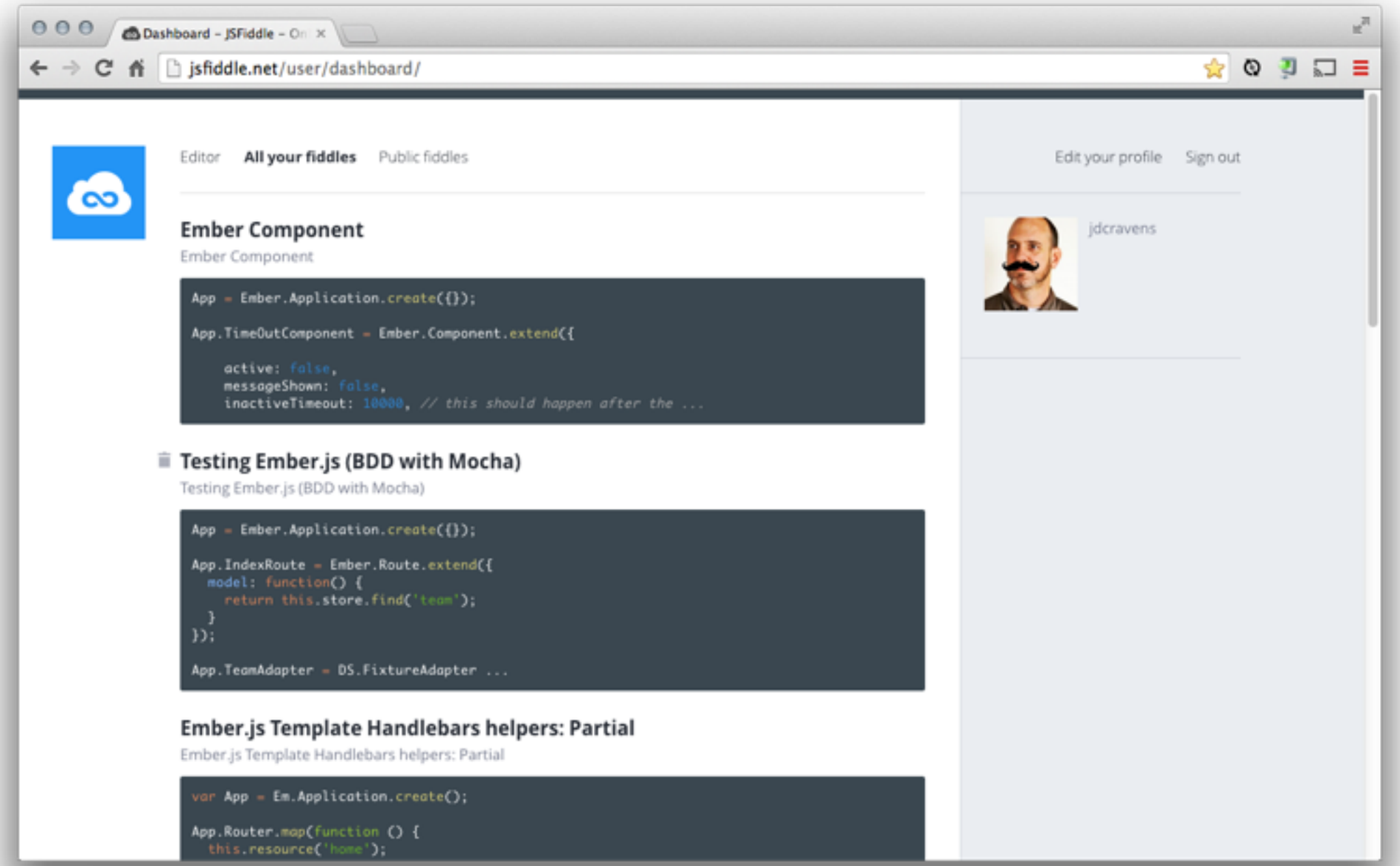


REPOS | FIDDLES | GISTS

TECH.PRO


Jesse Cravens blogs – Tec
tech.pro/jdcravens/blog
Jesse Cravens
posted 7 months ago
Ember
INTERMEDIATE
Ember.js Views and Live Templates with Handlebars.js Part 2
In my previous tutorial, Ember.js Views and Live Templates with Handlebars.js Part 1, we explored the Handlebars.js template library when used with...
Part 2
Jesse Cravens
posted 8 months ago
Ember
INTERMEDIATE
Ember.js Views and Live Templates with Handlebars.js Part 1
This is an exploration of Handlebars.js template library when used with Ember.js views. Many of the Handlebars.js tutorials on the web discuss the...
Part 1
Jesse Cravens
posted 9 months ago
Yeoman
INTERMEDIATE
Modern Ember.js Application Workflow with Yeoman and Mocha
The following tutorial will provide an overview for building Ember.js applications with Yeoman. Keep in mind, Yeoman is a framework agnostic...
Jesse Cravens
posted 10 months ago
Ember
BEGINNER
Getting Started with Ember.js
This is a guide that is intended for JavaScript developers that want to get started

FIDDLES


Dashboard – JSFiddle – On
jsfiddle.net/user/dashboard/
Editor
All your fiddles
Public fiddles
Edit your profile
Sign out
jdcravens
Ember Component
Ember Component
App = Ember.Application.create({});
App.TimeOutComponent = Ember.Component.extend({
active: false,
messageShown: false,
inactiveTimeout: 10000, // this should happen after the ...
Testing Ember.js (BDD with Mocha)
Testing Ember.js (BDD with Mocha)
App = Ember.Application.create({});
App.IndexRoute = Ember.Route.extend({
model: function() {
return this.store.find('team');
}
});
App.TeamAdapter = DS.FixtureAdapter ...
Ember.js Template Handlebars helpers: Partial
Ember.js Template Handlebars helpers: Partial
var App = Em.Application.create();
App.Router.map(function () {
this.resource('home');

WAIT...

“SHIP CODE!”

