with Play Scala, CoffeeScript and Jade

Matt Raible
http://raibledesigns.com

Photos by **Trish** - **http://mcginityphoto.com**

Introductions

- Have you used HTML5?
- Have you used Play Framework?
- Have you tried Scala?
- Tried CoffeeScript?
- Scalate or Jade?

© 2012 Raible Designs

Who is **Matt Raible**?

Father, Skier, Cyclist

Web Framework Connoisseur

Founder of AppFuse

Blogger on raibledesigns.com

Agenda

- Introductions
- Why am I doing this talk?
- What are these technologies?
- My Development Experience
- Demo
- Q and A
- Conclusion

© 2012 Raible Designs

Why am I doing this talk?

Why ÜberConf?

Why am I doing this talk?

© 2012 Raible Designs

Why am I doing this talk?

- I like a Challenge to...
  - Learn Scala
  - Via Play!
  - And Jade is cool too!
  - So is CoffeeScript!

© 2012 Raible Designs

Why am I doing this talk?

- Honestly, it's because of James Strachan...

© 2012 Raible Designs

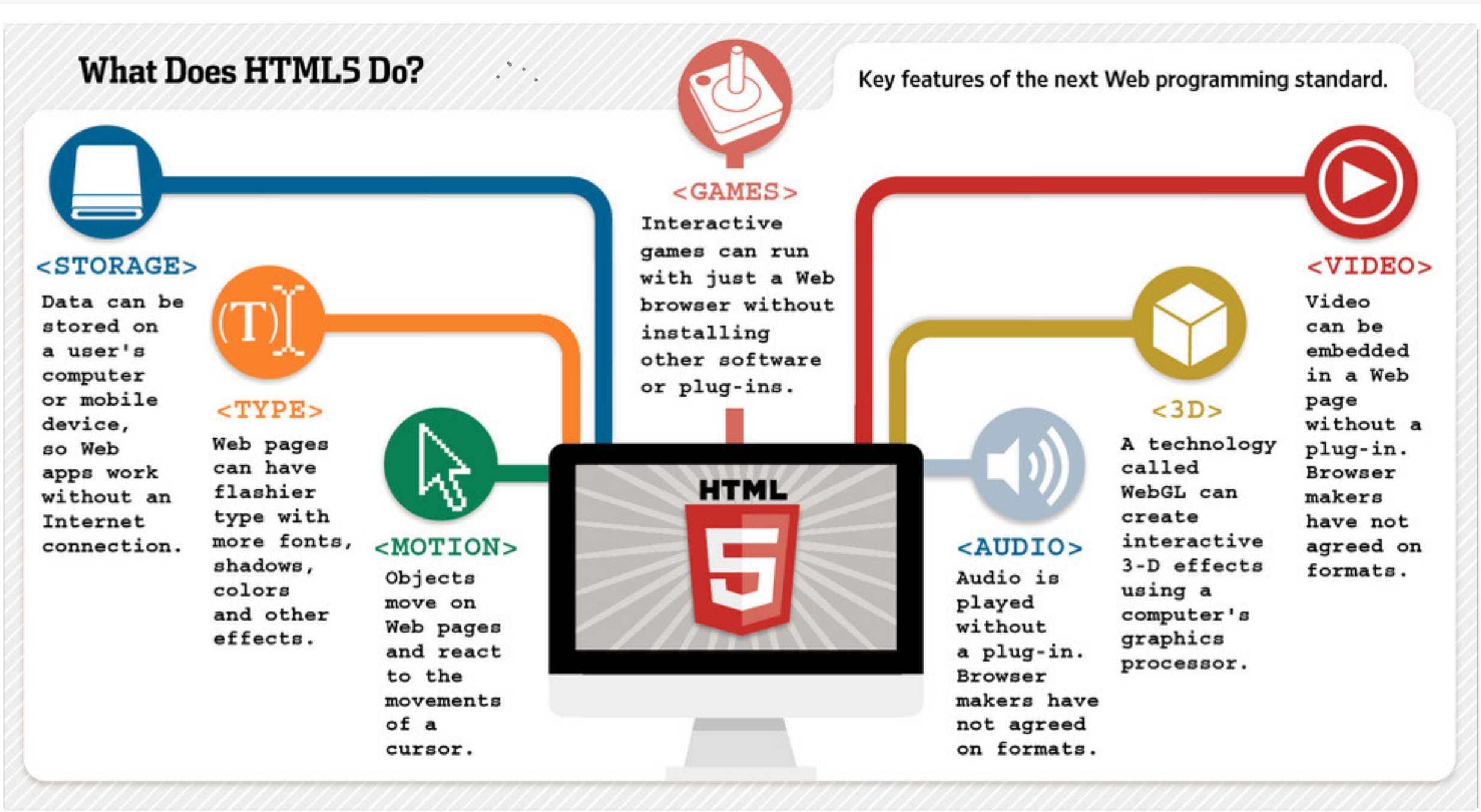
Why am I doing this talk?

© 2012 Raible Designs

What are these technologies?

© 2012 Raible Designs

HTML5



http://on.wsj.com/tEGIJL

© 2012 Raible Designs

How do you write HTML5?

<!DOCTYPE html>

<article> <aside> <section>
<header> <footer> <nav>
<audio> <canvas> <video>
<datalist> <details>

<applet> <font>
<frame> <frameset>

http://www.w3schools.com/html5/html5_reference.asp

© 2012 Raible Designs

CSS3

- Animated Transitions

```
transform: rotateY(180deg);
```

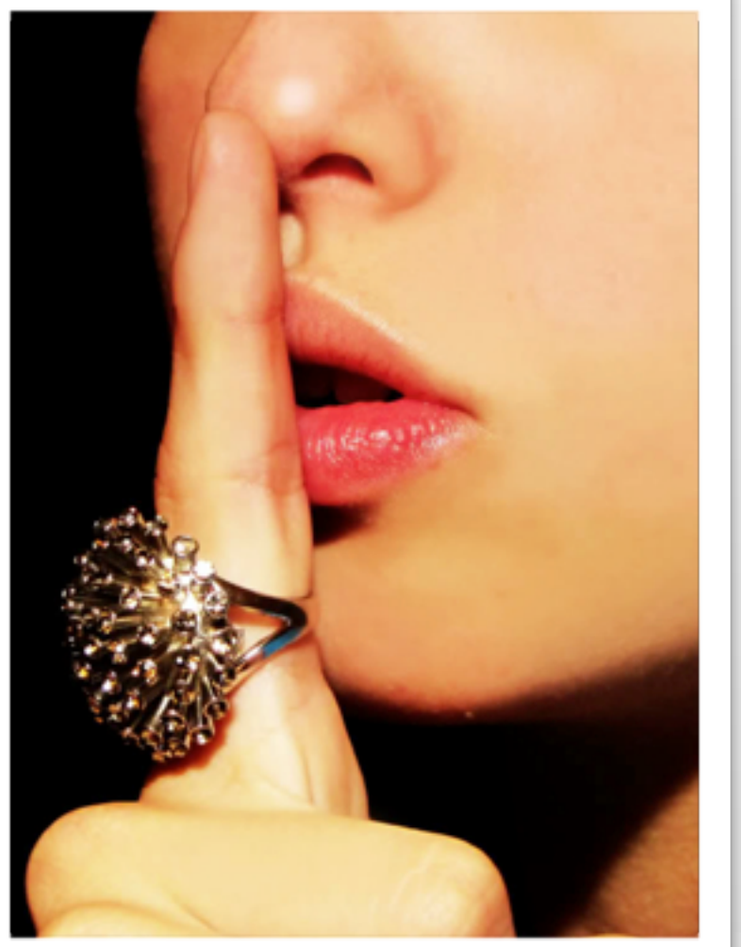
- Rounded Corners

```
border-radius: 8px 8px 0 0;
```

- Drop Shadows

```
box-shadow: 2px 2px 4px 4px;
```

- Gradient Colors
- Styling based on sibling count
- More cursors for better usability
- Custom Checkboxes and Radio Buttons



http://lea.verou.me/css3-secrets

© 2012 Raible Designs

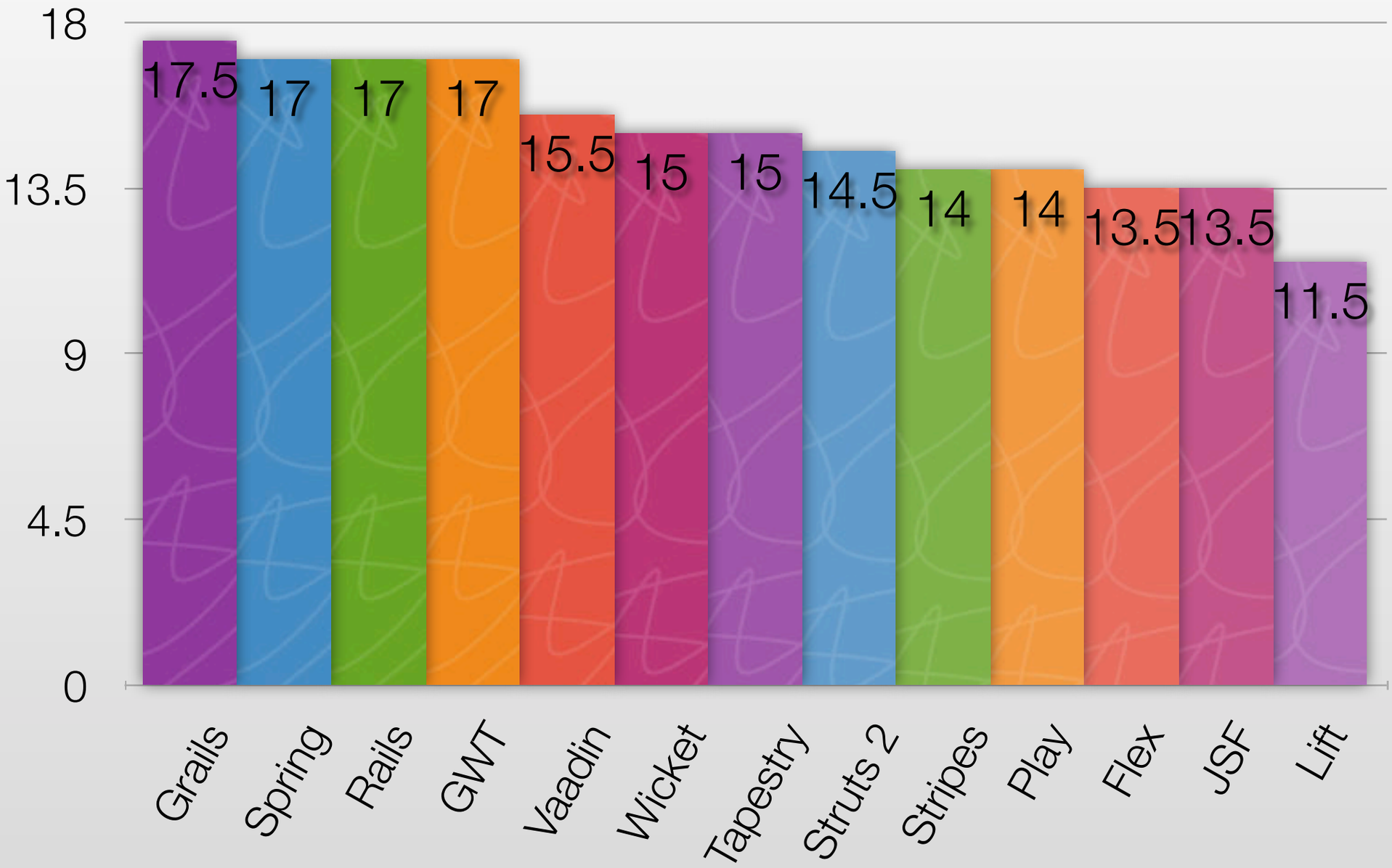
Play Framework

- A full-stack Java Web Framework made by Web Developers
- Compile on-the-fly

- Stateless Architecture
- Breaks web framework norms
  - Uses static methods
  - Doesn't use Servlet API