4 THEMES

ADDRESS THE MORE COMPLEX USE CASES

WORK FRONT TO BACK

DESIGN WITH YOUR MEDIUM

CONSIDER PROD EARLY

3 SECTIONS

I. INTRODUCTION

II. PROTOTYPING WITH EMBER

III. PRODUCTION EMBER

10 CHAPTERS

PRELUDE

I. INTRODUCTION

GETTING STARTED
EMBER BASICS

II. PROTOTYPING WITH EMBER

EMBER TOOLING, BOILERPLATE, AND WORKFLOW

TEMPLATES

THE ROUTER, ROUTES AND MODELS

CONTROLLERS, VIEWS, DATA-BINDING AND EVENTS

III. PRODUCTION EMBER

PERSISTING DATA

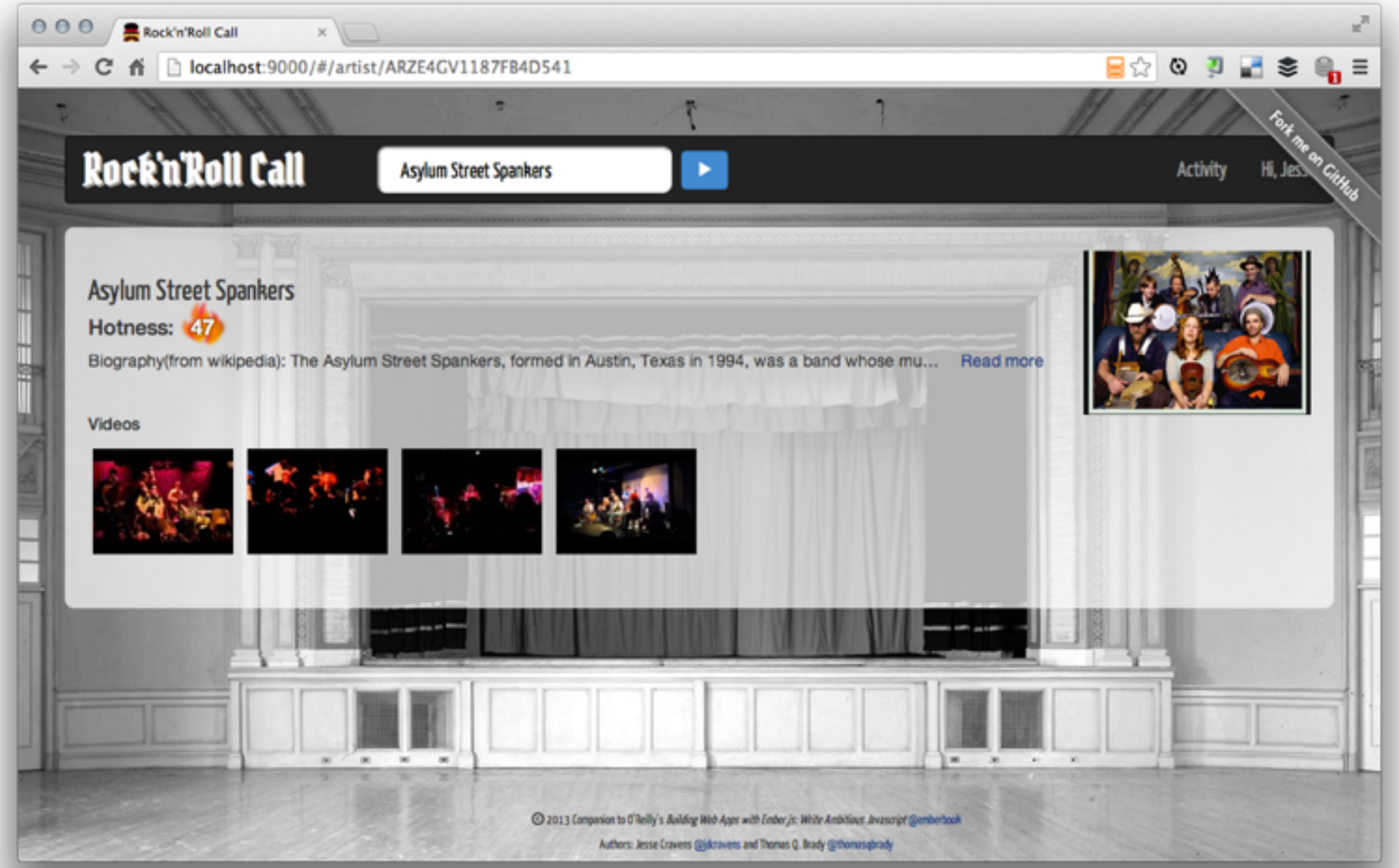
BUILDING AN EMBER BACKEND

EMBER COMPONENTS

EMBER TESTING

ADDRESS THE MORE COMPLEX USE CASES

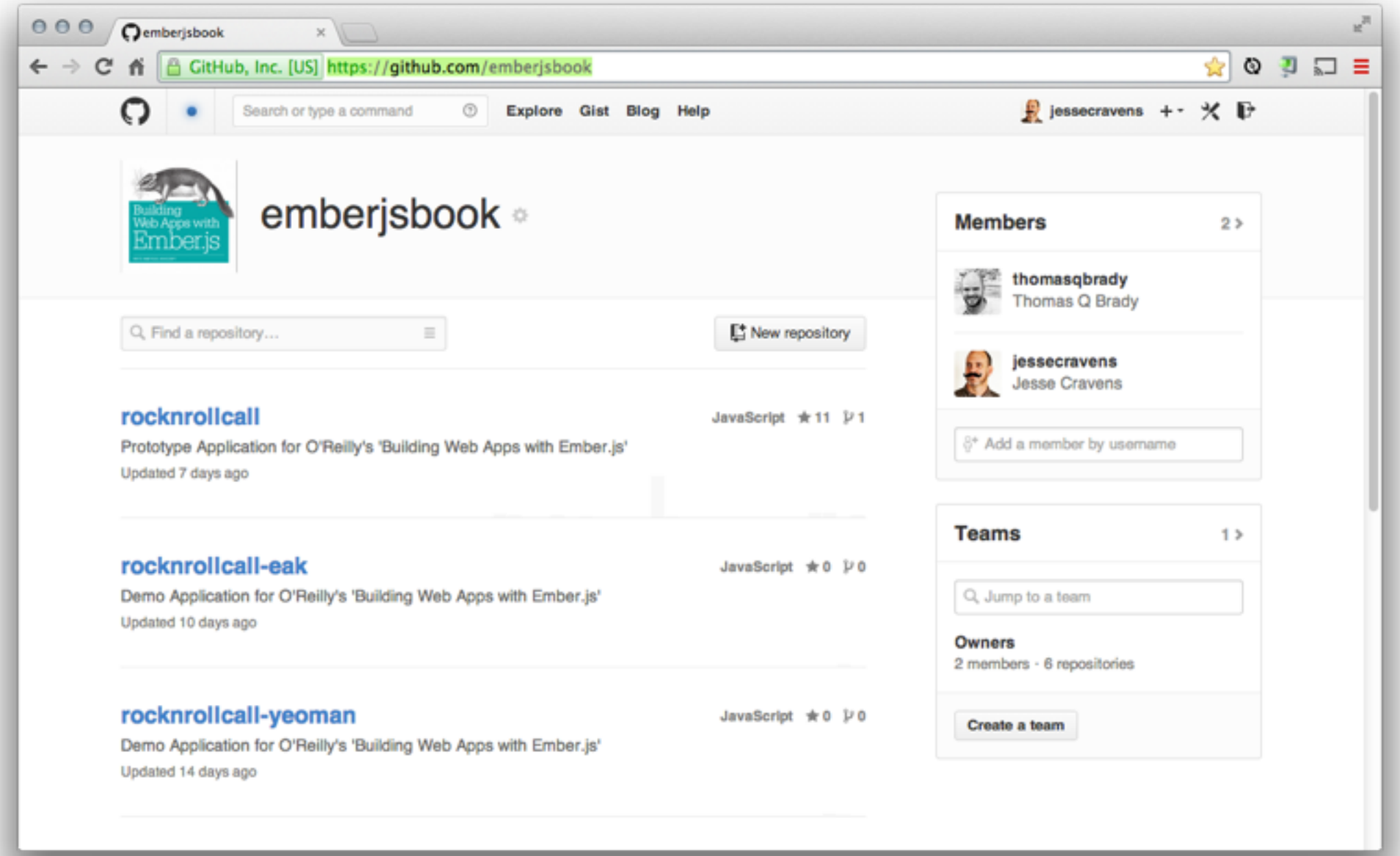
ROCKNROLLCALL



MOVIN' BEYOND THE BASICS

- Ember Tooling
- Ember Inspector
- Ember Boilerplates / Generators
- App Initializers
- RSVP Promises
- Needs
- Ember Components
- 3rd Party Integration (jQuery, D3)
- Ember-Testing
- Ember-Data
- JSON Serialization and Format

COMPANION CODE



WORK FRONT TO BACK

RAPID DEV

1ST

Early Design: HTML, Diagram Controllers

URL Driven: State Manager and Routes First

Populate the Controllers with Dummy Data

Models, Fixtures, FixtureAdapter

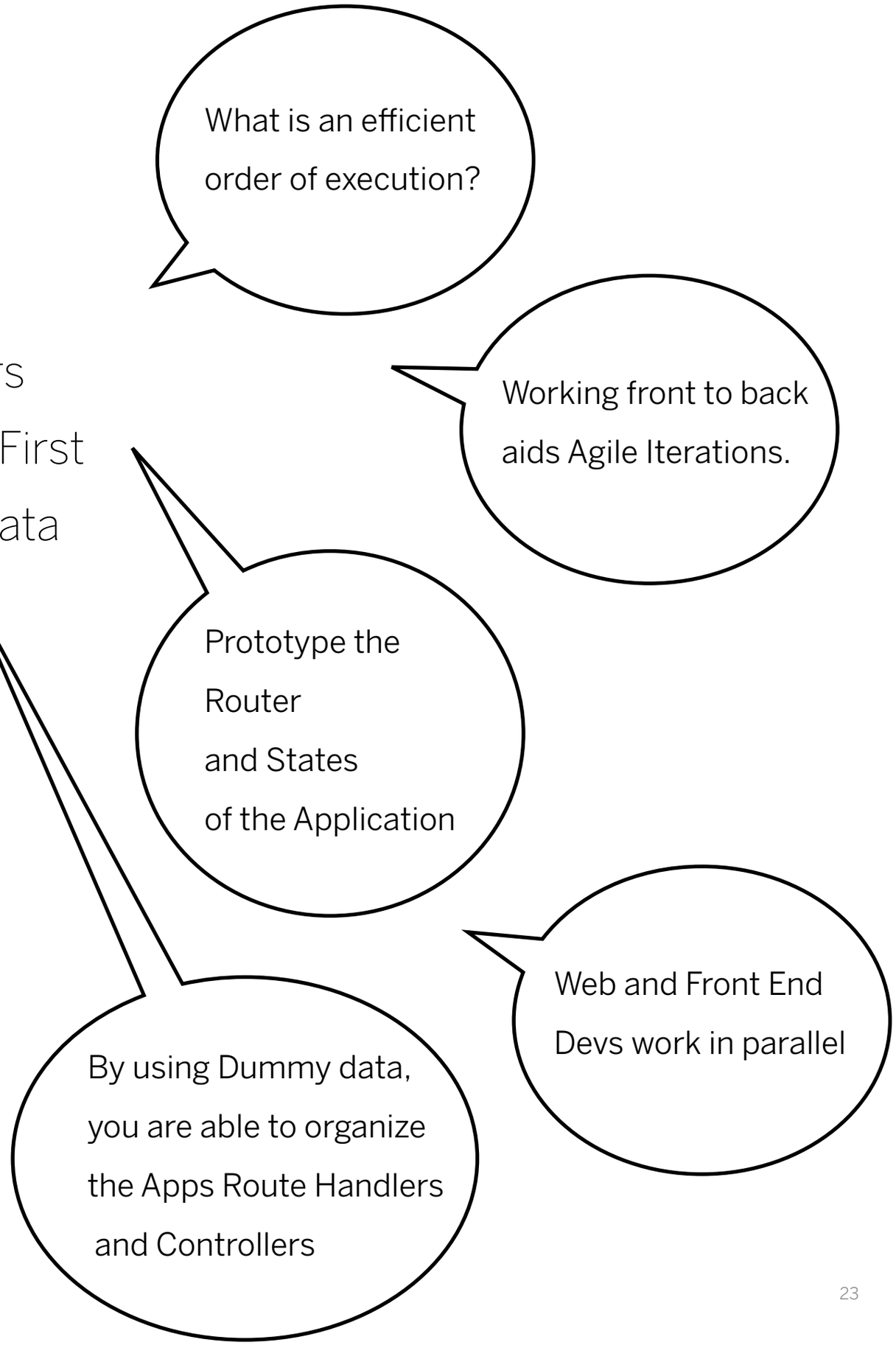
2ND

RESTAdapter

Build Server-Side Routes/Endpoints

Serialization and Formatting JSON

Remote Data Store and DB Sync



RAPID DEV

CONCEPTUAL

WEB/SERVICES DEVELOPMENT

FRONT END DEVELOPMENT

TEMPLATES

ROUTER

ROUTE HANDLERS

CONTROLLERS

VIEWS/COMPONENTS

MODELS

FIXTURE ADAPTER

REST ADAPTER

SERVICES (API DESIGN)

MVC, SPA (Bootstrap Object)

SERVICES (IMPLEMENTATION)

DEPLOY

EMBER BASICS

chapter 2

MINIMAL VIABLE APP

- Dummy Data
- Bindings
- Handlebars View helper
- Application Initialization

CONSIDER PROD EARLY

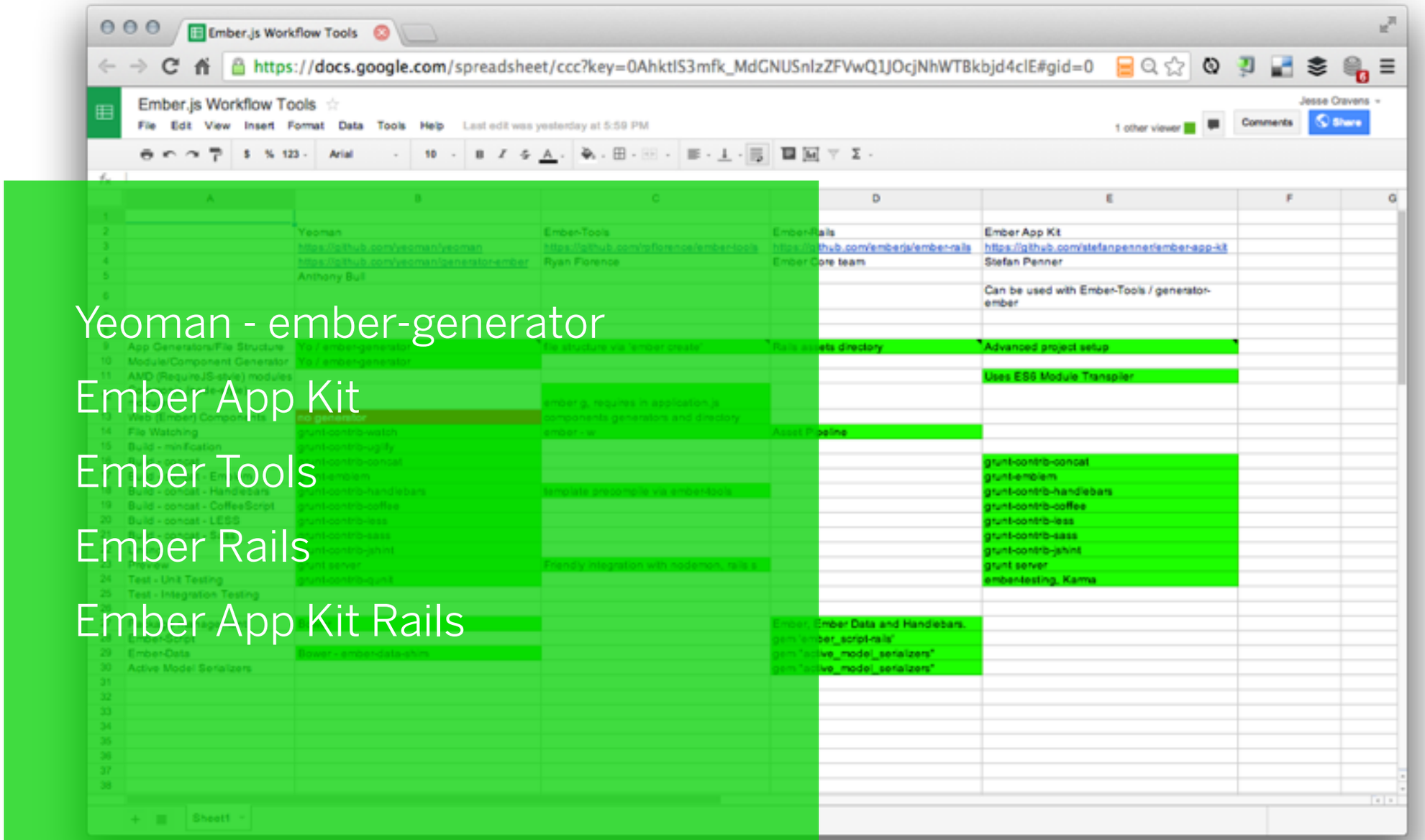
AUTOMATION

WORKFLOW

EMBER TOOLING

chapter 3

TOOLS MATRIX



Yeoman - ember-generator

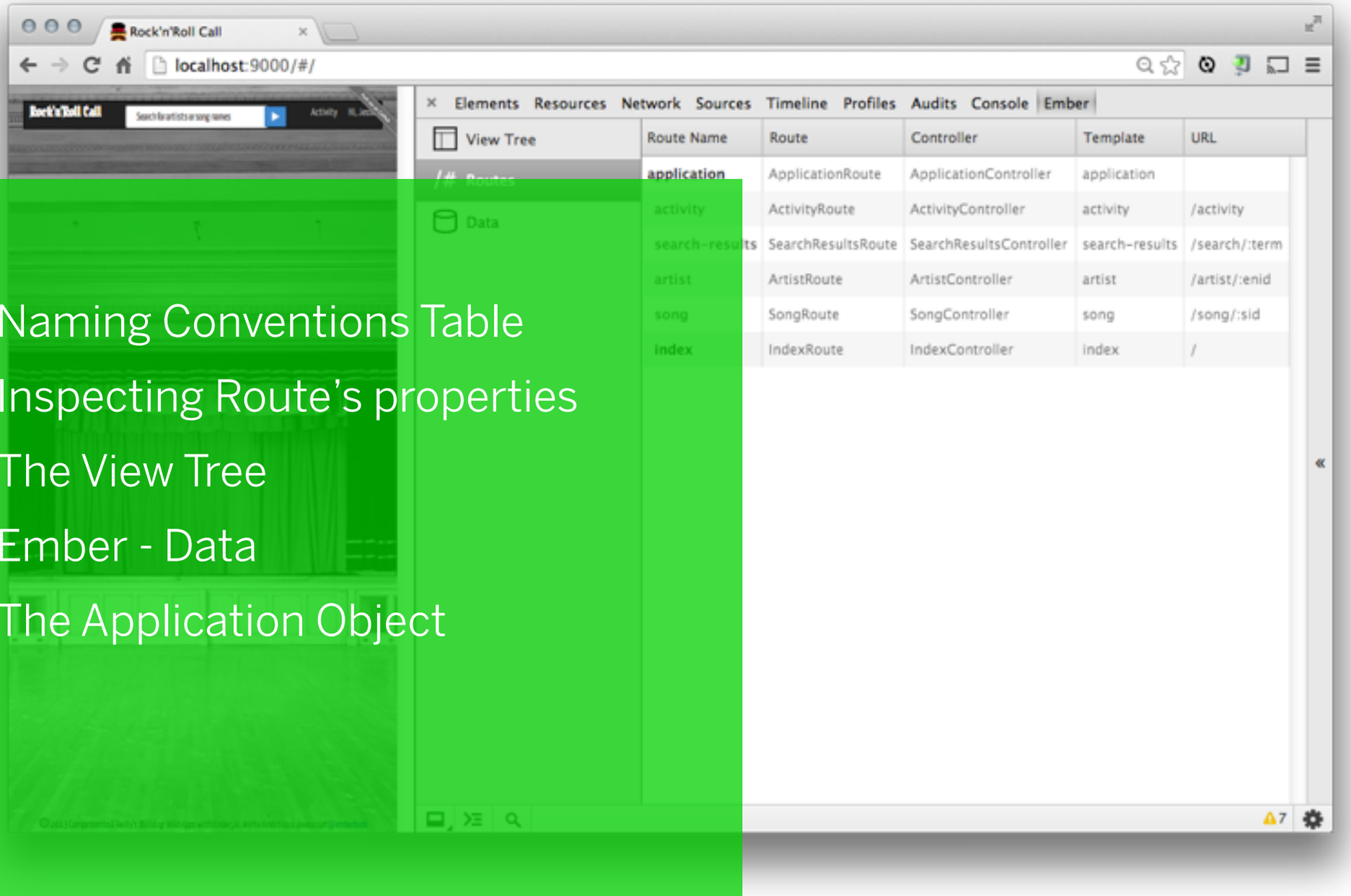
Ember App Kit

Ember Tools

Ember Rails

Ember App Kit Rails

EMBER INSPECTOR



Naming Conventions Table

Inspecting Route's properties

The View Tree

Ember - Data

The Application Object

PROTOTYPING W/ EMBER

section II

REQs AND EARLY DESIGN

BASIC REQs

- Search by Keyword to Find Transient Data (Songs and Artists)
- Persist the Activity of Selecting Songs and Artists into LocalStorage
- Display Detail information to the User
- Show a Visualization of the ‘Hotness’ of your Searches

BASIC FLOW

- Application Route, application.hbs, view App.SearchBox
- Transitions to the SearchResultsRoute passing the query
- Calls the searchByName(), searchByTitle on Artists/Songs Controllers
- Populates the artists, songs arrays on the SearchResultsController
- In search-results template iterate over those arrays
- Click on takes the user to the Detail page