© 2012 Raible Designs

Matrix Results



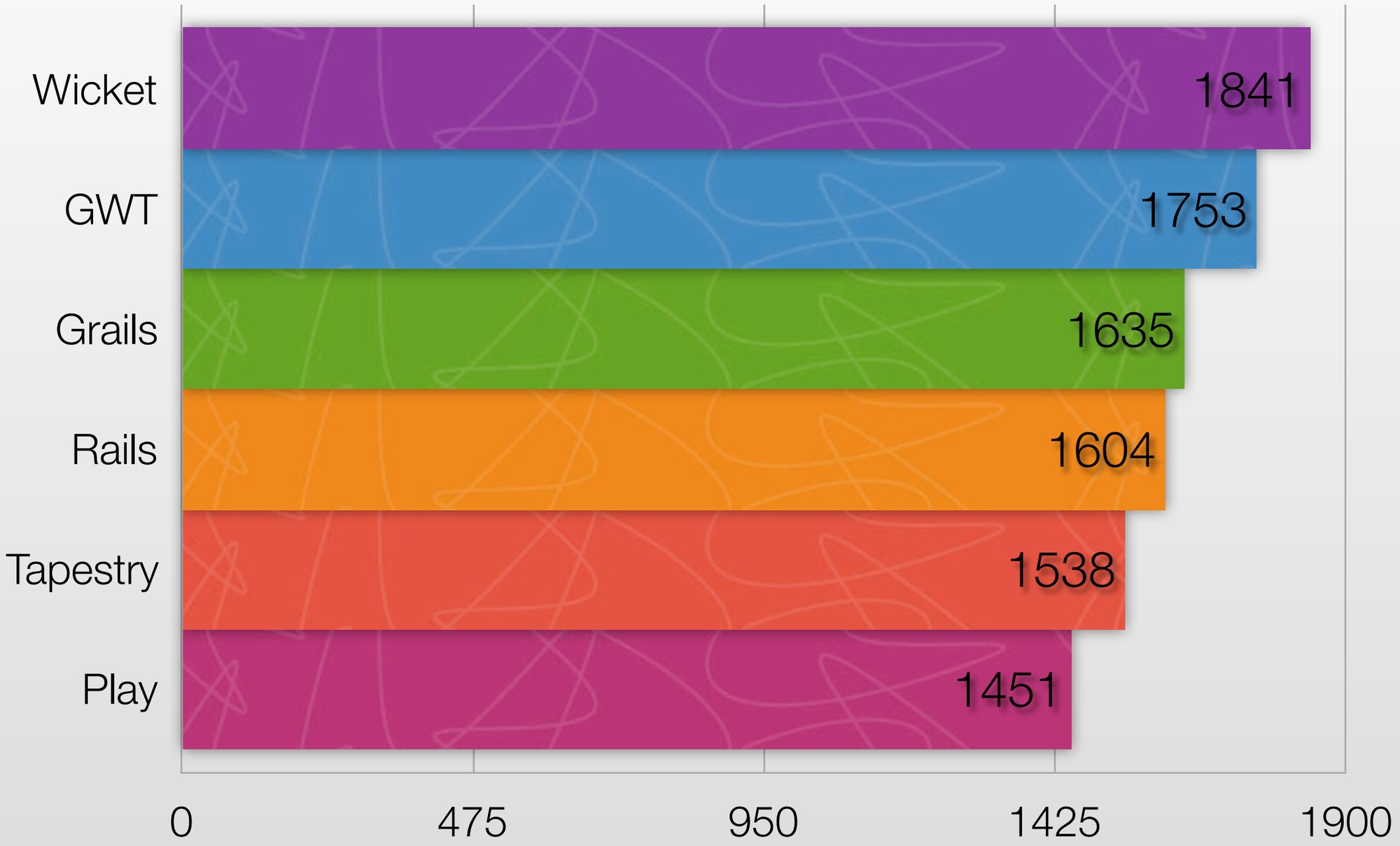
© 2012 Raible Designs

Weighted Results

- Grails (90)
- Spring MVC (85)
- Ruby on Rails (82.5)
- Vaadin (82.5)
- Play (82.5)
- GWT (80)

© 2012 Raible Designs

Mailing List Traffic



July 2011

* Spring MVC and Vaadin use Forums, which don't provide this data.

© 2012 Raible Designs

Play Scala

```
$ play install scala
$ play new myScalaWebapp --with scala
$ play run
```

But really, it's more like this

© 2012 Raible Designs

Play Scala

Scala templates

A type safe — Scala based, template engine, optimized around HTML generation using a code-focused templating approach.

```
<h1>Product: @proc

<ul>
  @product.versions
    <li>Get @versi
  }
</ul>
```

Scala flavored Play API

Use the "full stack" Play API, enabled for the expressivity and conciseness of Scala language.

```
def show(id:
  Order.find
    html.inc
  ).getOrEls
}
```

Powerful SQL databases access

Anorm is simplification of JDBC with a minimal interface reusing pre-existing Scala interfaces (collections, pattern-matching, parsers combinators).

```
val postsWithAuthor:List[(Post~User)] =
  SQL(
    """
      select * from Post p join User u on
      p.author_id = u.id order by p.postedAt desc
    """
  ).as( Post ~< User * )
```

© 2012 Raible Designs

Scala

"Scala is like the dragon in Avatar. It will try to kill you, but if you master it, you can fly great distances with it and have a wonderful time."

-- Venkat Subramaniam

© 2012 Raible Designs

Scala Basics

- `def` starts a method
- variables are started with `var` or `val`
- variables are defined with name:type
- semicolons are not required

```
import play.mvc.Http

trait Scalate {

  def render(args: (Symbol, Any)*) = {
    val template = Http.Request.current().action.replace(".", "/")
    ScalateTemplate(template).render(args: _*);
  }
}
```

© 2012 Raible Designs

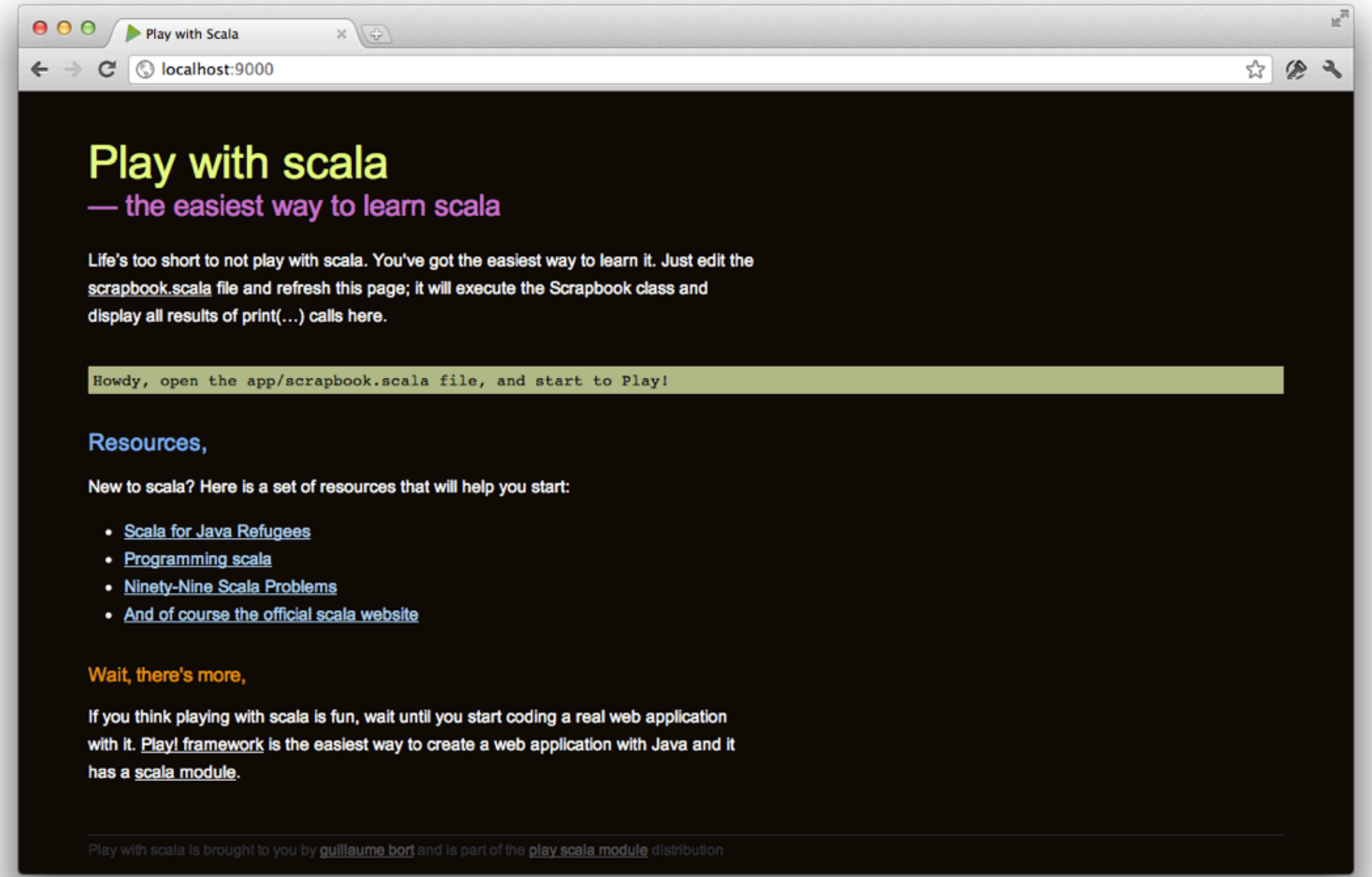
Scala vs. Java

```java
public class Car {
  private final int year;
  private int miles;

  public int getYear() { return year; }
  public int getMiles() { return miles; }
  public void setMiles(int theMiles) { miles = theMiles; }

  public Car(int theYear, int theMiles) {
    year = theYear;
    miles = theMiles;
  }
}
```

```scala
class Car(val year : Int, var miles : Int)
```

© 2012 Raible Designs

Play with Scala

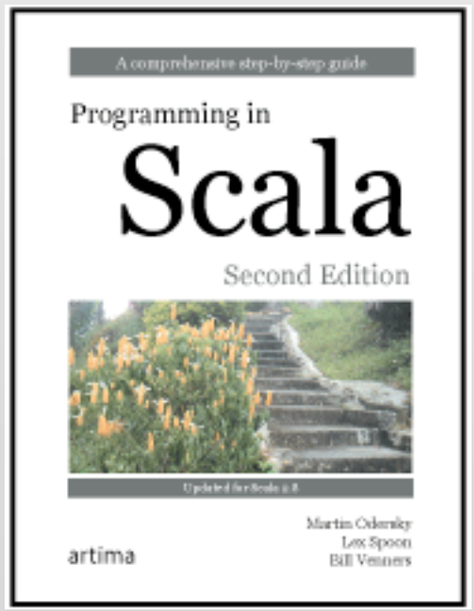


© 2012 Raible Designs

Learning Scala

- Venkat's **Scala for the Intrigued**
  - PragPub Magazine, starting in Sep 2011

- **Scala for the Impatient** - Cay Horstmann
- **Programming in Scala, 2nd Edition** - Martin Odersky, Lex Spoon, and Bill Venners
- Twitter's **Scala School**



© 2012 Raible Designs

CoffeeScript

JS → Coffee | Coffee → JS

Fork me on GitHub

```
/* Type here! */

(function ($) {
    $.fn.highlight = function () {
        $(this).css({ color: 'red', background: 'yellow' });
        $(this).fadeIn();
    };
})(jQuery);
```

```
(($) ->
  $.fn.highlight = ->
    $(this).css
      color: "red"
      background: "yellow"

    $(this).fadeIn()
) jQuery
```

MORE INFO ↓

© 2012 Raible Designs

JavaScript: The Good Parts

```coffeescript
square = (x) -> x * x
cube   = (x) -> square(x) * x
```

```javascript
var cube, square;

square = function(x) {
  return x * x;
};

cube = function(x) {
  return square(x) * x;
};
```

```coffeescript
fill = (container, liquid = "coffee") ->
  "Filling the #{container} with #{liquid}..."
```

```javascript
var fill;

fill = function(container, liquid) {
  if (liquid == null) liquid = "coffee";
  return "Filling the " + container + " with " + liquid +
"...";
};
```

```coffeescript
outer = 1
changeNumbers = ->
  inner = -1
  outer = 10
inner = changeNumbers()
```

```javascript
var changeNumbers, inner, outer;

outer = 1;

changeNumbers = function() {
  var inner;
  inner = -1;
  return outer = 10;
};

inner = changeNumbers();
```

© 2012 Raible Designs

Jade

jade

Node Template Engine

```
!!! 5
html(lang="en")
  head
    title= pageTitle
    script(type='text/javascript')
      if (foo) {
         bar()
      }
  body
    h1 Jade - node template engine
    #container
      - if (youAreUsingJade)
        p You are amazing
      - else
        p Get on it!
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Jade</title>
    <script type="text/javascript">
      if (foo) {
        bar()
      }
    </script>
  </head>
  <body>
    <h1>Jade - node template engine</h1>
    <div id="container">
      <p>You are amazing</p>
    </div>
  </body>
</html>
```