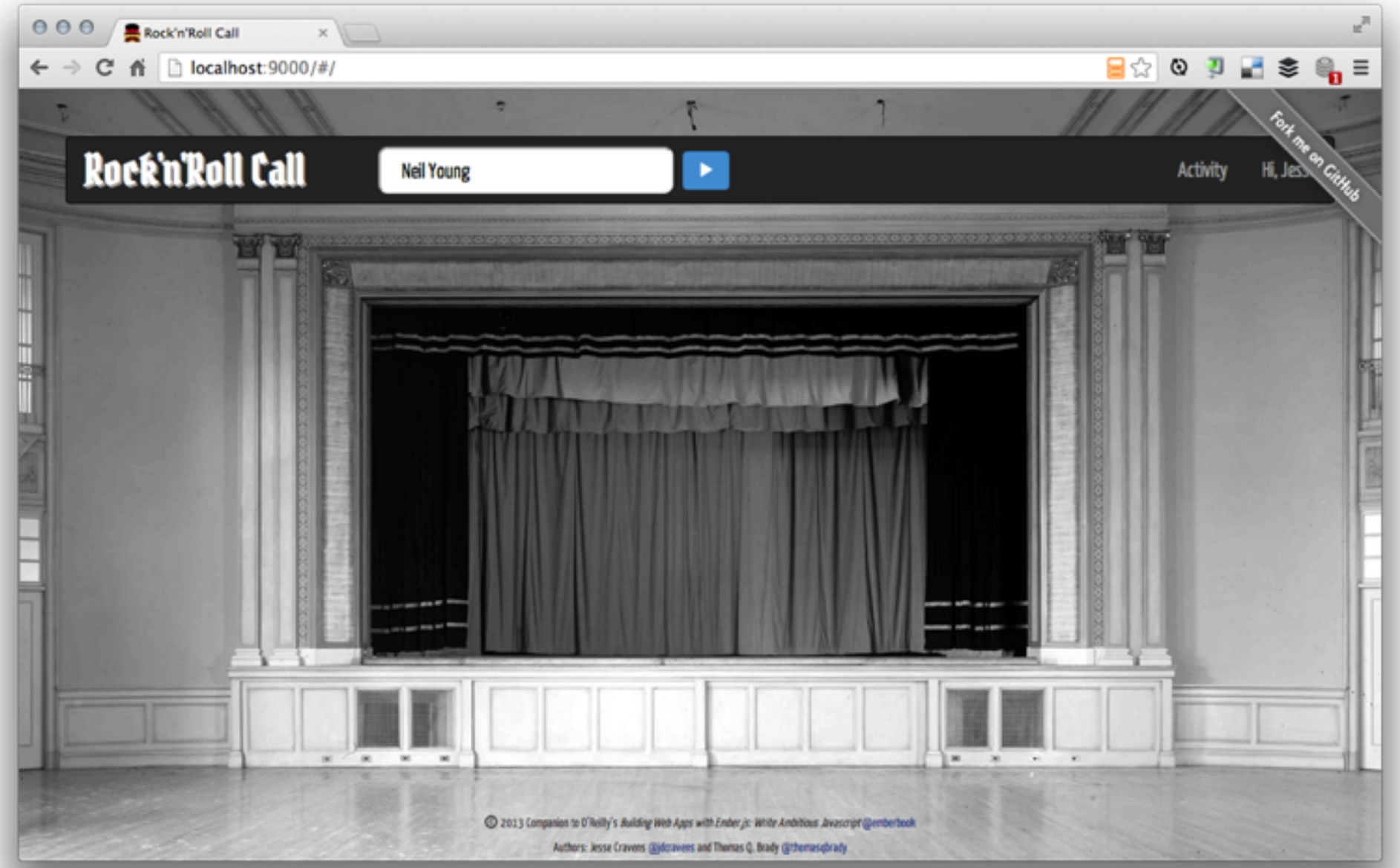
TEMPLATES

chapter 4

DESIGN WITH YOUR MEDIUM

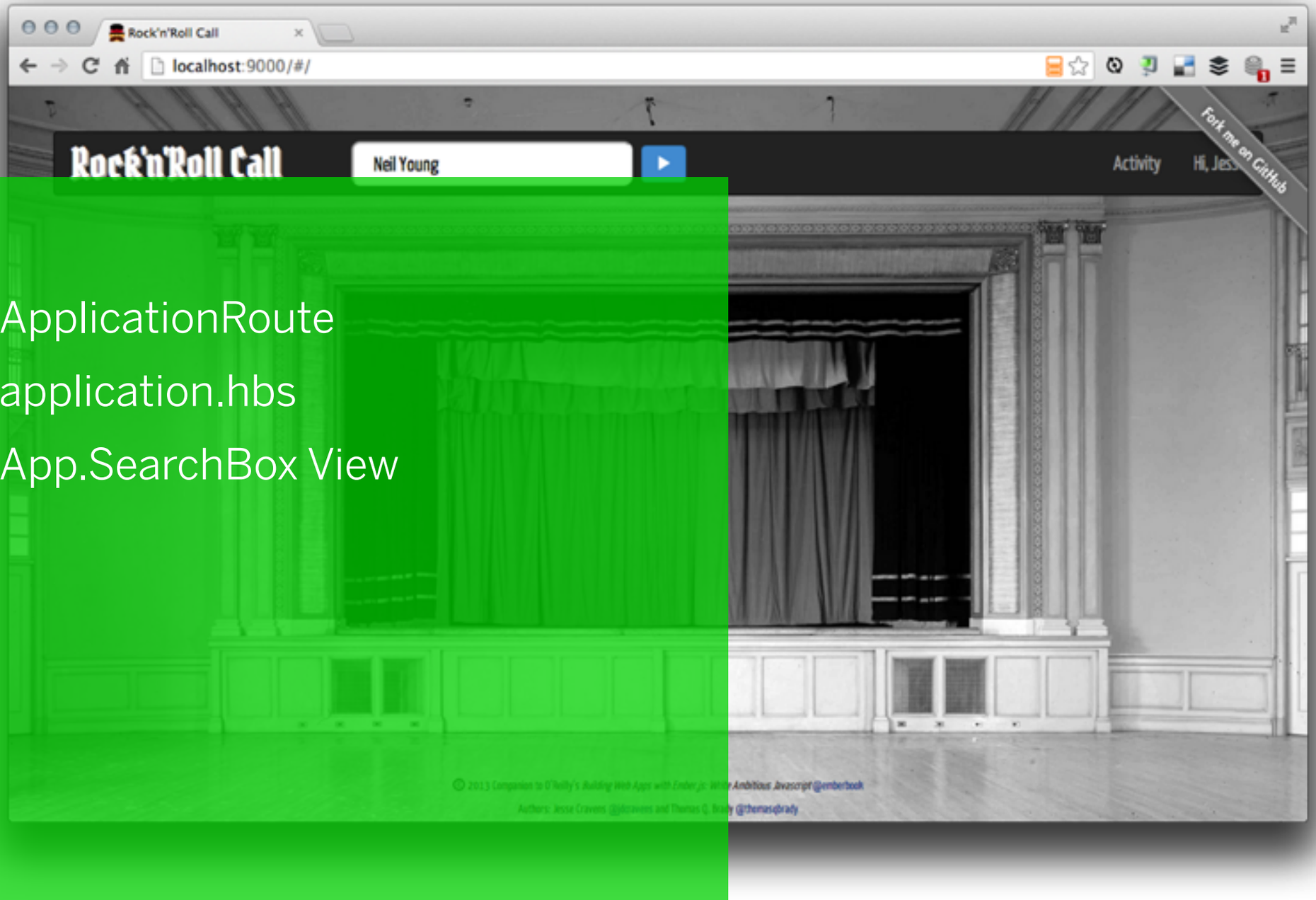
DESIGN IN .HBS

APPLICATION / SEARCH STATE



DESIGN IN .HBS

APPLICATION / SEARCH STATE



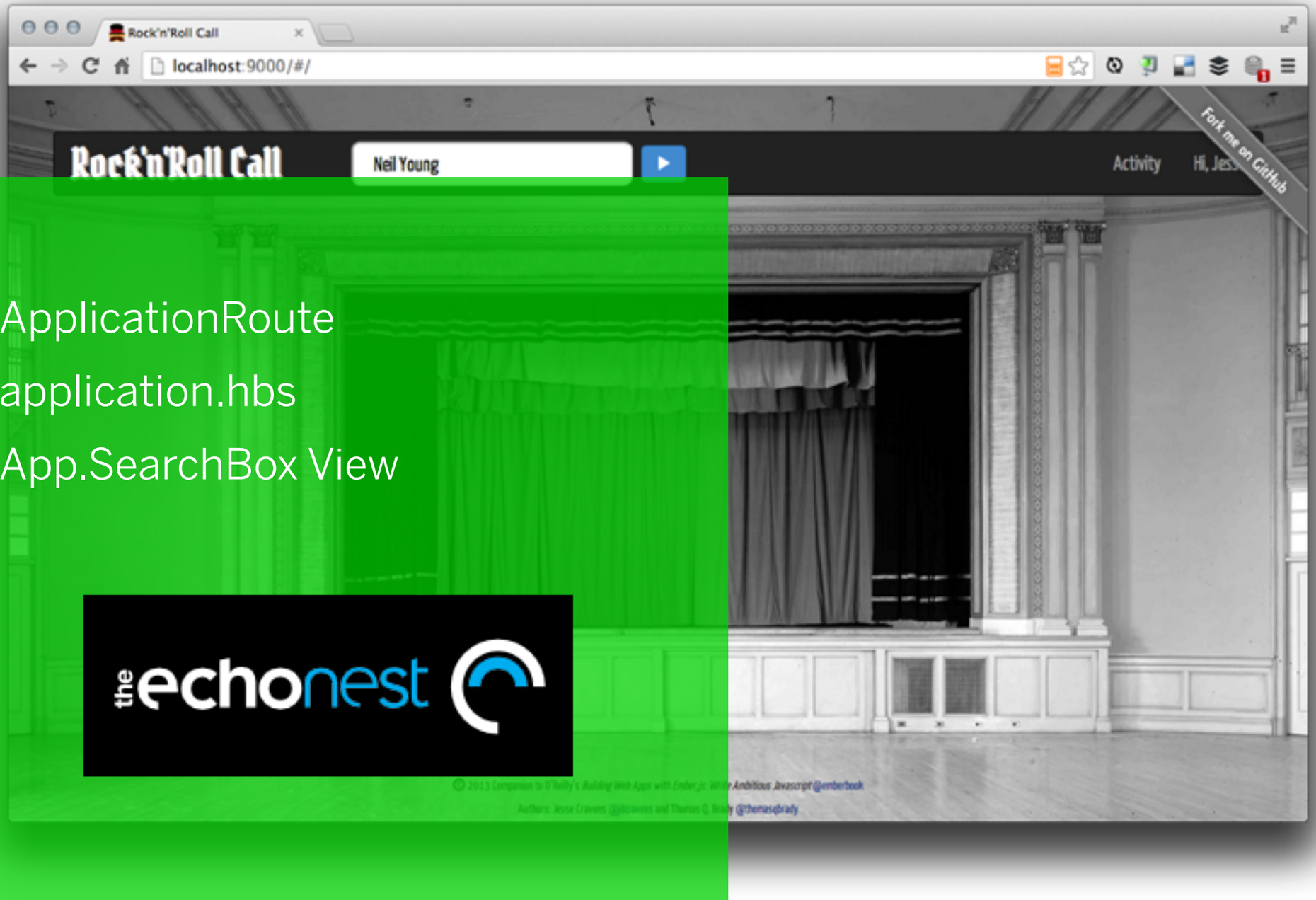
ApplicationRoute

application.hbs

App.SearchBox View

DESIGN IN .HBS

MUSIC WEB SERVICE



ApplicationRoute

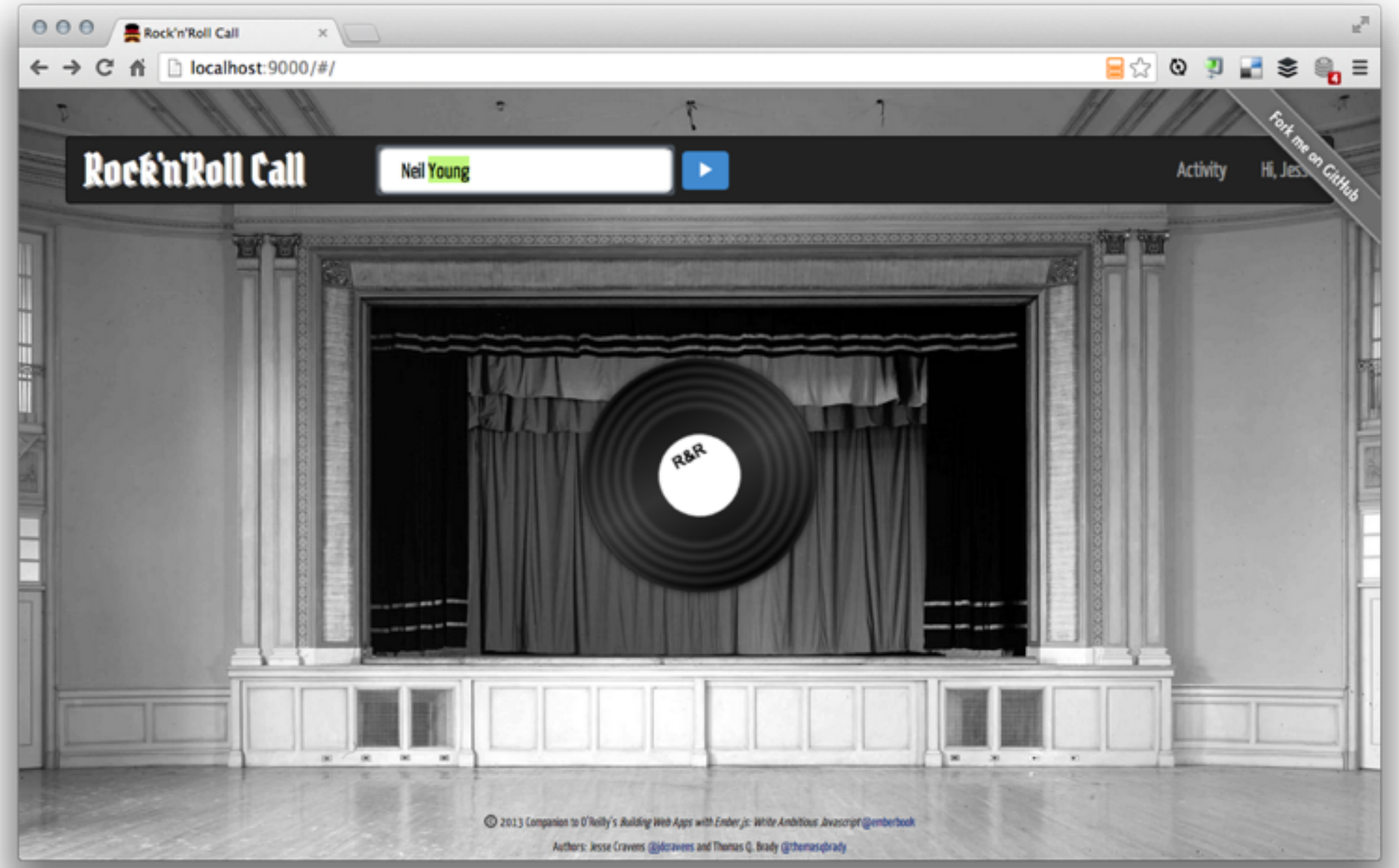
application.hbs

App.SearchBox View

the echonest

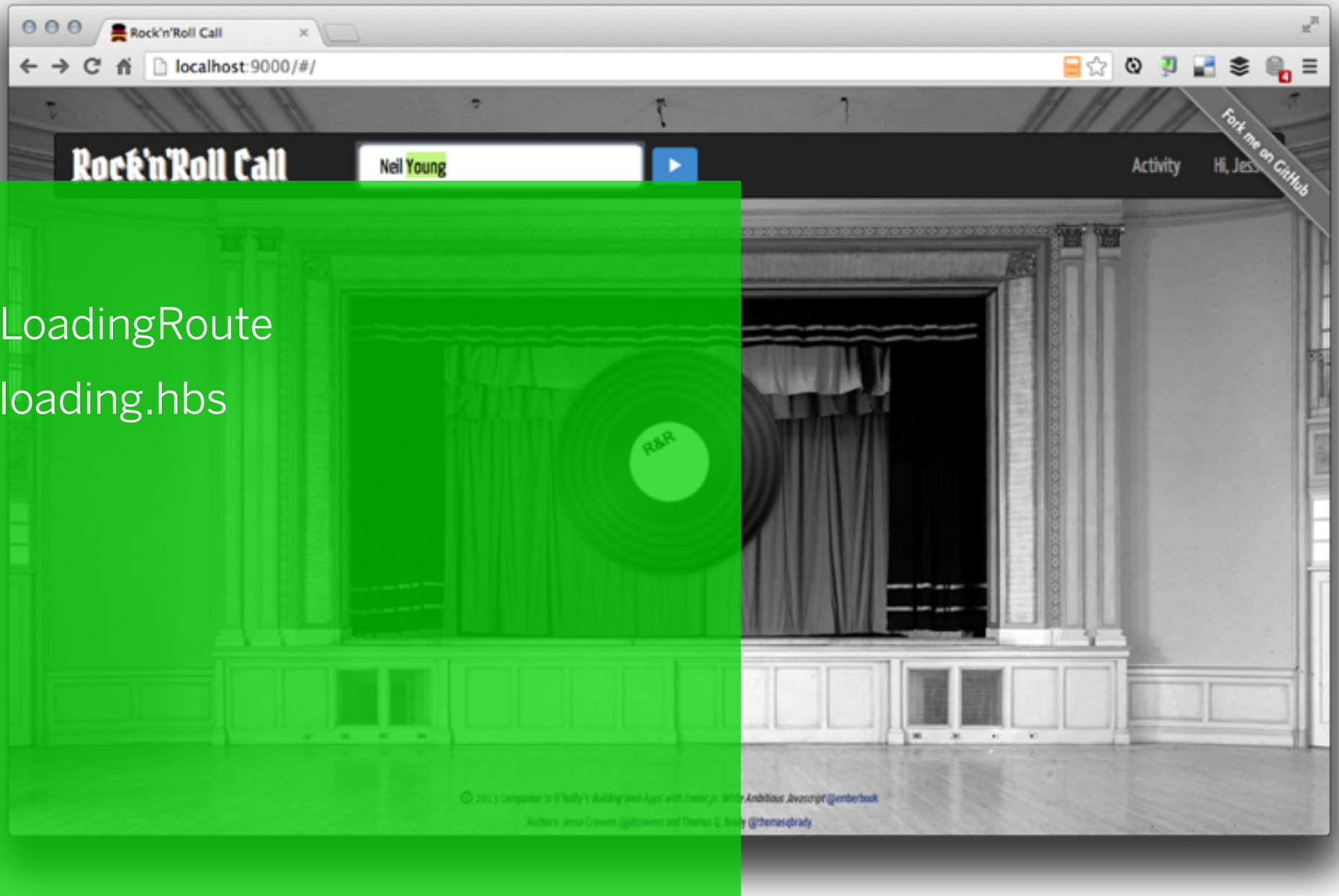
DESIGN IN .HBS

LOADING STATE



DESIGN IN .HBS

LOADING STATE

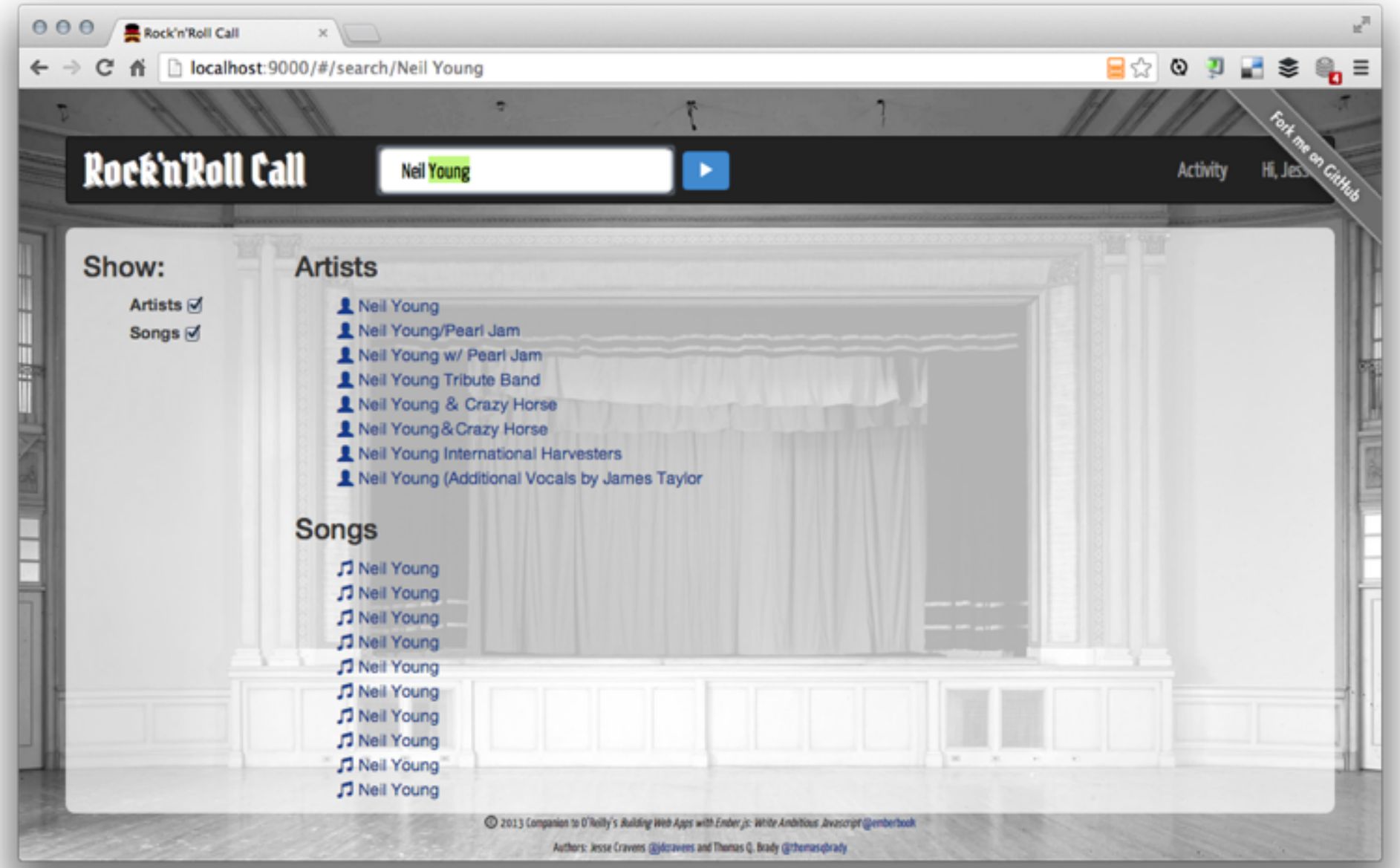


LoadingRoute

loading.hbs

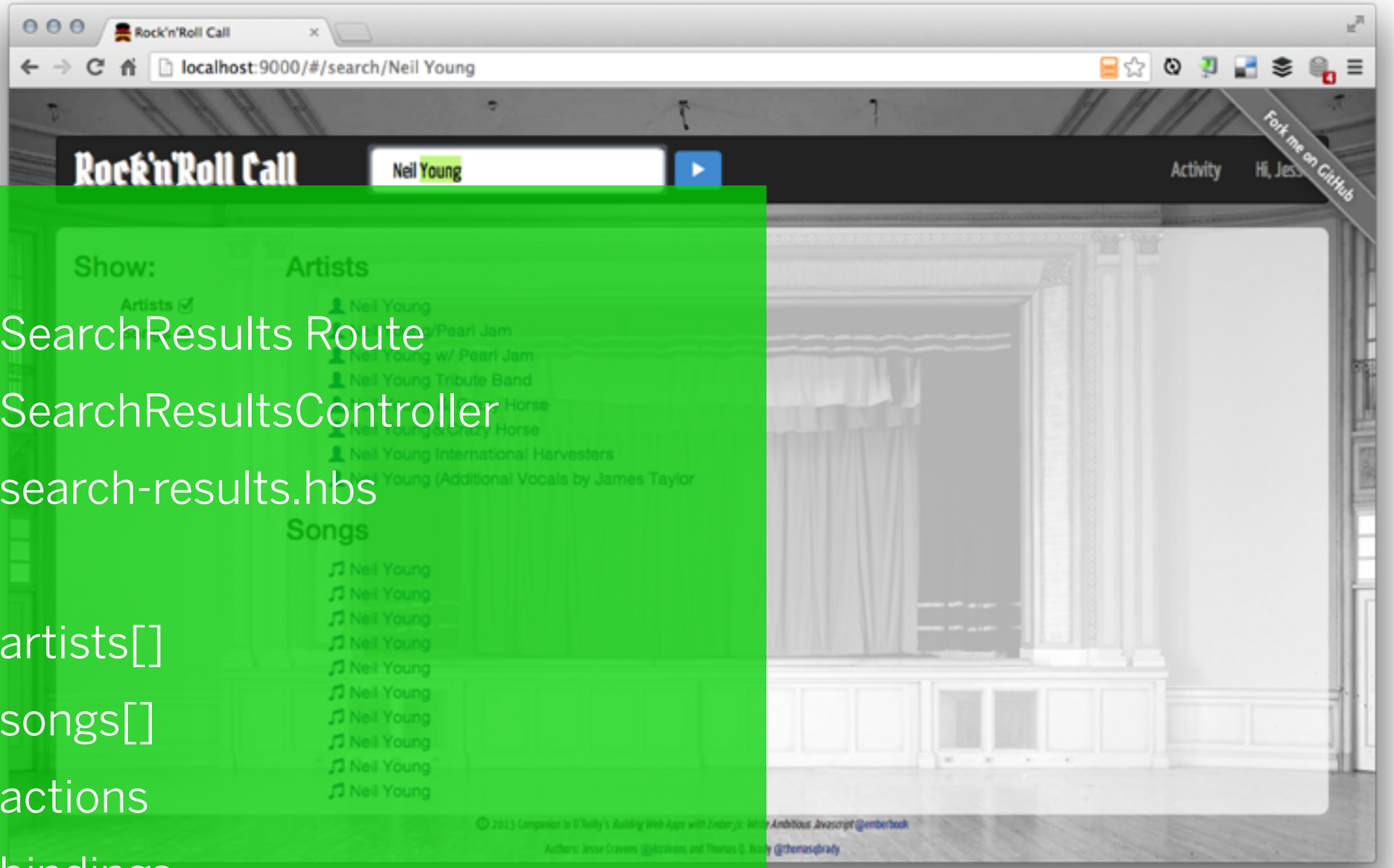
DESIGN IN .HBS

SEARCH RESULTS STATE



DESIGN IN .HBS

SEARCH RESULTS STATE



SearchResults Route

SearchResultsController

search-results.hbs

artists[]

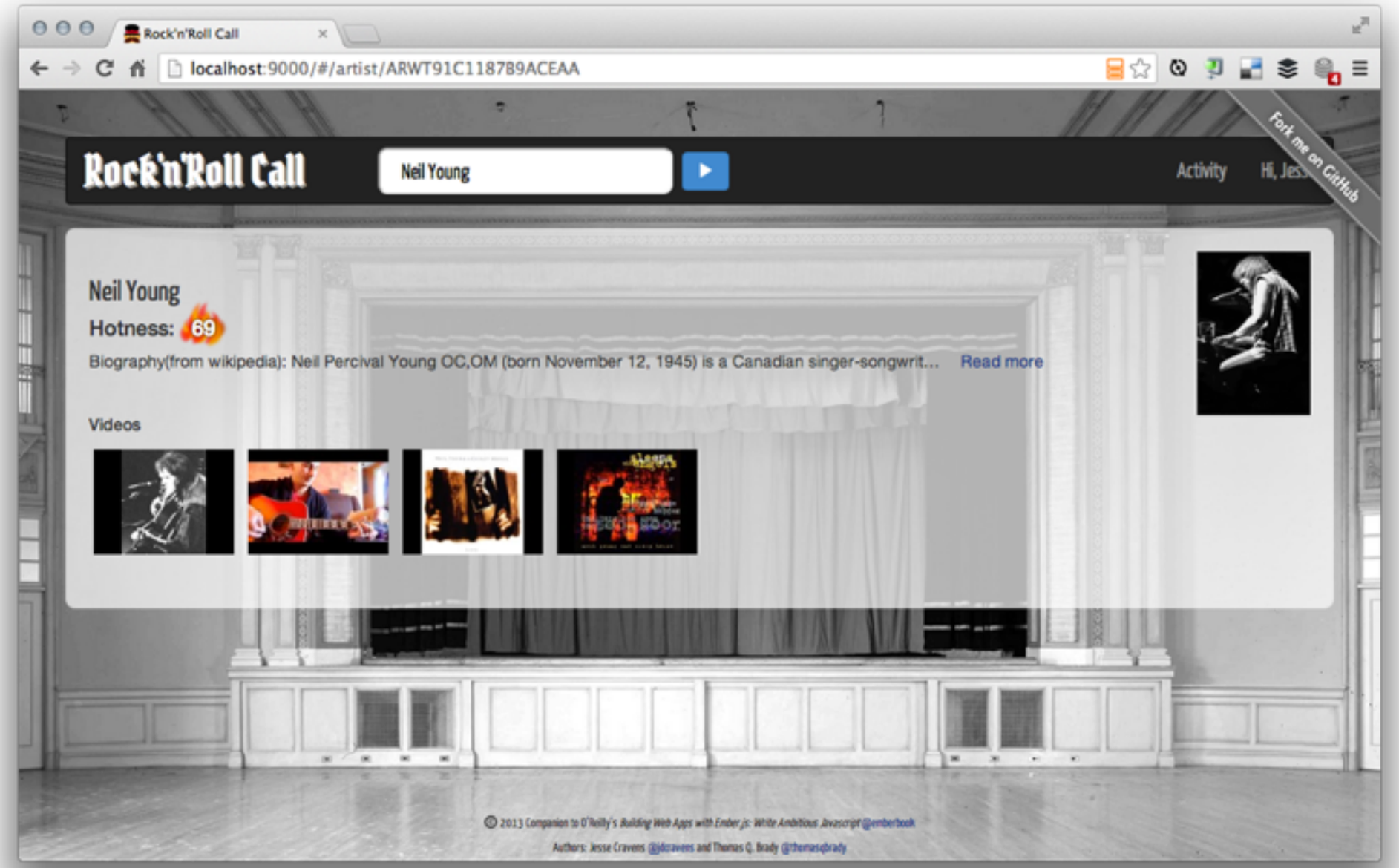
songs[]