© 2012 Raible Designs

Jade Example

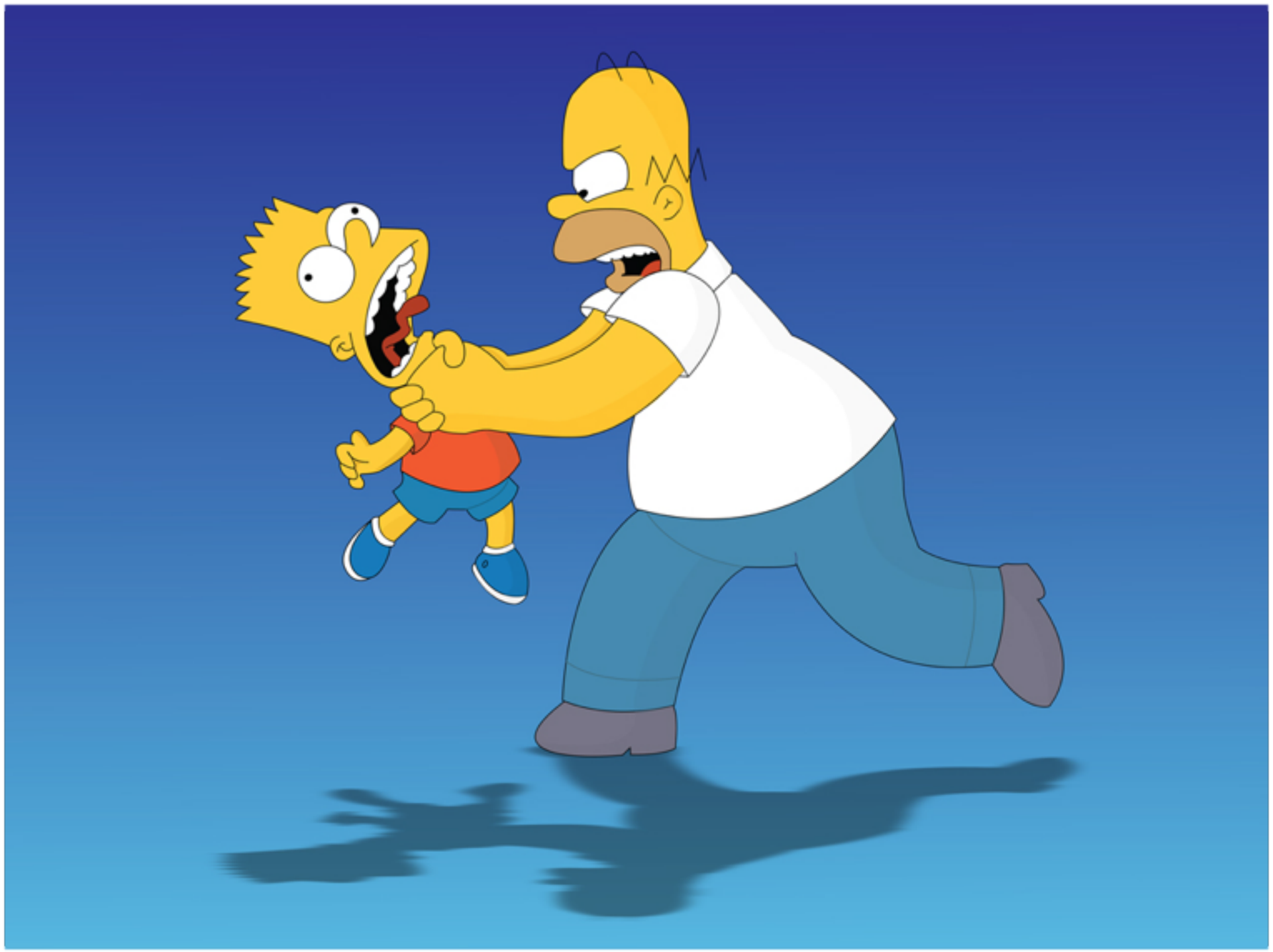
```
#display
  input(id="clock" class="xlarge" type="text" value="00:00:00.0" readonly="readonly")
#controls
  button(id="start" type="button" class="btn primary" disabled) Start
  button(id="reset" type="button" class="btn :disabled" disabled) Reset
  #options
    input#no-music(type="checkbox") No Music Please
#dashboard(style="display: none")
  #track
    | Distance Traveled: <span id="distance">0</span> mile(s)
    #actions
      button(id="save" type="sub
#where
  #map(class="odometer-map")
  p(id="location")
    span(class="label success"
    | Fetching your location v
```

```
<div id="display">
    <input id="clock" class="xlarge" type="text" value="00:00:00.0" readonly="readonly"/>
</div>
<div id="controls">
    <button id="start" type="button" class="btn primary" disabled>Start</button>
    <button id="reset" type="button" class="btn :disabled" disabled>Reset</button>
    <div id="options"><input type="checkbox" id="no-music"> No Music Please</div>
</div>
<div id="loading"><img src="images/ajax-loader.gif" alt="Loading"> Loading...</div>
<div id="app" style="display: none">
    <div id="track">
        Distance Traveled: <span id="distance">0</span> mile(s)
        <div id="actions">
            <button id="save" type="submit" class="btn success">Save</button>
        </div>
    </div>
    <div id="where">
        <div id="map" class="odometer-map"></div>
        <p id="location">
            <span class="label success">New</span>
            Fetching your location with HTML 5 geolocation...
        </p>
    </div>
```

Scalate

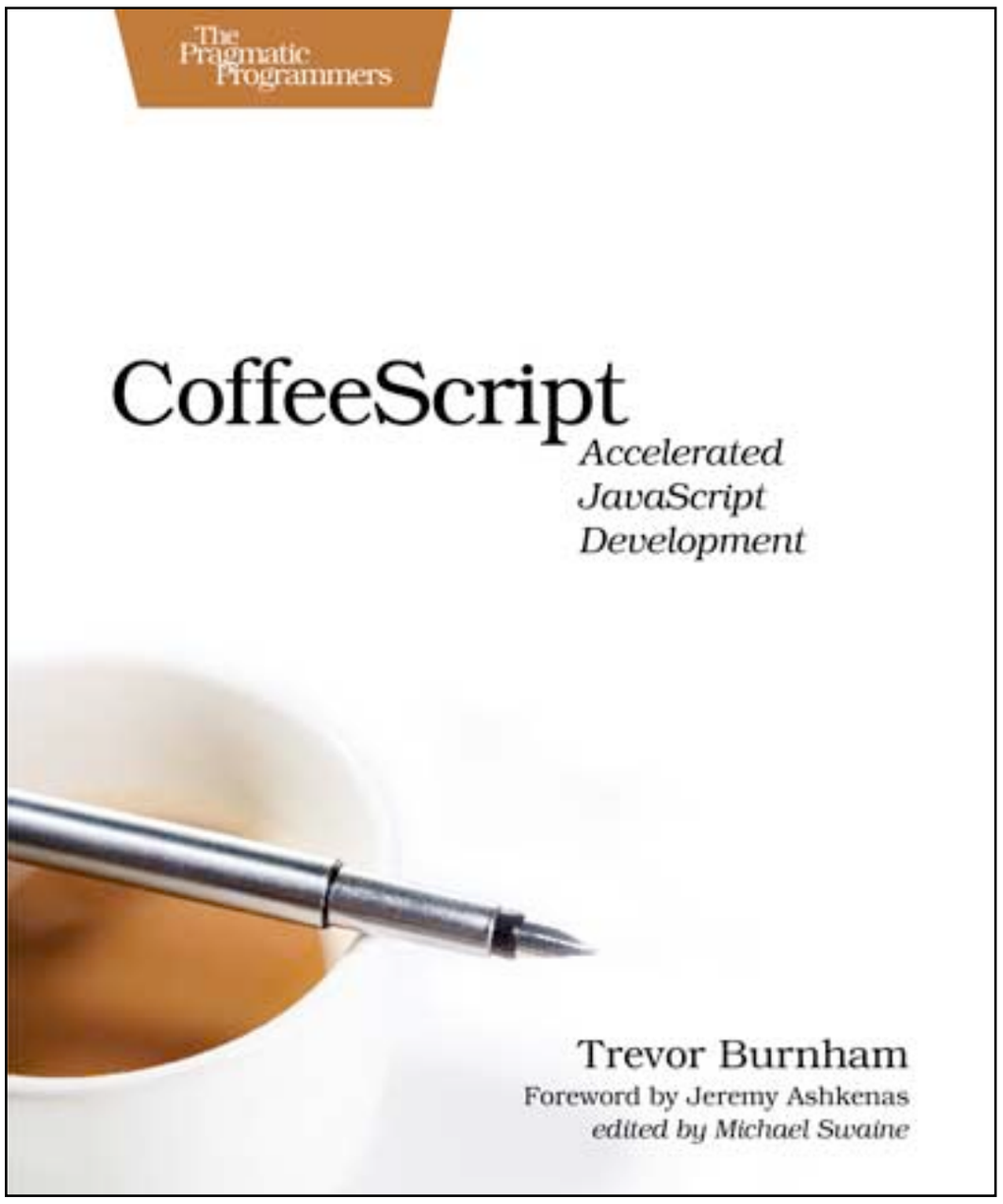
© 2012 Raible Designs

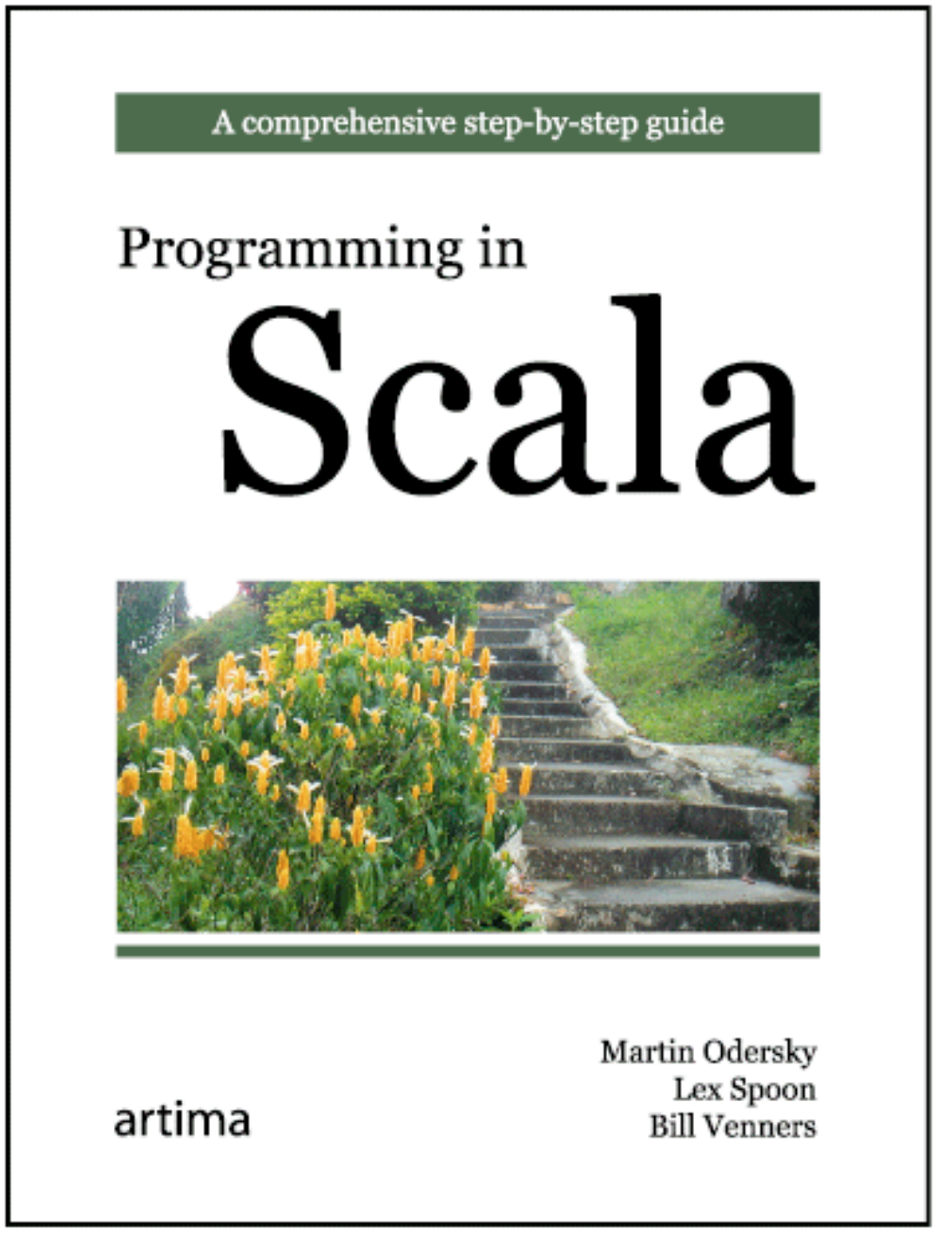
My Development Experience



© 2012 Raible Designs

Getting Started





© 2012 Raible Designs

Developing with Play Scala

© 2012 Raible Designs

Tools I started with...

Google

© 2012 Raible Designs

Scalate Module

Play! Home Learn Download Community Code Ecosystem **Modules**

Community contributed extensions

All modules
Scala
Google App Engine
PDF Generation
SASS and SCSS
Google Web Toolkit
MongoDB
Simple search
Objectify

Scalate [scalate] module

Scalate Template engine support (more info about scalate: http://scalate.fusesource.org).

This module depends on the scala module, so you will need to enable both modules.

required play version: 1.1r956
required play scala version: 0.7.2

You can start a new project with the following command:
play new myapp --with scala,scalate

(assuming you have only scala-0.7.2 and scalate-0.7.2 installed)

http://github.com/pk11/play-scalate

Written by Peter Hausel.

Published releases

scalate-0.7.2 ★	Jul 29, 2010	Documentation	Download

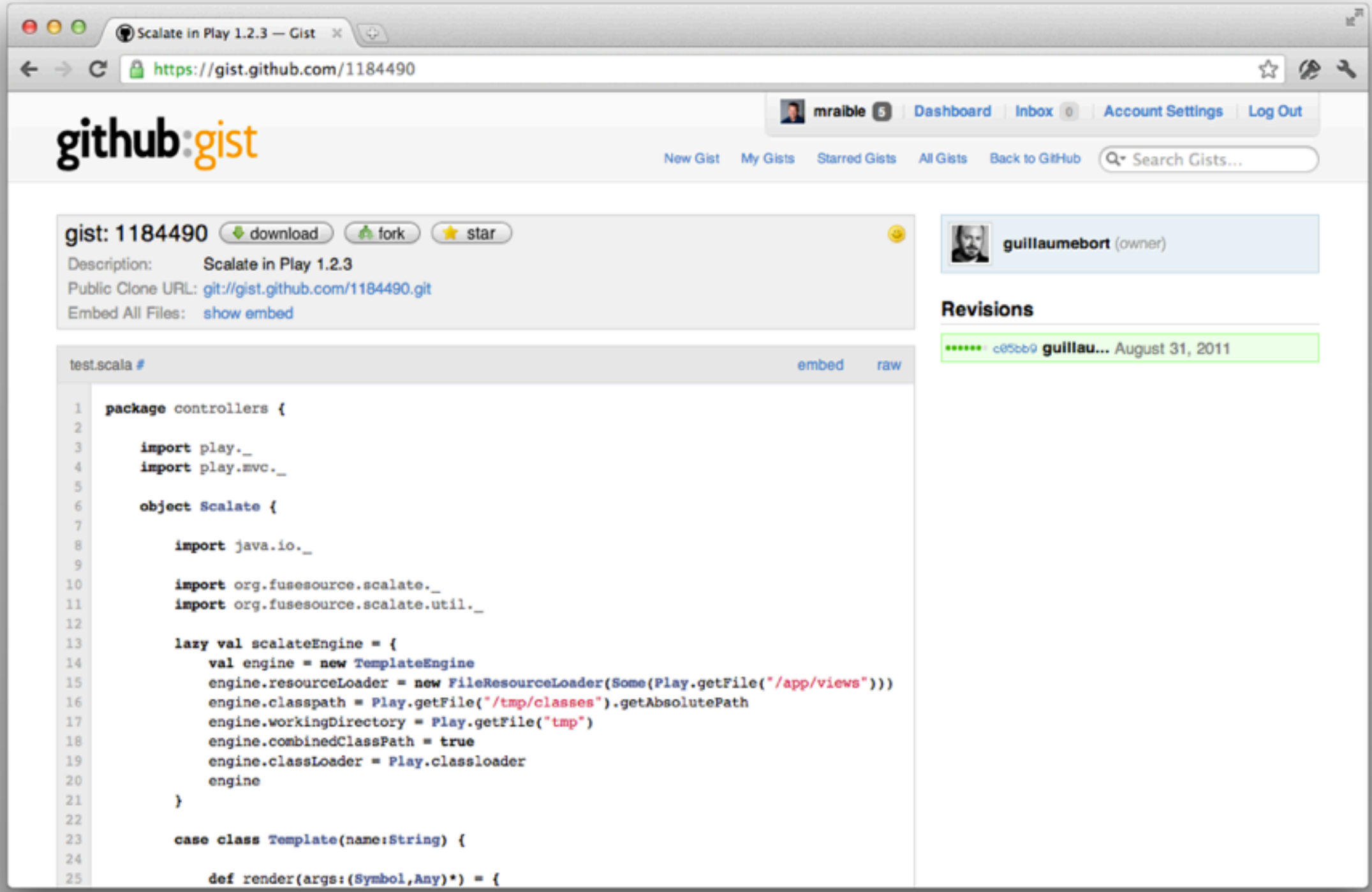
Developer login

Use your OpenID to connect and manage your modules.