actions

bindings

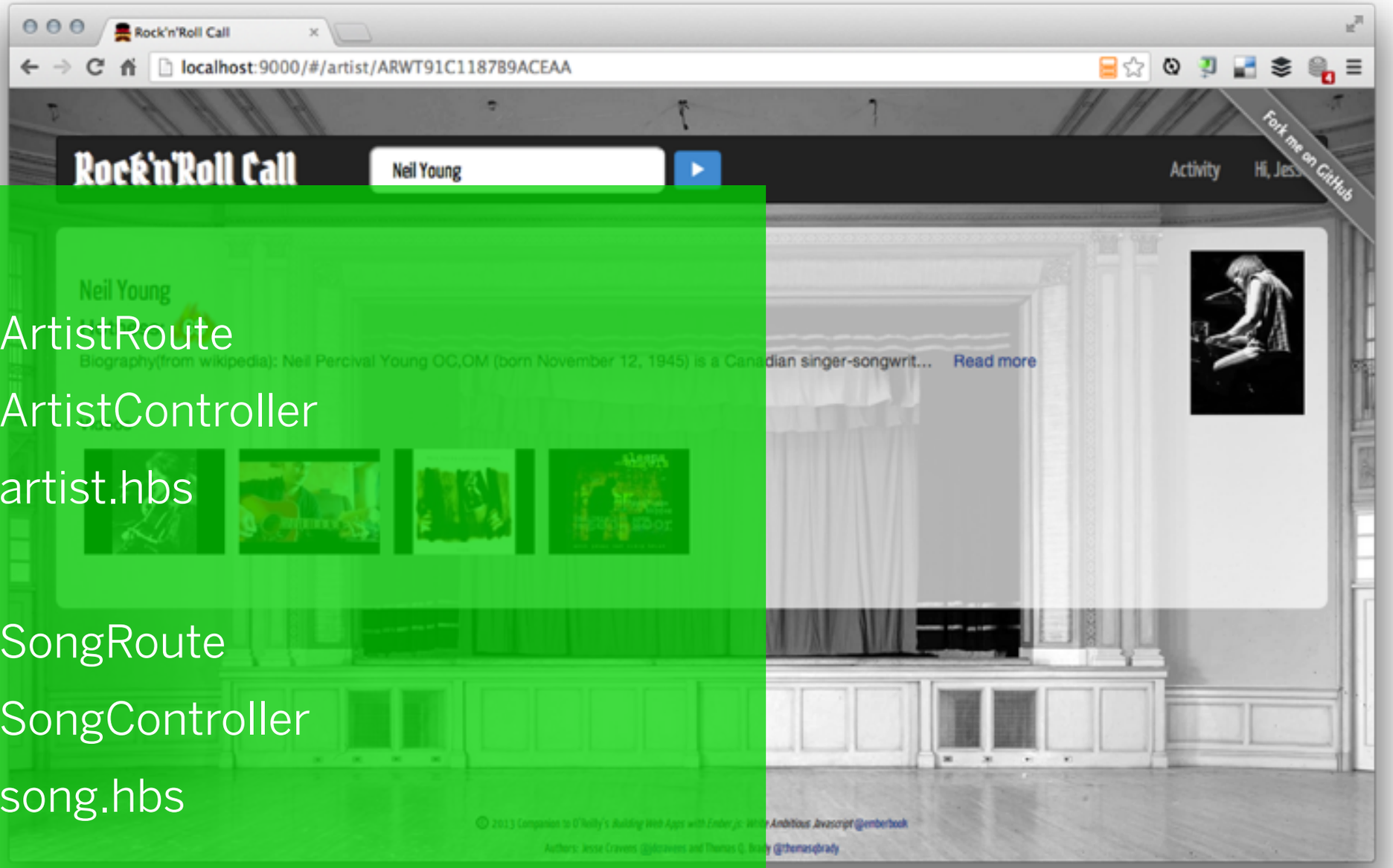
DESIGN IN .HBS

ARTIST/SONG DETAIL STATE



DESIGN IN .HBS

ARTIST/SONG DETAIL STATE



ArtistRoute

ArtistController

artist.hbs

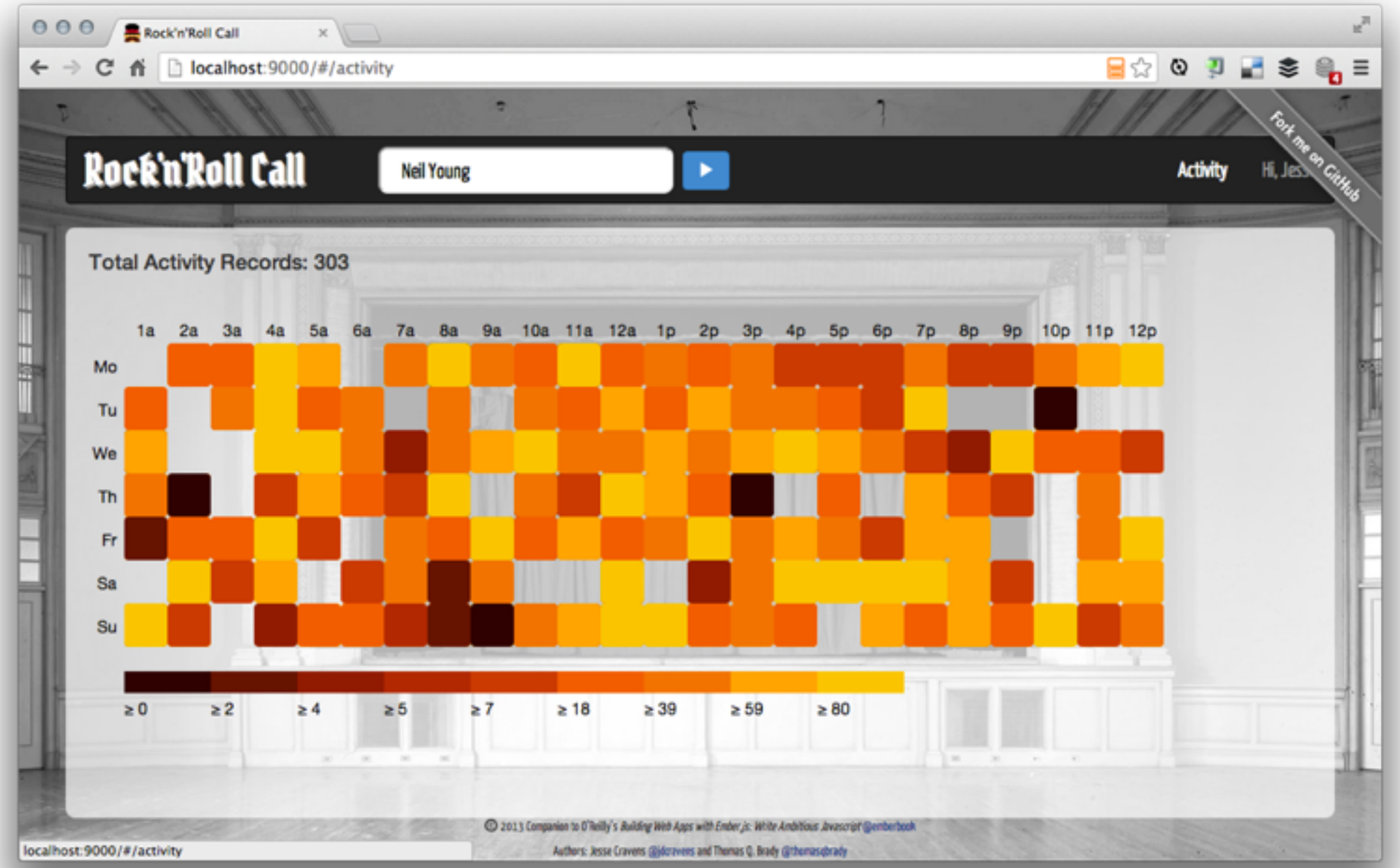
SongRoute

SongController

song.hbs

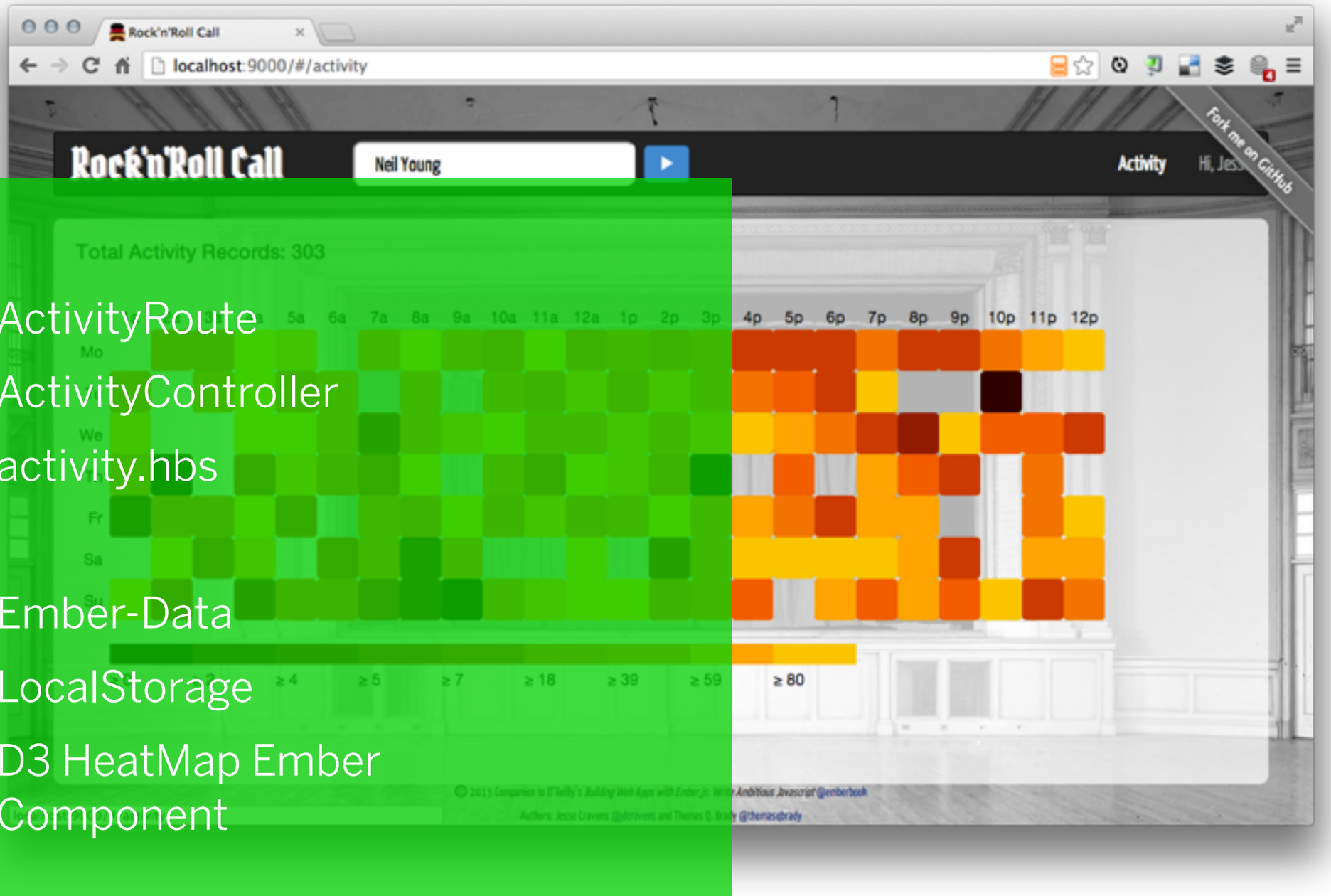
DESIGN IN .HBS

ACTIVITIES STATE



DESIGN IN .HBS

ACTIVITIES STATE



ActivityRoute

ActivityController

activity.hbs

Ember-Data

LocalStorage

D3 HeatMap Ember Component

HANDLEBARS BASICS

chapter 4

HANDLEBARS INLINE

<script type="text/x-handlebars" **data-template-name=“application"**>

  <!-- template code here -->

</script>

HANDLEBARS .HBS

```
grunt.initConfig({
    yeoman: yeomanConfig,
    watch: {
        emberTemplates: {
          files: '<%= yeoman.app %>/templates/**/*.hbs',
          tasks: ['emberTemplates', 'livereload']
        }
    }
});
```

HANDLEBARS VARIABLES

```
<a href="#" class="navbar-brand">

  {{App.applicationName}}

</a>
```

HANDLEBARS LINKS

{{**#linkTo** "index" class=“navbar-brand”}}

{{App.applicationName}}

{{**/linkTo**}}

HANDLEBARS LISTS

```
<ul class="search-results artists">
 {{#each artist in App.dummySearchResultsArtists}}

   {{#each nickname in artist.nicknames}}

     <li>
        <a href="#">
           {{nickname}}, AKA "{{artist.name}}"
        </a>
     </li>

   {{/each}}

 {{/each}}
</ul>
```

HANDLEBARS LOGIC

```
<ul class="search-results artists">
 {{#each artist in App.dummySearchResultsArtists}}

  {{#if artist.nicknames}}
   {{#each nickname in artist.nicknames}}
    <li>
      <a href="#">{{nickname}}"</a>
    </li>
   {{/each}}
  {{else}}
   <li><a href="#">{{artist.name}}</a></li>
  {{/if}}

 {{/each}}
</ul>
```

HBS BOUND ATTRIBUTES

```
<a {{bind-attr href=video.url}}>

    Watch now on Vimeo

</a>
```

HANDLEBARS HELPERS

```
Ember.Handlebars.helper('hotttnesss-icon',
function(value, options) {
  var h = parseFloat(value);
  var hotttnesss = Math.ceil(h * 10) - 1;
  var html = "<h4>Hotness: ";
  if (hotttnesss > 0) {
    html += '<i class="hotttnesss">';
          ...
    }
        ...
  } else {
    html += "0</h4>";
  }
  return new Handlebars.SafeString(html);
});
```

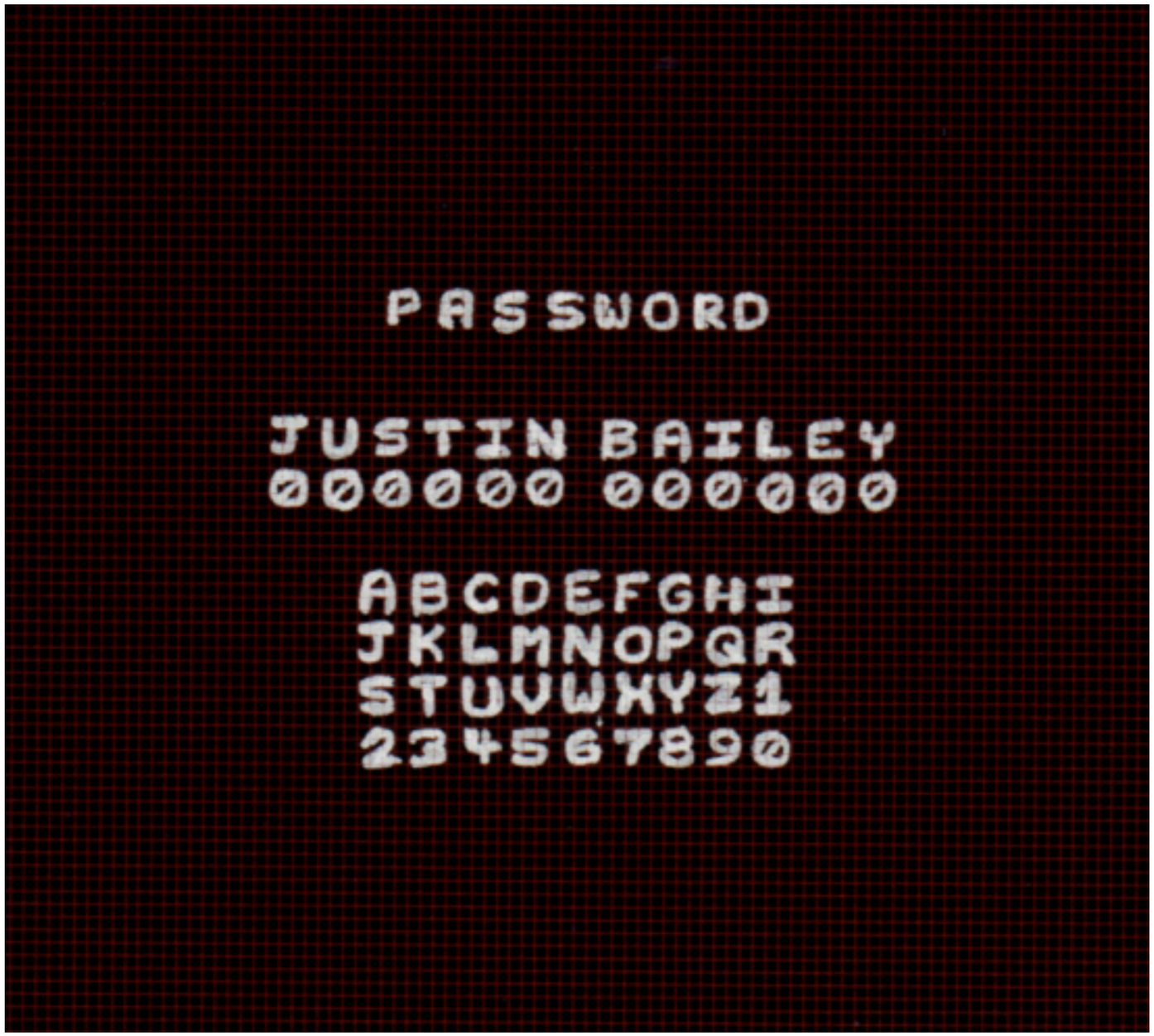
THE ROUTER

chapter 5

URL DRIVEN DESIGN

- Think ‘STATE’
- Router is a StateManager
- NO MODELS: return data in .model() hook
- NO VIEWS: templates first

SAVED GAME PASSWORDS



ROUTER

- Connect URLs to Code
- Manages ‘states’ of the Ember application

LOG TRANSITIONS

```
App = Ember.Application.create({

  LOG_TRANSITIONS: true

});
```

MAP

```
App.Router.map(function() {

    // routes

});
```

MAPPING ROUTES

```
App.Router.map(function() {

  this.route('search-results');
  this.route('artist');
  this.route('song');
  this.route('activity');

});
```

DYNAMIC ROUTES

```
App.Router.map(function() {
  this.route('search-results', {
    path: 'search/:term'
  });
  this.route('artist', {
    path: 'artist/:enid'
  });
  this.route('song', {
    path: 'song/:enid'
  });
});
```

ROUTE HANDLERS

```
App.ArtistRoute = Ember.Route.extend({

    // handle a transition to this Route

});
```

THE MODEL HOOK

```
App.ArtistRoute = Ember.Route.extend({
  model: function(params) {

  }
});
```

ROUTE HANDLERS

```
App.ArtistRoute = Ember.Route.extend({
 model: function(params) {

   XHR("some URL",{"id":params.enid},function
callback(response){
    var artist = App.Artist.create({
      name: response.name,
      hotttnesss: response.hotttnesss,
    });
  });

  }
});
```

EMBER OBJECT

```
App.Artist = Em.Object.extend({
  id: null,
  name: null,
  enid: null,
  biography: null,
  hotttnesss: null,
  image: null,
  videos: null
});
```

PROMISES

```
Promise.all([afunction(),another(), yetAnother()])

 .then(function() {
  console.log("They're all finished, success is ours!”);
}, function() {
  console.error("One or more FAILED!”);
});
```

ROUTE HANDLERS

```
App.ArtistRoute = Ember.Route.extend({

    // handle a transition to Artist Detail State
    // call Echo Nest API

});
```

```
App.SongRoute = Ember.Route.extend({

    // handle a transition to Song Detail State
    // call Echo Nest API

});
```

LOADING ROUTE

App.LoadingRoute = Ember.Route.extend({});

```
<div id=“lp"></div>
<audio
    src="sounds/tuning.mp3"
    autoplay=“autoplay">
</audio>
```

CONTROLLERS

chapter 6

CONTROLLERS

- Manipulate the data within the application's Models
- Store transient data, whether stand-alone or made up of data retrieved from Models
- Listen to events dispatched by and dispatch events intended for other Controllers, Views and Templates
- Instigate the transition from one route to another

APPLICATION CONTROLLER

```
App.ApplicationController =
Em.ObjectController.extend({

  searchTerms: '',

  applicationName: "Rock'n'Roll Call”,

  actions: {
    submit: function() {
      this.transitionToRoute('search-
results',this.get('searchTerms'));
    }
  }

});
```

VALUE BINDING

```
{{view Ember.TextField
    class="search-input"
    placeholder="Search for artists or song names"
valueBinding="controller.searchTerms"}}

 <button class="btn btn-primary">
    <i class="glyphicon glyphicon-play"></i>
</button>
```

.GET() AND .SET()

```
App.ApplicationController =
Em.ObjectController.extend({

  searchTerms: '',

  applicationName: function() {
    var st = this.get('searchTerms');
    if (st) {
      return st + "???"
    } else {
      return "Rock'n'Roll Call"
    }
  }.property('searchTerms'),
  ...
});
```

COMPUTED PROPERTIES

```
App.ApplicationController =
Em.ObjectController.extend({
  ...
  applicationName: function() {
    var st = this.get('searchTerms');
    if (st) {
      return st + "???"
    } else {
      return "Rock'n'Roll Call"
    }
  }.property('searchTerms'),
  ...
});
```

ACTIONS

```
App.ApplicationController =
Em.ObjectController.extend({

  ...

  actions: {

    ...

  }

});
```

SUBMIT

```
{{view Ember.TextField
    class="search-input"
    placeholder="Search for artists or song names”
    action=“submit"
valueBinding=“controller.searchTerms"}}
```

```
<button {{action "submit"}} class="btn btn-primary”>
    <i class="glyphicon glyphicon-play”></i>
</button>
```

TRANSITIONTOROUTE

```
App.ApplicationController =
Em.ObjectController.extend({

  ...

  actions: {
    submit: function() {
      this.transitionToRoute('search-
results',this.get('searchTerms'));
    }
  }

});
```

CHECKED BINDING

```
App.SearchResultsController =
Em.ObjectController.extend({
  ...
    artistsIsChecked: true,
    songsIsChecked: true
  ...
});

{{view Ember.Checkbox
checkedBinding="artistsIsChecked"}}

{{#if artistsIsChecked}}
    ...
{{/if}}
```

EMBER-DATA

chapter 7

ACTIVITY TEMPLATE

```
<h4>Total Activity Records: {{model.length}}</h4>

<ul>
 {{#each model}}
 <li class="activity">{{this.id}}</li>
  {{/each}}
</ul>
```

ACTIVITY ROUTE

```
App.Router.map(function() {
  ...
  this.route('activity');
});

App.ActivityRoute = Ember.Route.extend({
  model: function () {
    return this.store.find(‘activity');
  }
});
```

ACTIVITY MODEL

```
App.Activity = DS.Model.extend({
  display_id: DS.attr('string'),
  type: DS.attr('string'),
  display_name: DS.attr('string'),
  hotttnesss: DS.attr('number'),
  timestamp: DS.attr()
});
```

ACTIONS

```
{{#if artistsIsChecked}}
 {{#if artists.length}}
  <h3>Artists</h3>
  <ul class="search-results artists">
   {{#each artists}}
    <li>
     <a {{action 'viewedArtist' this }}>{{name}}</a>
    </li>
   {{/each}}
  </ul>
 {{/if}}
{{/if}}
```

CREATERECORD()

```
actions: {
   viewedSong: function(model) {

      var date = Date.now();
      var activity = this.store.createRecord('activity', {
         display_id: model.sid,
         type: model.type,
         display_name: model.artist_name,
         hotttnesss: model.hotttnesss,
         timestamp: date
      });
      ...
   }
},
```

SAVE()

```
actions: {
   viewedSong: function(model) {
      ...
      var activity = this.store.createRecord('activity', {
         display_id: model.sid,
         type: model.type,
         display_name: model.artist_name,
         hotttnesss: model.hotttnesss,
         timestamp: date
      });
      activity.save();
   }
},
```

FILTER()

```
store.filter('activity', function(activity){
  return activity.get('type', 'song');
});
```

ALL()

store.all('activity');

ADAPTERS

```
App.ApplicationAdapter = DS.FixtureAdapter.extend({
  namespace: 'rocknrollcall'
});

App.ApplicationAdapter = DS.LSAdapter.extend({
  namespace: 'rocknrollcall'
});

App.ActivityAdapter = DS.LSAdapter.extend({
  namespace: 'rocknrollcall'
});

App.ApplicationAdapter = DS.RESTAdapter.extend({
  namespace: 'rocknrollcall'
});
```

APP INITIALIZER / SEEDS

```
Ember.Application.initializer({
  name: "DBseeds",
  initialize: function(container, application) {

    localStorage.clear();
...
```

BUILDING A BACKEND

chapter 8

EMBER APP KIT

```
{
  "name": "app-kit",
  "namespace": "appkit",
  "APIMethod": "stub",
...
```

ROUTES.JS

```
module.exports = function(server) {

    // Create an API namespace, so that the root does not
     // have to be repeated for each end point.
     server.namespace("/api", function() {

         // Return fixture data for "/api/activities"
         server.get("/activities", function(req, res) {
             var activities = [ ];
             };
             res.send(activities);
         });
     });
};
```

RESTADAPTER

```
export default DS.RESTAdapter.extend({
  namespace: 'api'
});
```

EMBER APP KIT

```
{
  "name": "app-kit",
  "namespace": "appkit",
  "APIMethod": "stub",
...

{
  "name": "app-kit",
  "namespace": "appkit",
  "APIMethod": “proxy”,
  "proxyURL": "http://whatever.api:3232",
...
```

RAILS AND EMBER

- Rails helps you manage your Ruby code, server side MVC.
- Active Record provides an ORM for interacting with your data.
- Rails Asset Pipeline, and the ember-rails gem, help you manage your assets.
- Active Model Serializers provide the REST API that the Ember Data RESTAdapter expects by default.
- Testing is built-in.