Login

© 2012 Raible Designs

Scalate Integration Solution



© 2012 Raible Designs

Integrating Scalate with Play

```
require:
    - play
    - play -> scala 0.9.1
    - org.fusesource.scalate -> scalate-core 1.5.2-scala_2.8.1:
        transitive: false
    - org.fusesource.scalate -> scalate-util 1.5.2-scala_2.8.1:
transitive: false
```

```
play deps --sync
```

© 2012 Raible Designs

Integrating Scalate with Play

```
import play.Play

object ScalateTemplate {

  import org.fusesource.scalate._
  import org.fusesource.scalate.util._

  lazy val scalateEngine = {
    val engine = new TemplateEngine
    engine.resourceLoader = new FileResourceLoader(Some(Play.getFile("/app/views")))
    engine.classpath = Play.getFile("/tmp/classes").getAbsolutePath
    engine.workingDirectory = Play.getFile("tmp")
    engine.combinedClassPath = true
    engine.classLoader = Play.classloader
    engine
  }

  case class Template(name: String) {
    val scalateType = "." + Play.configuration.get("scalate");

    def render(args: (Symbol, Any)*) = {
      scalateEngine.layout(name + scalateType, args.map {
        case (k, v) => k.name -> v
      } toMap)
    }
  }

  def apply(template: String) = Template(template)
}
```

© 2012 Raible Designs

Integrating Scalate with Play

```scala
import play.mvc.Http

trait Scalate {

  def render(args: (Symbol, Any)*) = {
    val template = Http.Request.current().action.replace(".", "/")
    ScalateTemplate(template).render(args: _*);
  }
}
```

```scala
import play.mvc._
import models._

object Application extends Controller with Scalate {

  def index = {
    render('user -> User("Raible"))
  }
}
```

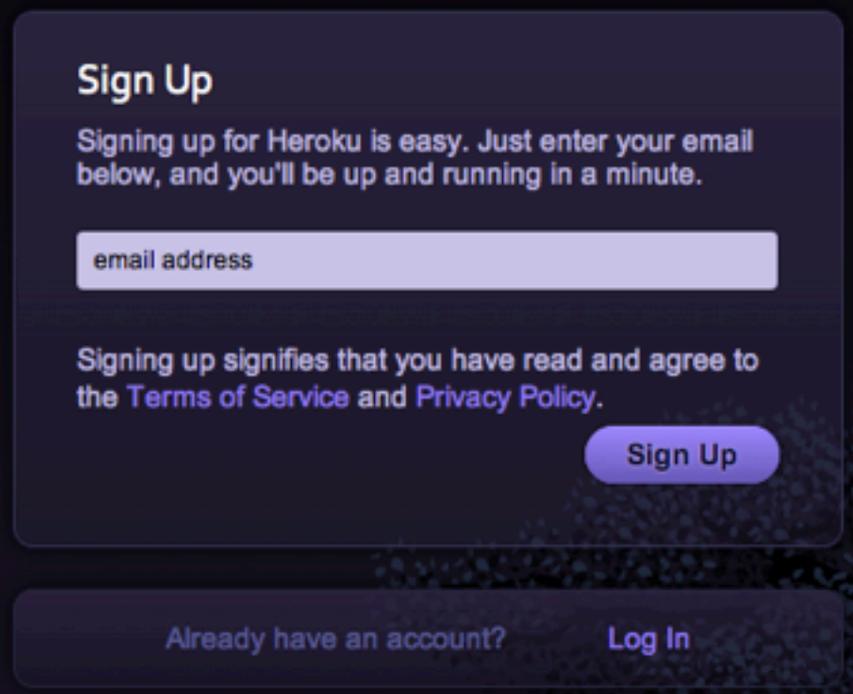
```scala
package models

case class User(name:String)
```

```
-@ var user: models.User
p Hi #{user.name},
- for(i <- 1 to 3)
  p= i
p See, I can count!
```

© 2012 Raible Designs

heroku

http://blog.heroku.com/archives/2011/8/29/play/




```
Cannot start in PROD mode with errors
Template compilation error (In /app/views/Application/index.jade around line 2)
The template /app/views/Application/index.jade does not compile : #{user.name} is not closed.
play.exceptions.TemplateCompilationException: #{user.name} is not closed.
       at play.templates.TemplateCompiler.generate(TemplateCompiler.java:102)
       at play.templates.TemplateCompiler.compile(TemplateCompiler.java:15)
       at play.templates.GroovyTemplateCompiler.compile(GroovyTemplateCompiler.java:4 1)
```

Integrating Scalate with Play

Wednesday September 07, 2011

Integrating Scalate and Jade with Play 1.2.3

At the beginning of this year, I decided I wanted to learn **Scala**. Since I'm a Web Frameworks Aficionado, I figured the best way to do that would be to learn **Lift**. I entered these two items on my todo list and let them lie for a couple months. After attending **TSSJS 2011** and having a conversation with **James Strachan**, I added a couple more technologies to my learning list. James had great things to say about both **CoffeeScript** and **Jade** and I decided to learn those as well.

In May, **Devoxx** announced their Call For Papers and I started reminiscing about how awesome **last year's trip** was. I decided I'd try to get accepted again and started brainstorming about talks I'd like to give. I came up with "Comparing Scala Web Frameworks" and "HTML5 with Play Scala, CoffeeScript and Jade". The reason I chose Play over Lift for the latter talk is because I think it fits a lot more with the MVC mindset I have and the easy-to-learn nature of web frameworks I enjoy using. Both topics sounded very interesting to me, and I figured they'd also inspire me to learn the technologies in a brute-force fashion; where I was under a time constraint and would be embarrassed in front of a large audience if I didn't succeed.

In mid-July, I got an email from **Stephan** inviting me to speak again at the 10th edition of Devoxx. I smile splashed across my face and I quickly realized I had a lot to learn. Since I was still in vacation mode after **summer vacation in Montana**, I decided to wait until I returned from **Cape Cod** to start studying. While on my 2nd summer vacation, I received an email from Devoxx stating that they'd like me present "HTML5 with Play/Scala, CoffeeScript and Jade".

http://raibledesigns.com/rd/entry/integrating_scalate_and_jade_with

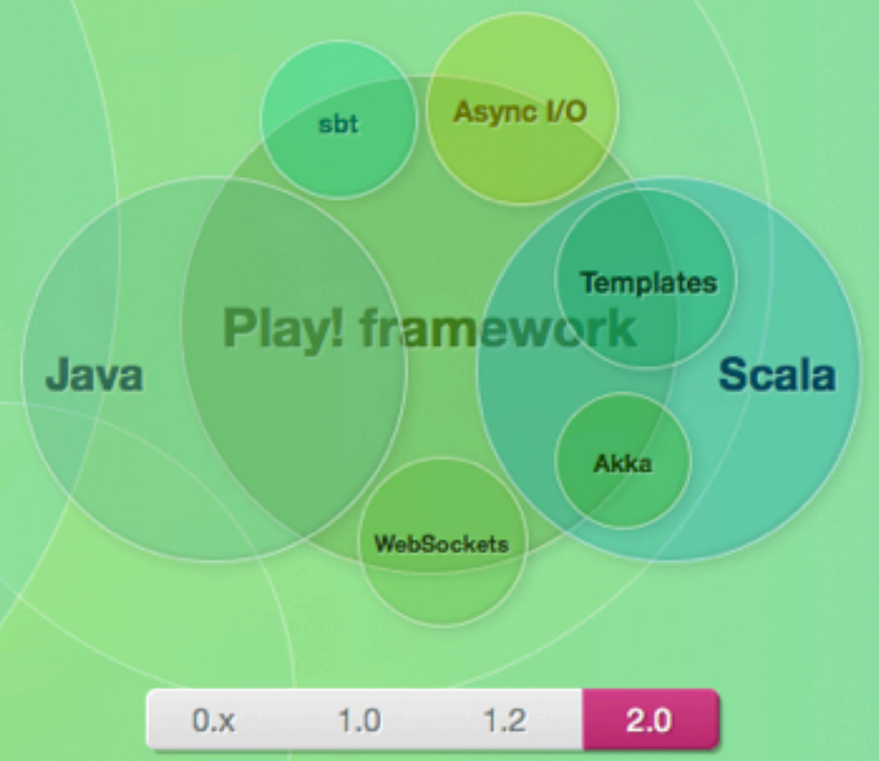
© 2012 Raible Designs

Play 2.0

Working on Play 2.0

It's time to move on! We are working on the next major version of Play framework, integrating a brand new build system and awesome asynchronous features all with native Java and Scala support.

Play 2.0 is still under heavy development and APIs are likely to change, but you can already have a look and download the preview version.



© 2012 Raible Designs

Play 2.0

Track our progress *and discover what's new in Play 2.0.*

Build system 11 /13

HTTP, Server and MVC 8 /9

Java and Scala API 5 /9

Datastores bindings 2 /2

Test environment 0 /0

Documentation and samples 0 /0

Related tickets on lighthouse

- #12 Support multiple routes file and inclusion
- #7 Play console
- #13 Multi projects support
- ~~#11 Compile routes file~~
- ~~#10 Compile Play templates~~
- ~~#9 Report compilation and execution errors in Web browser~~
- #8 WAR packaging
- #6 Allow to package and publish Play application
- ~~#5 Create a Play SBT plugin~~
- ~~#2 Live compilation and reloading for both Java and Scala~~
- #3 Improve Java compilation error messages

© 2012 Raible Designs

Play 2.0

Track our progress *and discover what's new in Play 2.0.*

Build system 11 /13

HTTP, Server and MVC 9 /9

Java and Scala API 7 /9

Datastores bindings 2 /2

Test environment 0 /0

Documentation and samples 0 /0

Related tickets on lighthouse

- #7 Play console
- #13 Multi projects support
- #12 Support multiple routes file and inclusion
- ~~#3 Improve Java compilation error messages~~
- ~~#11 Compile routes file~~
- ~~#10 Compile Play templates~~
- ~~#9 Report compilation and execution errors in Web browser~~
- #8 WAR packaging
- #6 Allow to package and publish Play application
- ~~#5 Create a Play SBT plugin~~
- ~~#2 Live compilation and reloading for both Java and Scala~~

© 2012 Raible Designs

Play 2.0 Beta

Wednesday November 16, 2011

Play 2.0, A web framework for a new era [Edit]

This week, I'm in Antwerp, Belgium for the annual **Devoxx** conference. After traveling 21 hours door-to-door yesterday, I woke up and came to the conference to attend some talks on Play and PhoneGap. I just got out of the **session on Play 2.0**, which was presented by **Sadek Drobi** and **Guillaume Bort**. Below are my notes from this presentation.

The Play 2.0 beta is out! You can read more about this release **on the mailing list**. This beta includes native support for both Scala and Java, meaning you can use both in the same project. The release also bundles **Akka** and **SBT** by default.

In other news, **Play 2.0 is now part of the Typesafe Stack**. Typesafe is the Scala company, started by the founder of Scala (**Martin Odersky**) and the founder of Akka (**Jonas Bonér**). Guillaume is also joining the Typesafe Advisory Board.

Sadek and Guillaume both work at **zenexity**, where Play is the secret weapon for the web applications they've built for the last decade. Play was born in the real world. They kept listening to the market to see what they should add to the project. At some point, they realized they couldn't keep adding to the old model and they needed to create something new.

The web has evolved from static pages to dynamic pages (ASP, PHP). From there, we moved to structured web applications with frameworks and MVC. Then the web moved to Ajax and long-polling to more real-time, live features. And this changes everything.

Now we need to adapt our tools. We need to handle tremendous flows of data. Need to improve expressiveness for concurrent code. We need to pick the appropriate datastore for the problem (not only SQL). We need to integrate with rapidly-evolving client side technologies like JavaScript, CoffeeScript, and Dart. We need to use elastic deployment that allows scaling up and scaling down.

http://raibledesigns.com/rd/entry/play_2_0_a_web

© 2012 Raible Designs

A Nice Break ...

CoffeeScript with Play

```
require:
    - play
    - play -> coffee 1.0
```

```
script(type="text/javascript" src={uri("/public/javascripts/script.coffee")})
```

```
:plain
  <script type="text/coffeescript">
    $(document).ready ->
      $("#start,#reset").removeAttr "disabled"

      $('#start').click ->
        StopWatch.start this, $('#clock')
        $('#dashboard').show()
        Map.start()
  </script>
```

```
script(type="text/javascript" src={uri("/public/javascripts/libs/coffee-script.js")})
```

© 2012 Raible Designs

CoffeeScript with Play

Tuesday September 27, 2011

Trying to make CoffeeScript work with Scalate and Play

A few weeks ago, I wrote about **integrating Scalate with Play**.

> The next steps in my Play Scala adventure will be trying to get the **CoffeeScript module** to work. I also hope to integrate **HTML5 Boilerplate** with Jade and **Scalate Layouts**.

Since my last writing, the Scalate Team has created a new branch for Scala 2.8.x (that's compatible with Play) and **released 1.5.2**. To upgrade my Play application to use this version, I changed my dependencies.yml to have the following:

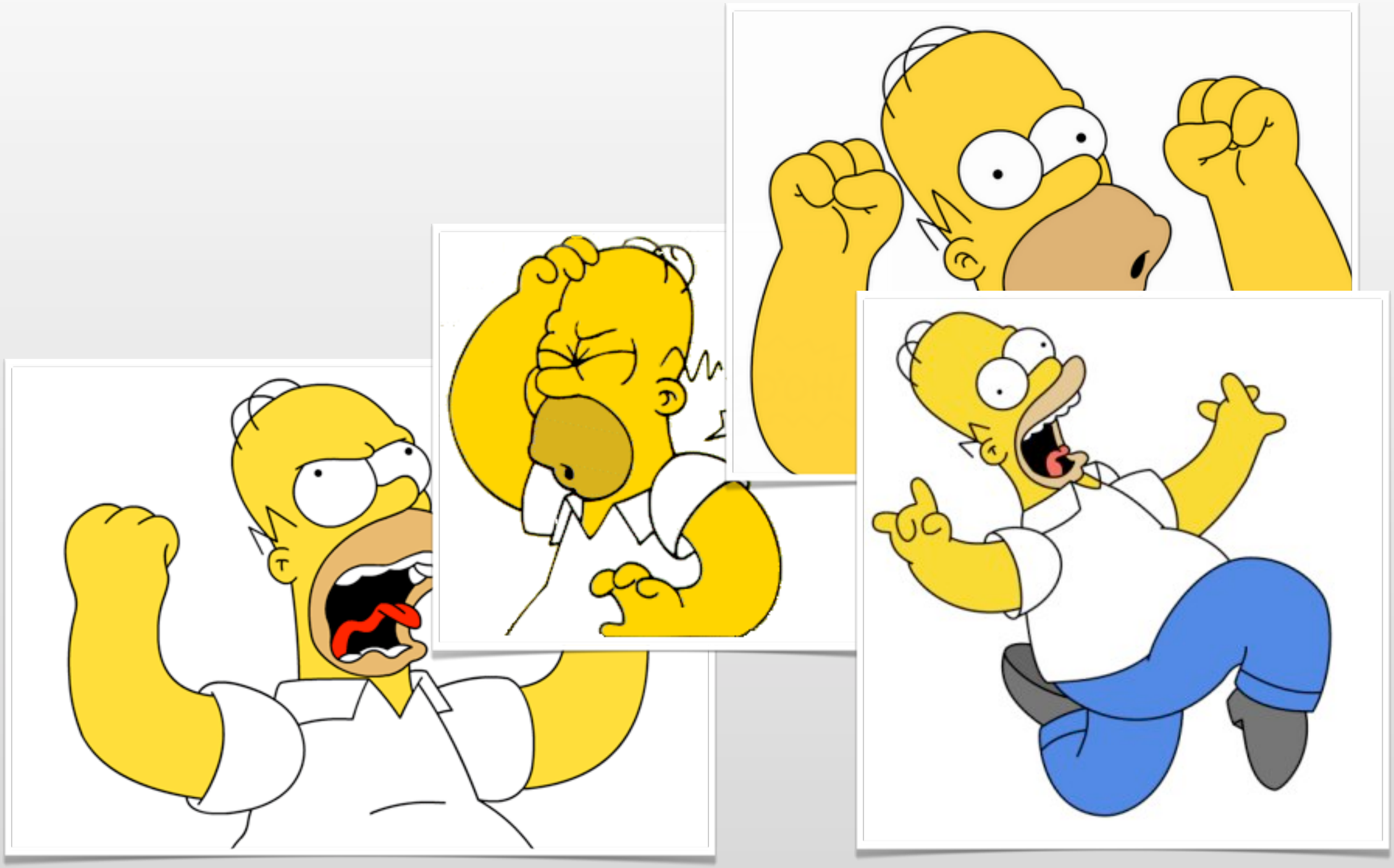
```
- org.fusesource.scalate -> scalate-core 1.5.2-scala_2.8.1:
    transitive: false
- org.fusesource.scalate -> scalate-util 1.5.2-scala_2.8.1:
    transitive: false
```

Unfortunately, this release breaks Scalate's CoffeeScript support because it **wraps the code with illegal comments**. This has been fixed in the latest snapshot, but no new release has been cut. However, even if it did work, it's not quite what I'm looking for. The 1.5.2 release allows for compiling inline CoffeeScript on-the-fly, but I'd rather store my .coffee files external to the page.

http://raibledesigns.com/rd/entry/trying_to_make_coffeescript_work

© 2012 Raible Designs

My Development Experience



© 2012 Raible Designs

HTML5 ★ BOILERPLATE

A rock-solid default for HTML5 awesome.

WHY IT IS AWESOME

- ★ Cross-browser compatible (IE6, yeah we got that.)
- ★ HTML5 ready. Use the new tags with certainty.
- ★ Optimal caching and compression rules for grade-A performance
- ★ Best practice site configuration defaults
- ★ Mobile browser optimizations
- ★ Progressive enhancement graceful degradation … yeah yeah we got that
- ★ IE specific classes for maximum cross-browser control
- ★ Handy .no-js and .js classes to style based on capability
- ★ Want to write unit tests but lazy? A full, hooked up test suite is waiting for you.

Modernizr

FRONT-END DEVELOPMENT DONE RIGHT

© 2012 Raible Designs

H5BP and Play

```
cd $boilerplate-download
cp 404.html ~/dev/play-more/app/views/errors/404.html
cp *.png ~/dev/play-more/public/.
cp crossdomain.xml ~/dev/play-more/public/.
cp -r css ~/dev/play-more/public/stylesheets/.
cp favicon.ico ~/dev/play-more/public/.
cp humans.txt ~/dev/play-more/public/.
cp -r js/libs ~/dev/play-more/public/javascripts/.
cp robots.txt ~/dev/play-more/public/.
```

© 2012 Raible Designs

Scalate Layouts

```
lazy val scalateEngine = {
  val engine = new TemplateEngine
  engine.resourceLoader = new FileResourceLoader(Some(Play.getFile("/app/views")))
  engine.classpath = Play.getFile("/tmp/classes").getAbsolutePath
  engine.workingDirectory = Play.getFile("tmp")
  engine.combinedClassPath = true
  engine.classLoader = Play.classloader
  engine.layoutStrategy = new DefaultLayoutStrategy(engine,
    Play.getFile("/app/templates/layouts/default" + scalateType).getAbsolutePath)
  engine
}
```

```
-@ val body: String
-@ val title: String = "Play More!"
!!! 5
/ paulirish.com/2008/conditional-stylesheets-vs-css-hacks-answer-neither/
<!--[if lt IE 7]> <html class="no-js ie6 oldie" lang="en"> <![endif]-->
<!--[if IE 7]>    <html class="no-js ie7 oldie" lang="en"> <![endif]-->
<!--[if IE 8]>    <html class="no-js ie8 oldie" lang="en"> <![endif]-->
-# Consider adding an manifest.appcache: h5bp.com/d/Offline
<!--[if gt IE 8]><!--> <html class="no-js" lang="en"> <!--<![endif]-->
head
  meta(charset="utf-8")

  -# Use the .htaccess and remove these lines to avoid edge case issues. More info: h5bp.com/b/378
  meta(http-equiv="X-UA-Compatible" content="IE=edge,chrome=1")

  title=title
```

© 2012 Raible Designs

HTML5 Boilerplate

Wednesday September 28, 2011

Integrating HTML5 Boilerplate with Scalate and Play

HTML5 Boilerplate is a project that provides a number of basic files to help you build an HTML5 application. At its core, it's an HTML template that puts CSS at the top, JavaScript at the bottom, installs Chrome Frame for IE6 users and leverages Modernizr for legacy browser support. It also includes jQuery with the download. One of the major benefits of HTML5 Boilerplate is it ships with a build system (powered by Ant) that concatenates and minimizes CSS and JS for maximum performance. From **html5boilerplate.com**:

> Boilerplate is not a framework, nor does it prescribe any philosophy of development, it's just got some tricks to get your project off the ground quickly and right-footed.

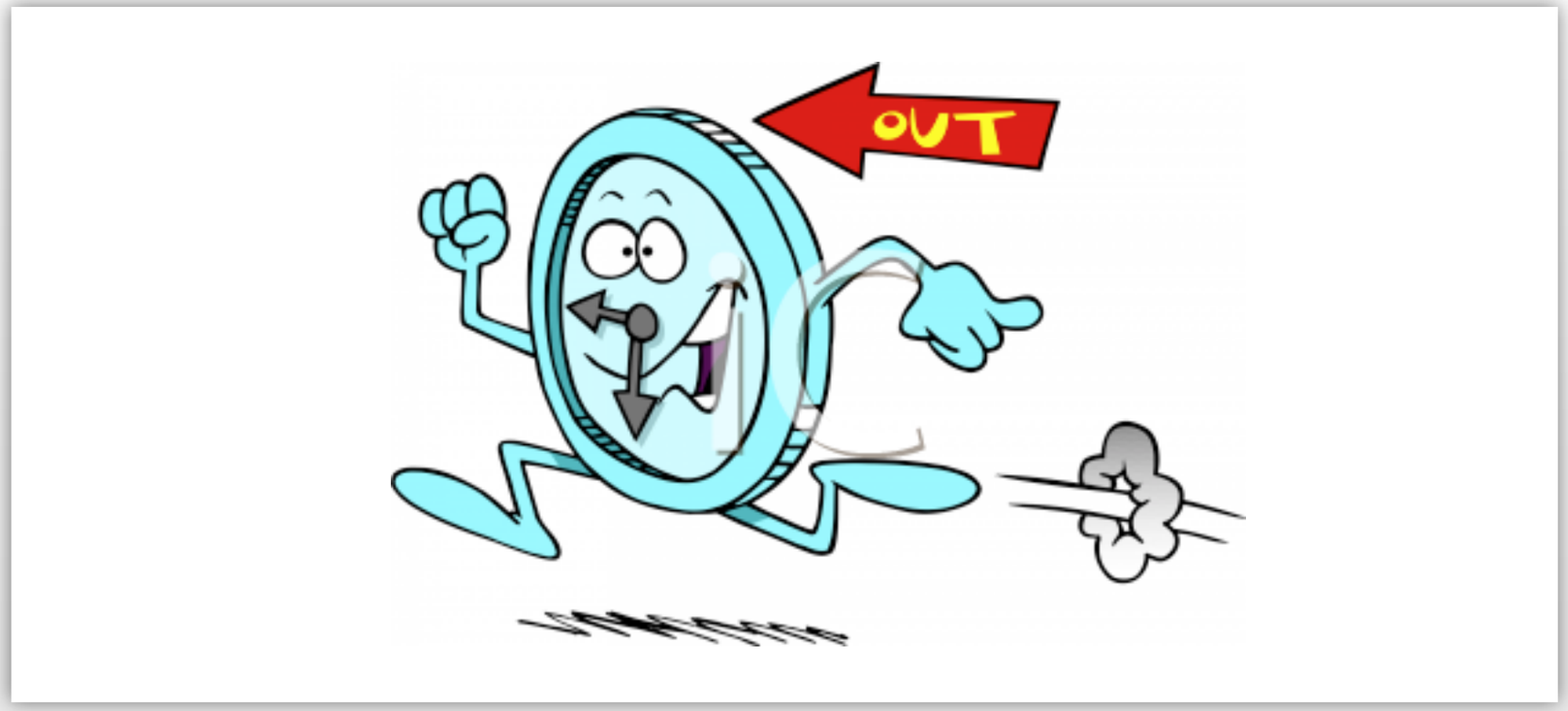
I like the idea of its build system to minify and gzip, but I'd probably only use it if I was working on a project that uses Ant. Since I'm using it in a Play project, the whole Ant build system doesn't help me. Besides, I prefer something like **wro4j**. Wro4j allows you to specify a group of files and then it compiles, minimizes and gzips them all on-the-fly. As far as I know, Play doesn't have any support for Servlet Filters, so using wro4j in Play is not trivial.