ACTIVE:MODEL SERIALIZERS

```
class ActivitySerializer < ActiveModel::Serializer

  # look up 'typ' on the model, but use 'type' in the JSON
  def type
    object.typ
  end

  attributes :id, :display_id, :type, :display_name, :timestamp

end
```

COMPONENTS

chapter 9

WEB COMPONENTS

- HTML Imports .... Em.TEMPLATES
- Shadow DOM
- Model Driven Views
- Custom Elements

- For more on Web Components: http://jessecravens.com/blog/2013/07/25/building-next-generation-widgets-with-web-components/

ANATOMY

ACTIVITY LOG

- File System: templates/components/activity-log.hbs
- Custom Elements naming convention: {{activity-log}}

HEAT MAP

- File System: templates/components/heat-map.hbs
- Custom Elements naming convention: {{heat-map}}

REUSE ACTIVITY-LOG.HBS

```
<p> display_id: {{display_id}}</p>
<p> type: {{type}}</p>
<p> display_name: {{display_name}}</p>
<p> hotttnesss: {{hotttnesss}}</p>
<p> timestamp: {{timestamp}}</p>
<hr>
```

And in activities.hbs:

```
{{#each}}
  {{activity-log }}
{{/each}}
```

SCOPE ISSUE

{{#each}}

{{activity-log **display_id=display_id type=type display_name=display_name hotttnesss=hotttnesss timestamp=timestamp** }}

{{/each}}

SUBCLASS

App.HeatMapComponent =
Ember.Component.extend({});

DIDINSERTELEMENT()

```
App.HeatMapComponent = Ember.Component.extend({

 didInsertElement: function(){
   var data = this.get('controller.data.content');
   this.draw(data);
 }

});
```

3RD PARTY PATTERN: D3.JS

```
App.HeatMapComponent = Ember.Component.extend({

  width: 900,
  height: 280,

  draw: function(myData){
    // draw the heat map
  },

  didInsertElement: function(){
    var data = this.get('controller.data.content');
    this.draw(data);
  }
});
```

TESTING
chapter 10

TESTING SETUP - EAK

```
document.write('<div id="ember-testing-
container"><div id="ember-testing"></div></div>');

Ember.testing = true;

window.startApp = require('appkit/tests/helpers/
start_app')['default'];
window.isolatedContainer = require('appkit/tests/
helpers/isolated_container')['default'];

... // test-helpers

window.exists = exists;
window.equal = equal;
window.strictEqual = strictEqual;
```

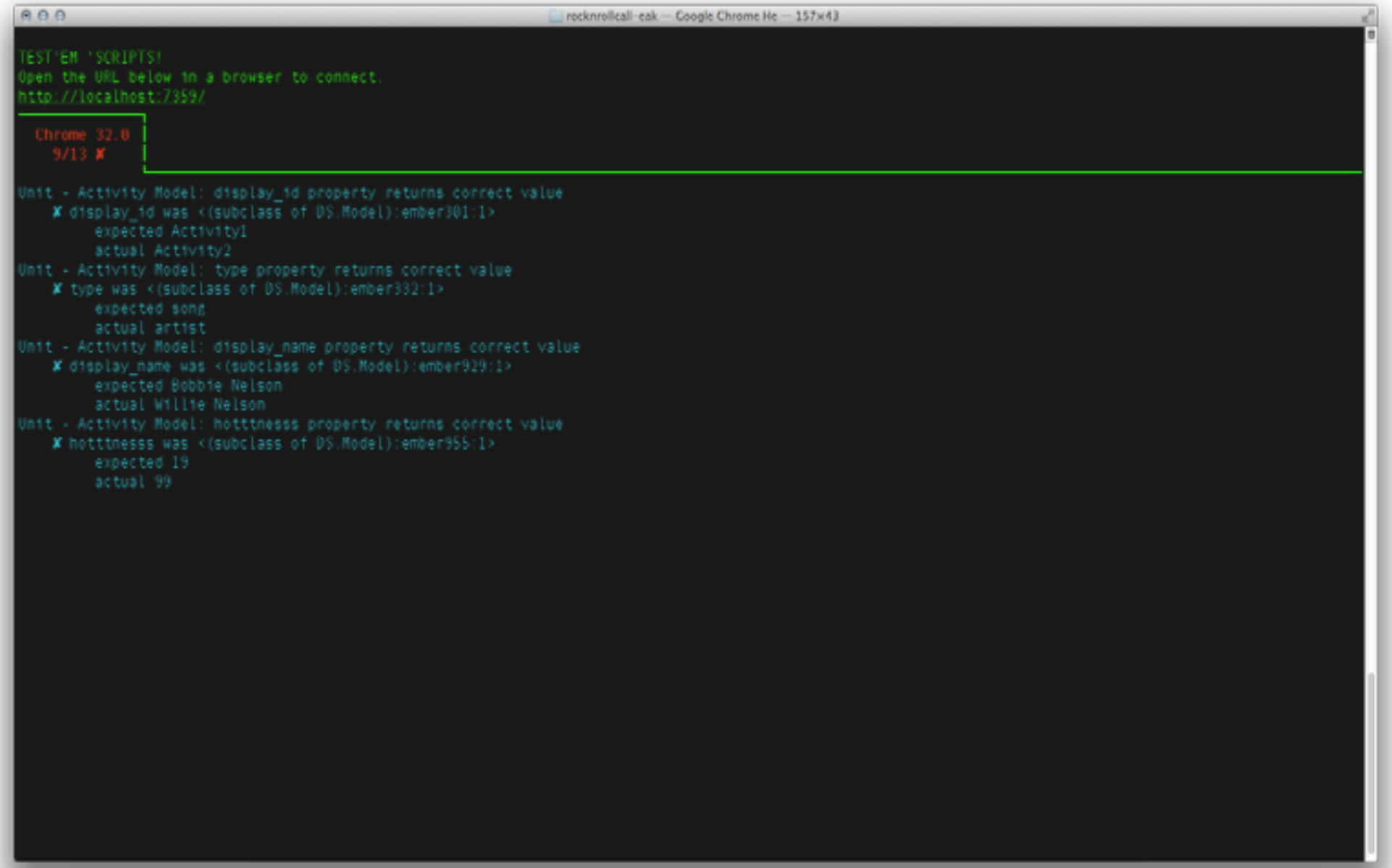
STARTAPP()

```
import Application from 'appkit/app';

function startApp(attrs) {
  var App;
    ...
  Ember.run(function(){
    App = Application.create(attributes);
    App.setupForTesting();
    App.injectTestHelpers();
  });

  return App;
}

export default startApp;
```

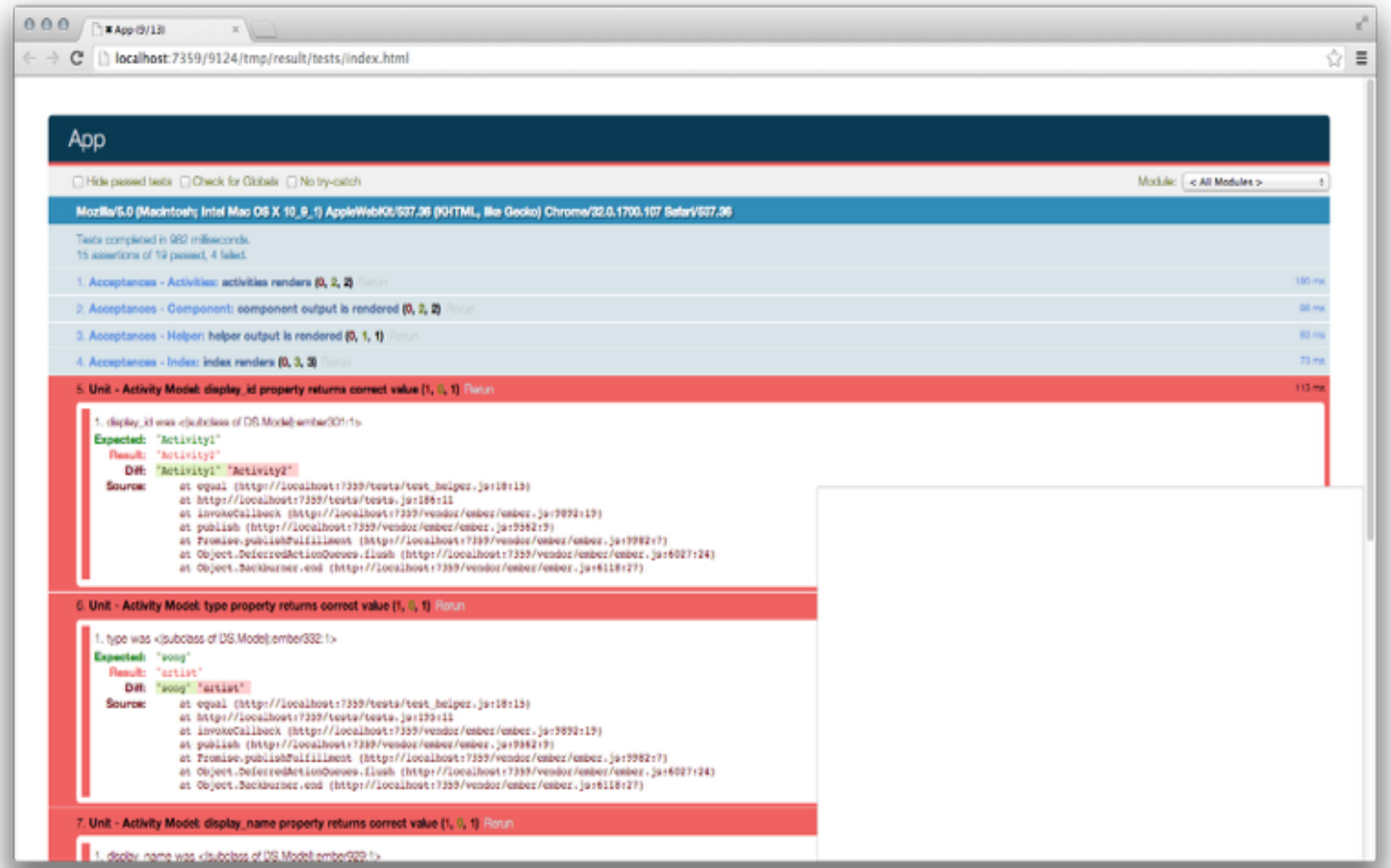
HELPERS

```
function exists(selector) {
  return !!find(selector).length;
}

function equal(actual, expected, message) {
  message = getAssertionMessage(actual, expected,
message);
  QUnit.equal.call(this, actual, expected, message);
}

function strictEqual(actual, expected, message) {
  message = getAssertionMessage(actual, expected,
message);
  QUnit.strictEqual.call(this, actual, expected, message);
}
```

TESTEM TEST RUNNER



QUNIT TEST RUNNER



INTEGRATION MODULES

```
var App;

module('Acceptances - Activities', {
  setup: function(){
    App = startApp();
  },
  teardown: function() {
    Ember.run(App, 'destroy');
  }
});

    // tests

});
```

INTEGRATION TESTS

```
module('Acceptances - Activities', {
  ...
});

test('activities renders', function(){
  visit('/activities').then(function(){
    var title = find('h4');
    var list = find('ul li.activity');

    equal(title.text(), 'Total Activity Records: 2');
    equal(list.length, 2);
  });
});
```

TESTING ROUTES

```
export default Ember.Route.extend({
  model: function() {
    return ['red', 'yellow', 'blue'];
  }
});
```

```
export default Ember.Route.extend({
  model: function() {
    return this.get('store').find('activity');
  }
});
```

UNIT TESTING ROUTES

```
import Activities from 'appkit/routes/activities';

var route;

module('Unit - ActivitiesRoute', {
  setup: function() {
    var container = isolatedContainer([
      'route:activities'
    ]);
    route = container.lookup('route:activities');
  }
});
```

ISOLATED CONTAINER

```
import Resolver from 'resolver';

function isolatedContainer(fullNames) {
  var container = new Ember.Container();
     ...
  var resolver = Resolver['default'].create();

  for (var i = fullNames.length; i > 0; i--) {
    var fullName = fullNames[i - 1];
    container.register(fullName,
resolver.resolve(fullName));
  }
  return container;
}
export default isolatedContainer;
```

TESTING ROUTES

```
test('route exists', function() {
  ok(route);
  ok(route instanceof Activities);
});
```

TESTING ROUTES MODEL()

```
test('#model', function() {

  var store = {
    find: function() {
      return [{...}];
    }
  };

  route.set('store', store);

  deepEqual(route.model(), [{ ... }]);

});
```

USING FIXTURES

```
var Activity = DS.Model.extend({
  display_id: DS.attr('string'),
     ...
});

Activity.FIXTURES = [
  {...},
  {...}
];

export default Activity;
```

USING FIXTURES

```
import Activities from 'appkit/routes/activities';
import Activity from 'appkit/models/activity';

test('#model', function() {

  var store = {
    find: function() {
      return Activity.FIXTURES;
    }
  };
  route.set('store', store);

  deepEqual(route.model(), Activity.FIXTURES);

});
```

TESTING PROMISES

```
module("Unit - ActivitiesRoute", {
  setup: function() {

    var store = {
      find: function(type) {
        return new Em.RSVP.Promise(function(resolve) {
          resolve(Activity.FIXTURES);
        });
      }
    };
    route.set('store', store);
  }
});
```

ASYNCTEST, EM.RUN()

```
asyncTest('#model', function(){

  Em.run(function(){
    route.model().then(function(result){
      equal(result, Activity.FIXTURES);
      start();
    });

  });
});
```

TESTING MODELS

```
module('Unit - Activity Model', {
  setup: function() {
    App = startApp();
    store = App.__container__.lookup('store:main');
  },
  teardown: function() {
    Ember.run(App, 'destroy');
  }
});
```

TESTING MODELS

```
asyncTest('display_id property returns correct value',
function() {

  Ember.run(function () {
    store.find('activity', 1).then(function(result){

      equal(result.get('display_id'), 'Activity2', 'display_id
was ' + result);

      start();
    });
  });
});
```

GRACIAS Y ADIOS.