The good news is Play has a **GreenScript module** that contains much of the wro4j functionality. However, since I'm using **Scalate** in my project, this goodness is unavailable to me. In the future, the Scalate Team is considering adding **better wro4j, JavaScript and CSS integration**. In the meantime, I'm going to pretend I don't care about concatenation and minimization and trundle along without this feature.

http://raibledesigns.com/rd/entry/integrating_html5_boilerplate_with_scalate

© 2012 Raible Designs

HTML5 Development



© 2012 Raible Designs

StopWatch with Coffee

```coffeescript
# Created by Kåre Byberg © 21.01.2005. Please acknowledge if used on
# other domains than http://www.timpelen.com
# Ported to CoffeeScript by Matt Raible. Also added hours support.
flagClock = 0
flagStop = 0
stopTime = 0
refresh = null
clock = null

start = (button, display) ->
  clock = display
  startDate = new Date()
  startTime = startDate.getTime()
  if flagClock == 0
    $(button).html("Stop")
    flagClock = 1
    counter startTime, display
else
  $(button).html("Start")


@StopWatch = {
  start: start
  reset: reset
}
```

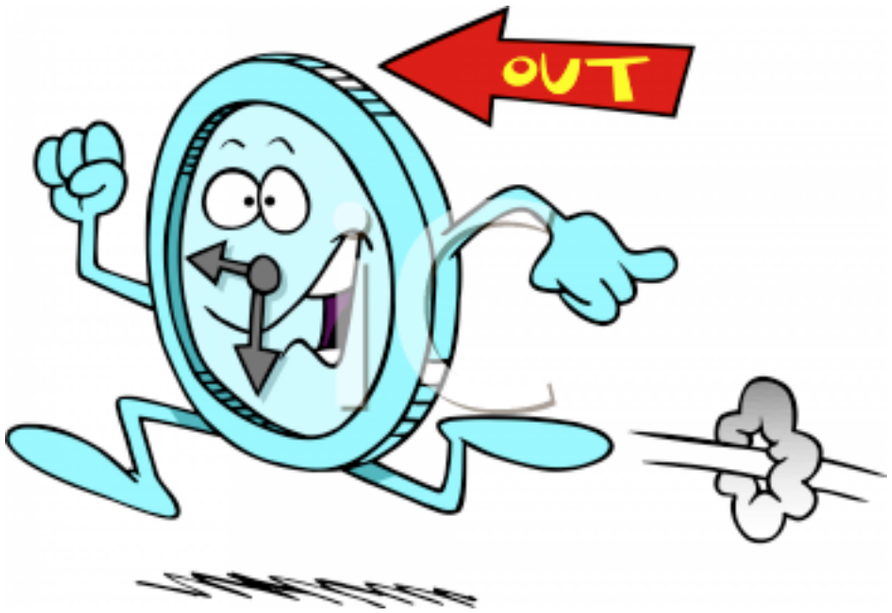
© 2012 Raible Designs

Jade Template for Watch

```
script(type="text/javascript" src={uri("/public/javascripts/stopwatch.coffee")})

#display
  input(id="clock" class="xlarge" type="text" value="00:00:00.0" readonly="readonly")
#controls
  button(id="start" type="button" class="btn primary") Start
  button(id="reset" type="button" class="btn :disabled") Reset

:plain
  <script type="text/coffeescript">
    $(document).ready ->
      $('#start').click ->
        StopWatch.start this, $('#clock')

      $('#reset').click ->
        StopWatch.reset()
  </script>
```

© 2012 Raible Designs

HTML5 Development



© 2012 Raible Designs

HTML5 Geo API

```
// Gets the users current position
navigator.geolocation.getCurrentPosition(successCallback,
                                         errorCallback,
                                         options);
// Request repeated updates of position
watchId = navigator.geolocation.watchPosition(successCallback, errorCallback);

// Cancel the updates
navigator.geolocation.clearWatch(watchId);
```

© 2012 Raible Designs

Google Maps JS API

```coffeescript
# Geolocation with HTML 5 and Google Maps API based on example from maxheapsize:
# http://maxheapsize.com/2009/04/11/getting-the-browsers-geolocation-with-html-5/
# This script is by Merge Database and Design, http://merged.ca/ -- if you use some,
# all, or any of this code, please offer a return link.

map = null
mapCenter = null
geocoder = null
latlng = null
geolocationOptions = { timeout: 10000, enableHighAccuracy: true }
timeoutId = null
```

```coffeescript
initialize = ->
  if Modernizr.geolocation
    navigator.geolocation.getCurrentPosition showMap, geolocationError, geolocationOptions

showMap = (position) ->
  latitude = position.coords.latitude
  longitude = position.coords.longitude
  mapOptions = {
    zoom: 15,
    mapTypeId: google.maps.MapTypeId.ROADMAP
  }
  map = new google.maps.Map(document.getElementById("map"), mapOptions)
  latlng = new google.maps.LatLng(latitude, longitude)
  map.setCenter(latlng)
```

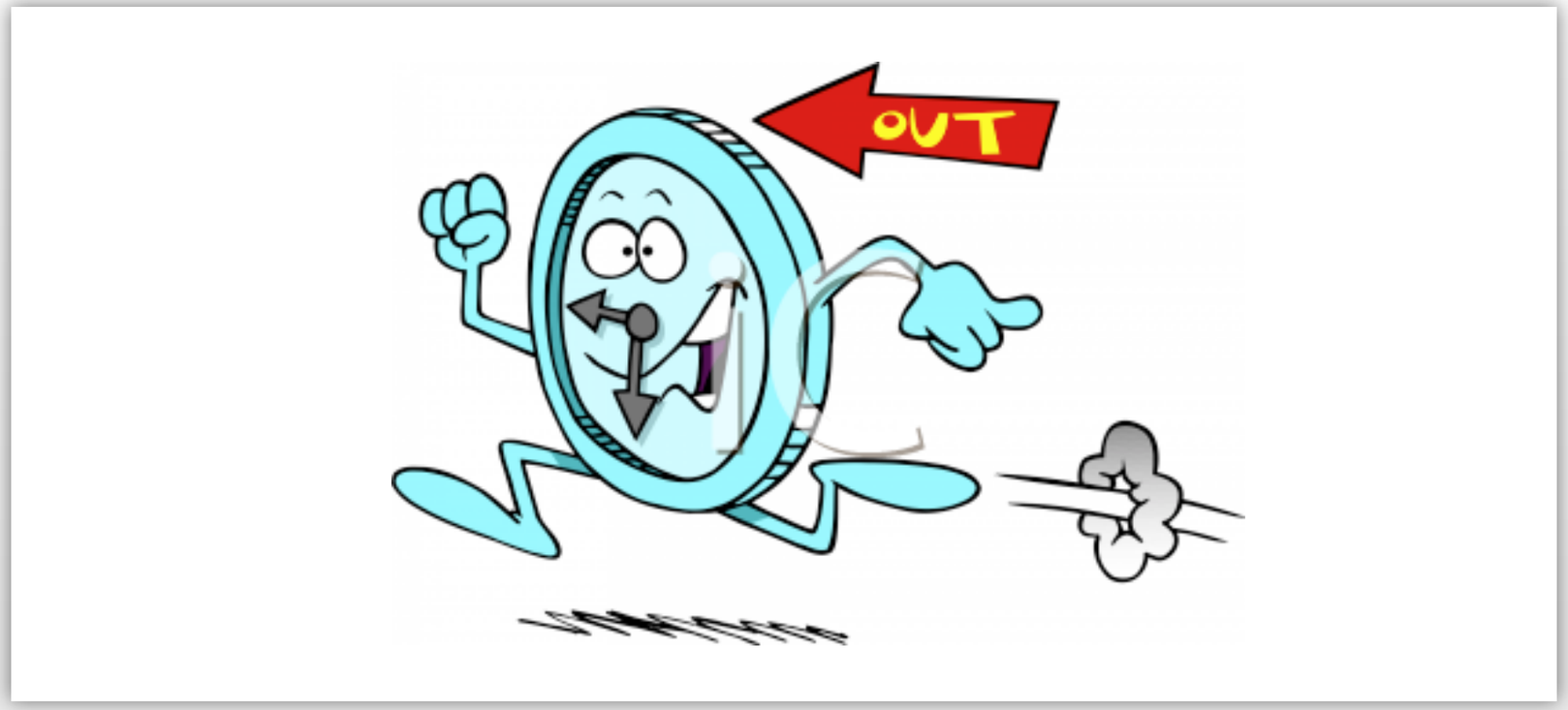
© 2012 Raible Designs

Jade View for Map

```
-# http://merged.ca/iphone/html5-geolocation

script(type="text/javascript" src="http://www.google.com/jsapi")
script(type="text/javascript" src="http://maps.googleapis.com/maps/api/js?sensor=false")

:css
  .demo-map {
    border: 1px solid silver;
    height: 200px;
    margin: 10px auto;
    width: 280px;
  }

#map(class="demo-map")

p(id="location")
  span(class="label success") New
  | Fetching your location with HTML 5 geolocation...

script(type="text/javascript" src={uri("/public/javascripts/odometer.coffee")})
script(type="text/javascript" src={uri("/public/javascripts/map.coffee")})

:javascript
    Map.start();
```

© 2012 Raible Designs

HTML5 Development



© 2012 Raible Designs

Odometer

```coffee
start = (config) ->
  log = config.log
  callback = config.callback
  map = config.map

  if Modernizr.geolocation
    if not config.position
      navigator.geolocation.getCurrentPosition ((position) ->
        startPos = position
        lastPos = position
        $("#startLat").html(startPos.coords.latitude)
        $("#startLon").html(startPos.coords.longitude)
      ), null, geolocationOptions
    else
      startPos = config.position
      lastPos = config.position

    watchId = navigator.geolocation.watchPosition showDistance, null, geolocationOptions

showDistance = (position) ->
  lat = position.coords.latitude
  lng = position.coords.longitude
  $("#currentLat").html(lat)
  $("#currentLon").html(lng)
```

© 2012 Raible Designs

Testing

- Tried Trip Meter on a bike ride
- Said I'd traveled 5 km, when I knew I'd gone 10
  - Was calculating start to end w/o waypoints
- To Visualize: integrated odometer with maps using Google Maps Polylines

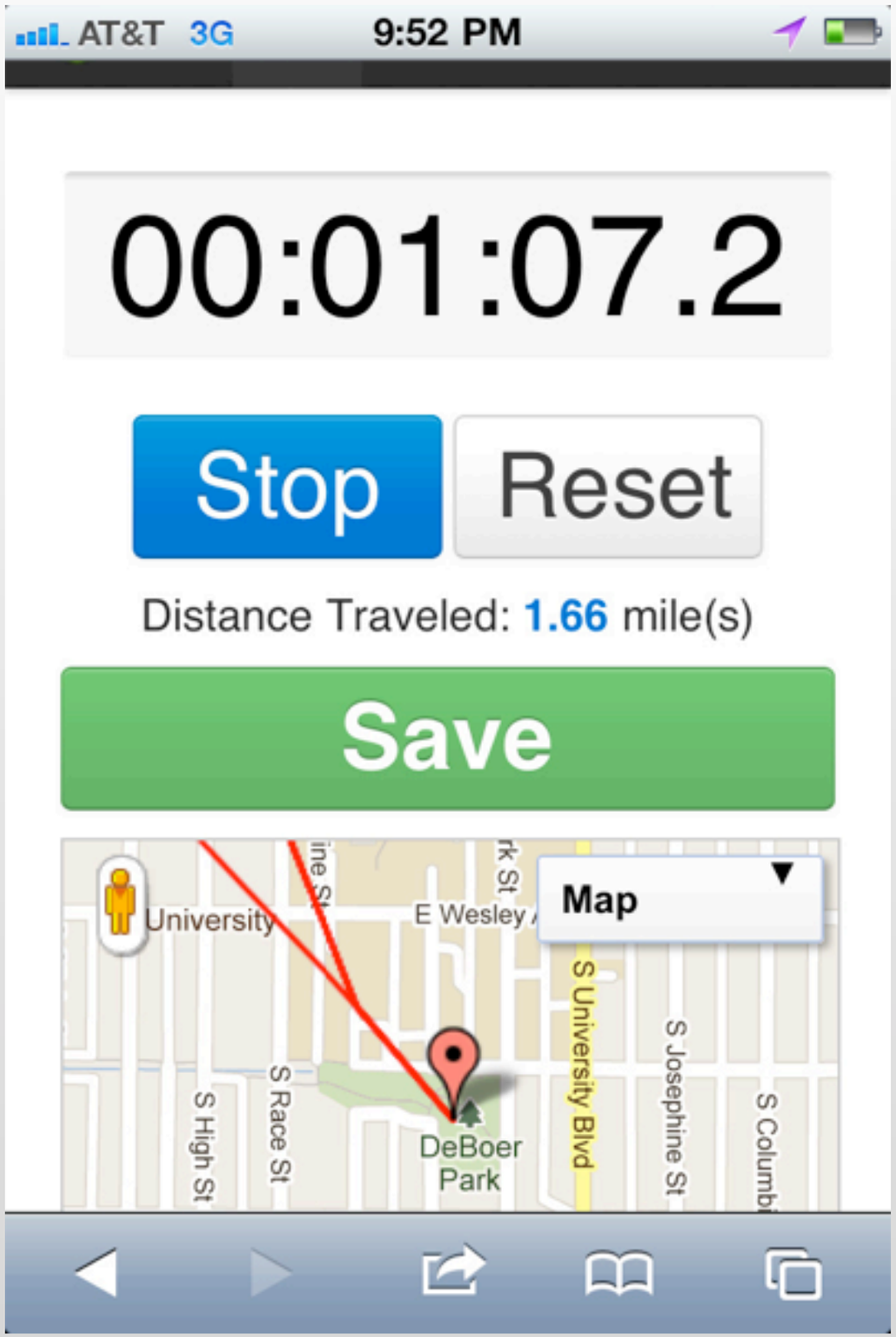
© 2012 Raible Designs

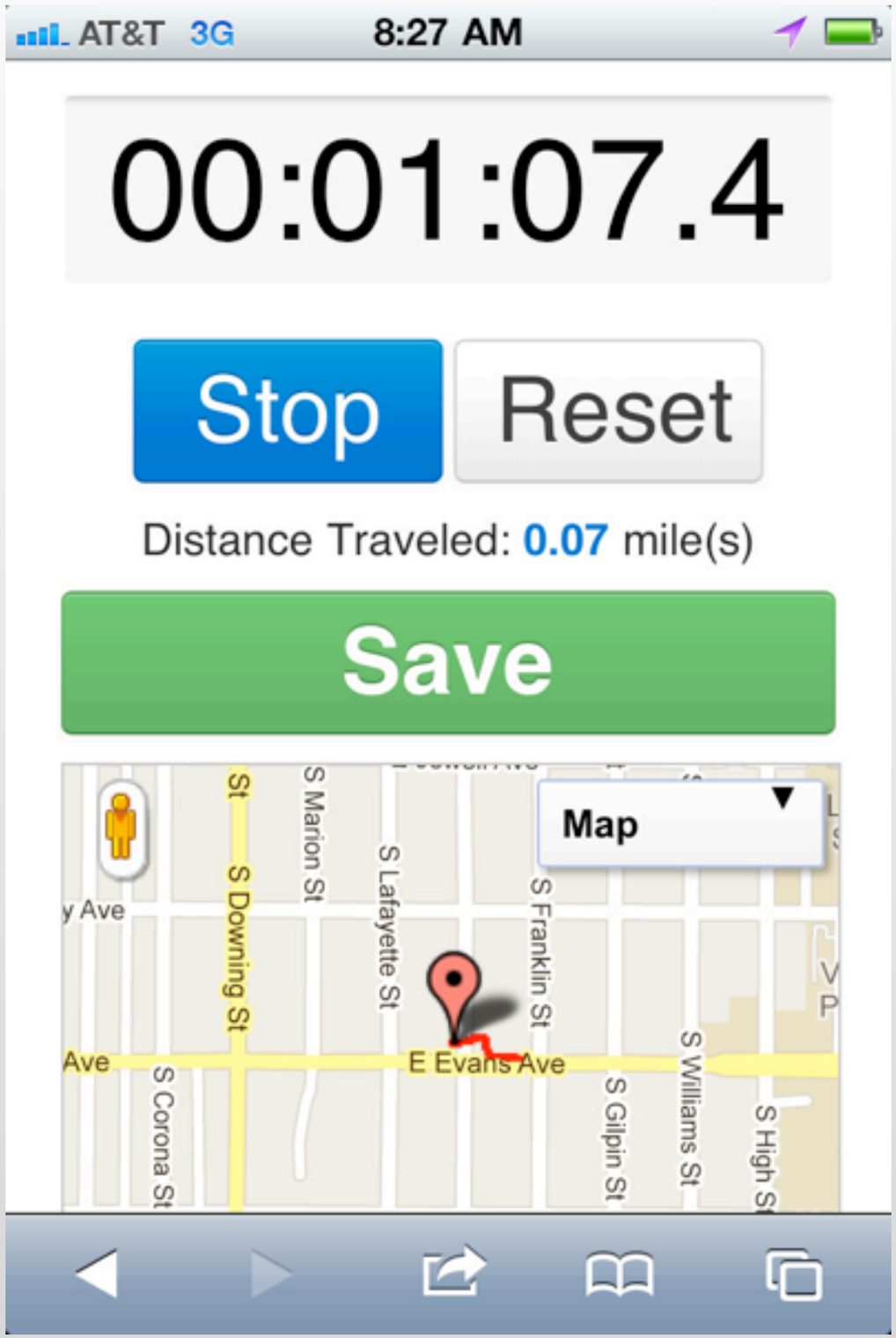
Discovered

- HTML5 Geolocation was highly inaccurate
  - Fixed by passing {enableHighAccuracy: true} to navigator.geolocation.watchPosition()

© 2012 Raible Designs

Discovered

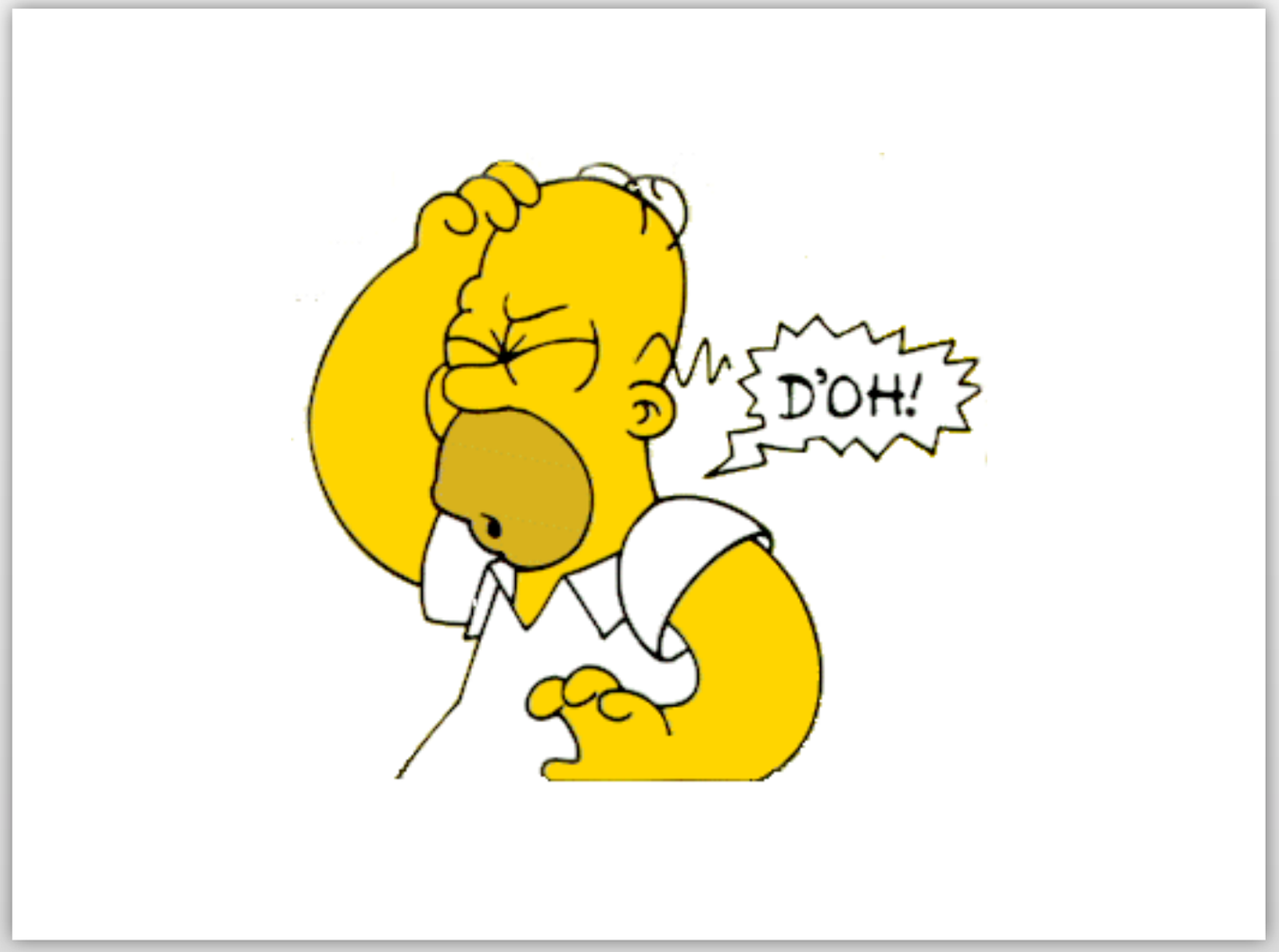




© 2012 Raible Designs

Show Stopper?

- Geolocation doesn't run in the background



© 2012 Raible Designs

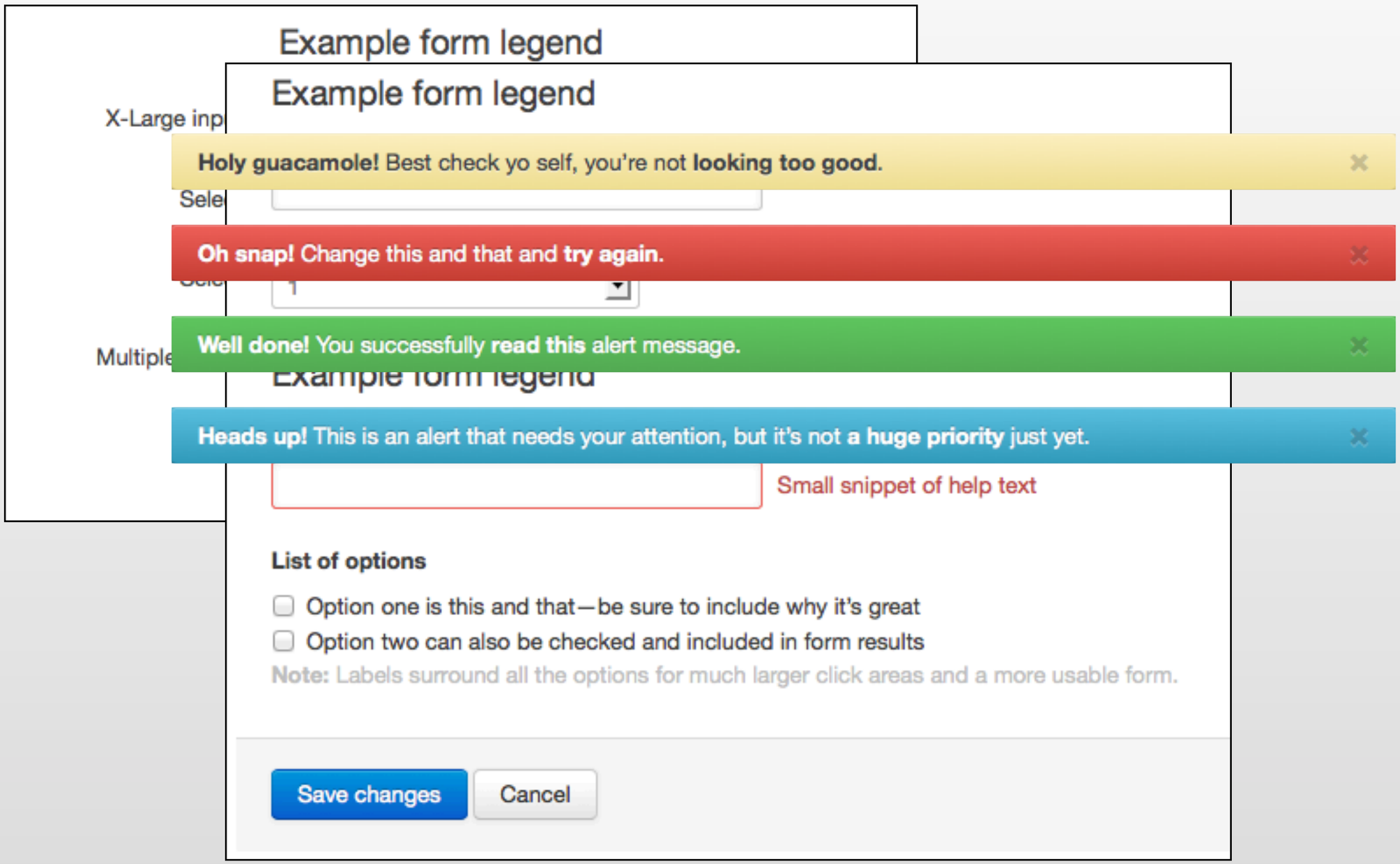
Making it look good

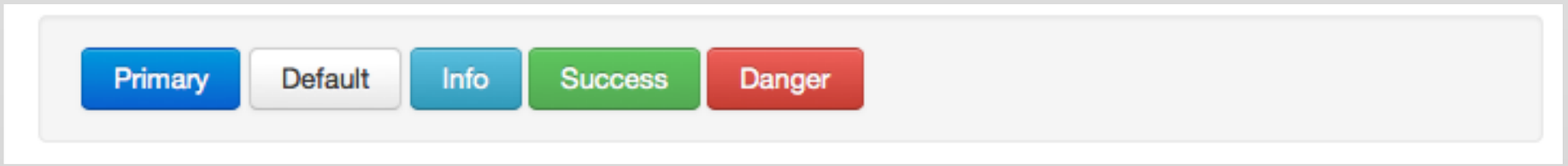
© 2012 Raible Designs

Twitter's Bootstrap

© 2012 Raible Designs

Bootstrap





© 2012 Raible Designs

LESS

```less
// LESS

@color: #4D926F;

#header {
  color: @color;
}
h2 {
  color: @color;
}
```

```css
/* Compiled CSS */

#header {
  color: #4D926F;
}
h2 {
  color: #4D926F;
}
```

© 2012 Raible Designs

LESS

```less
// LESS

.rounded-corners (@radius: 5px) {
  border-radius: @radius;
  -webkit-border-radius: @radius;
  -moz-border-radius: @radius;
}

#header {
  .rounded-corners;
}
#footer {
  .rounded-corners(10px);
}
```

```css
/* Compiled CSS */

#header {
  border-radius: 5px;
  -webkit-border-radius: 5px;
  -moz-border-radius: 5px;
}
#footer {
  border-radius: 10px;
  -webkit-border-radius: 10px;
  -moz-border-radius: 10px;
}
```

© 2012 Raible Designs

CSS3 Media Queries

```
@media all and (max-device-width: 480px) {
  /* hide scrollbar on mobile */
  html { overflow-y:hidden }
  /* hide sidebar on mobile */
  .home .span4, .home .page-header, .topbar form {
    display: none
  }
  .home .container {
    width: 320px;
  }
  .about {
    .container, .span10 {
      width: 280px;
    }
    .span10 {
      padding-top: 0px;
    }
  }
}
```

© 2012 Raible Designs

HTML5 Features

- Geolocation
- CSS 3
- Audio
- History
- Local Storage
- Canvas

© 2012 Raible Designs

HTML5 and Bootstrap

Thursday October 20, 2011

Developing with HTML5, CoffeeScript and Twitter's Bootstrap

This article is the fourth in a series about my adventures developing a Fitness Tracking application with HTML5, Play Scala, CoffeeScript and Jade. Previous articles can be found at:

1. **Integrating Scalate and Jade with Play 1.2.3**
2. **Trying to make CoffeeScript work with Scalate and Play**
3. **Integrating HTML5 Boilerplate with Scalate and Play**

Developing Features

After getting my desired infrastructure setup, I started coding like a madman. The first feature I needed was a stopwatch to track the duration of a workout, so I started writing one with CoffeeScript. After spending 20 minutes playing with dates and setTimeout, I searched and found a **stopwatch jQuery plug-in**. I added this to my app, deployed it to **Heroku**, brought up the app on my iPhone 3G, clicked *Start* and started riding my bike to work.

When I arrived, I unlocked my phone and discovered that the time had stopped. At first, I thought this was a major setback. My disappointed disappeared when I found a **Super Neat JavaScript Stopwatch** and **Kåre Byberg's version** that worked just fine. This stopwatch used setTimeout, so by keeping the start time, the app on the phone would *catch up* as soon as you unlocked it. I ported Kåre's script to CoffeeScript and rejoiced in my working stopwatch.

http://raibledesigns.com/rd/entry/developing_with_html5_coffeescript_and

© 2012 Raible Designs

Anorm and PostgreSQL

- I'm a big fan of ORMs like Hibernate and JPA
  - Learn a new JDBC abstraction? Really!?
- Anorm is *and will be* the default for Play Scala
- Chose PostgreSQL since that's what Heroku uses

© 2012 Raible Designs

Data Model

```
package models

import play.db.anorm._
import play.db.anorm.defaults._

case class Athlete(
  id: Pk[Long],
  email: String, password: String, firstName: String, lastName: String
  ) {
}

object Athlete extends Magic[Athlete] {
  def connect(email: String, password: String) = {
    Athlete.find("email = {email} and password = {password}")
      .on("email" -> email, "password" -> password)
      .first()
  }
}
```

© 2012 Raible Designs

ScalaTest

```
import play.test._

import org.scalatest._
import org.scalatest.matchers._

class BasicTests extends UnitFlatSpec with ShouldMatchers with BeforeAndAfterEach {

  import models._
  import play.db.anorm._

  override def beforeEach() {
      Fixtures.deleteDatabase()
  }

  it should "create and retrieve a Athlete" in {

      var user = Athlete(NotAssigned, "jim@gmail.com", "secret", "Jim", "Smith")
      Athlete.create(user)

      val jim = Athlete.find(
          "email={email}").on("email" -> "jim@gmail.com"
      ).first()

      jim should not be (None)
      jim.get.firstName should be("Jim")

  }
```

© 2012 Hallo Designs

Anorm in Action

```
object Workout extends Magic[Workout] {

  def allWithAthlete: List[(Workout, Athlete)] =
    SQL(
      """
          select * from Workout w
          join Athlete a on w.athleteId = a.id
          order by w.postedAt desc
      """
    ).as(Workout ~< Athlete ^^ flatten *)

  def allWithAthleteAndComments: List[(Workout, Athlete, List[Comment])] =
    SQL(
      """
          select * from Workout w
          join Athlete a on w.athleteId = a.id
          left join Comment c on c.workoutId = w.id
          order by w.postedAt desc
      """
    ).as(Workout ~< Athlete ~< Workout.spanM(Comment) ^^ flatten *)
}
```

© 2012 Raible Designs

Controller and View

```
def show(id: Long) = {
  Workout.byIdWithAthleteAndComments(id).map { w =>
    render(
      'workout -> w,
      'pagination -> w._1.prevNext
    )
  } getOrElse {
    NotFound("No such Profile")
  }
}
```

```
-@ val workout:(models.Workout,models.Athlete,Seq[models.Comment])
-
  var commentsTitle = "No Comments"
  if (workout._3.size > 0)
    commentsTitle = workout._3.size + " comments, lastest by " +
    | workout._3(workout._3.size - 1).author

div(class="workout")
  h2.title
    a(href={action(controllers.Profile.show(workout._1.id()))}) #{workout._1.title}
    .metadata
      span.user Posted by #{workout._2.firstName} on
      span.date #{workout._1.postedAt}
    .description
      = workout._1.description
```

Anorm, Dates & PostgreSQL

```
@OnApplicationStart
class BootStrap extends Job {

  override def doJob() {

    import models._
    import play.test._

    // Import initial data if the database is empty
    if (Athlete.count().single() == 0) {
      Yaml[List[Any]]("initial-data.yml").foreach {
        _ match {
          case a: Athlete => Athlete.create(a)
          case w: Workout => Workout.create(w)
          case c: Comment => Comment.create(c)
        }
      }
    }
  }
}
```

© 2012 Raible Designs

Anorm, Dates & PostgreSQL

- Discovered “support of Date for insertion” was added to Anorm in August 2011
- Cloned play-scala, built locally and uploaded
- Modified dependencies.yml to use new version

```
require:
    - play
    - play -> coffee 1.0
    - play -> less 0.3.compatibility
    - upgrades -> scala 0.9.1-20111025

repositories:
    - upgrades:
        type: http
        artifact: "http://static.raibledesigns.com/[module]-[revision].zip"
        contains:
            - upgrades -> *
```

© 2012 Raible Designs

Anorm and PostgreSQL

Wednesday November 02, 2011

Play Scala's Anorm, Heroku and PostgreSQL Issues

This article is the 5th in a series on about my adventures developing a Fitness Tracking application for **my talk at Devoxx** in two weeks. Previous articles can be found at:

1. **Integrating Scalate and Jade with Play 1.2.3**
2. **Trying to make CoffeeScript work with Scalate and Play**
3. **Integrating HTML5 Boilerplate with Scalate and Play**
4. **Developing with HTML5, CoffeeScript and Twitter's Bootstrap**

Anorm

In my **previous article**, I described how I created my application's features using CoffeeScript and make it look good using Twitter's Bootstrap. Next, I turned to persisting this data with **Anorm**.

> The Scala module includes a brand new data access layer called Anorm that uses plain SQL to make your database request and provides several API to parse and transform the resulting dataset.

http://raibledesigns.com/rd/entry/play_scala_s_anorm_heroku

© 2012 Raible Designs

More Scalate Goodness

```
def populateRenderArgs(args: (Symbol, Any)*): Map[String, Any] = {
  val renderArgs = Scope.RenderArgs.current();

  args.foreach {
    o =>
      renderArgs.put(o._1.name, o._2)
  }

  renderArgs.put("session", Scope.Session.current())
  renderArgs.put("request", Http.Request.current())
  renderArgs.put("flash", Scope.Flash.current())
  renderArgs.put("params", Scope.Params.current())
  renderArgs.put("errors", validationErrors)
  renderArgs.put("config", Play.configuration)

  // CSS class to add to body
  renderArgs.put("bodyClass", Http.Request.current().action.replace(".", " ").toLowerCase)
  renderArgs.data.toMap
}
```

© 2012 Raible Designs

More Scalate Goodness

```
- front.map { front =>
  - render("workout.jade", Map('workout -> front, 'mode -> "home"))

- captureAttribute("sidebar")
  - Option(older).filterNot(_.isEmpty).map { workouts =>
    .older-workouts
      h3
        | Older workouts
        span.from from this app
      - workouts.map { workout =>
        - render("workout.jade", Map('workout -> workout, 'mode -> "teaser"))
      - }
  - }
- }
```

© 2012 Raible Designs

More Scalate Goodness

```
-@ val sidebar: String = ""
-@ val flash: play.mvc.Scope.Flash
-@ val params: play.mvc.Scope.Params

  .container
    .content
      .page-header
        h1
          = pageHeader
          small
            = pageTagline
    .row
      .span10
        - if (flash.get("success") != null) {
          div(class="alert-message success" data-alert="alert")
            a(class="close" href="#") &times;
            | #{flash.get("success")}
        - }
        !~~ body
      .span4
        = unescape(sidebar)
footer
```

© 2012 Raible Designs

Scalate as a Play Module

Monday November 07, 2011

More Scalate Goodness for Play

This article is the 6th in a series on about my adventures developing a web application with HTML5, Play Scala, CoffeeScript and Jade. Previous articles can be found at:

1. **Integrating Scalate and Jade with Play 1.2.3**
2. **Trying to make CoffeeScript work with Scalate and Play**
3. **Integrating HTML5 Boilerplate with Scalate and Play**
4. **Developing with HTML5, CoffeeScript and Twitter's Bootstrap**
5. **Play Scala's Anorm, Heroku and PostgreSQL Issues**

Last week, I wrote about my adventures with **Anorm** and mentioned I'd made some improvements to Scalate Play interoperability. First of all, I've been using a Scalate trait and ScalateTemplate class to render Jade templates in my application. I described this setup in my **first article on Scalate and Play**.

Adding SiteMesh Features and Default Variables

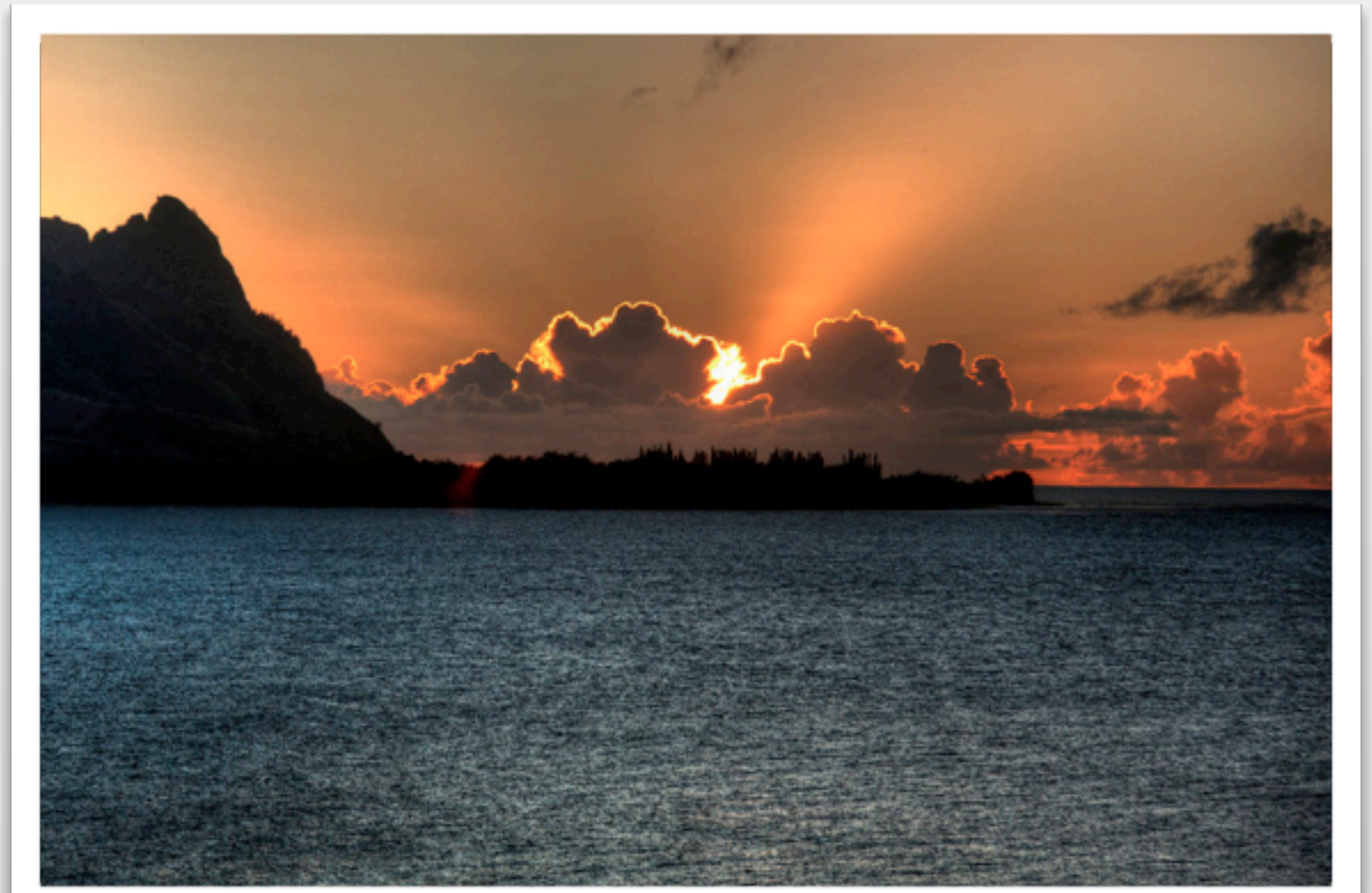
When I started making my app look good with CSS, I started longing for a feature I've used in SiteMesh. That is, to have a body id or class that can identify the page and allow per-page CSS rules. To do this with SiteMesh, you'd have something like the following in your page:

http://raibledesigns.com/rd/entry/more_scalate_goodness_for_play

© 2012 Raible Designs

App was still unusable

- I still hadn’t solved the fundamental problem
- The app couldn’t run in the background on a mobile phone

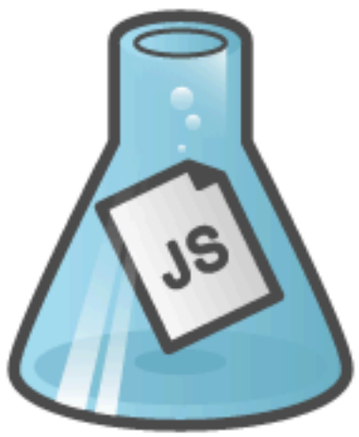


© 2012 Raible Designs

PhoneGap to the Rescue!

With **PhoneGap** you can,

Take advantage of **HTML5** and **CSS3**



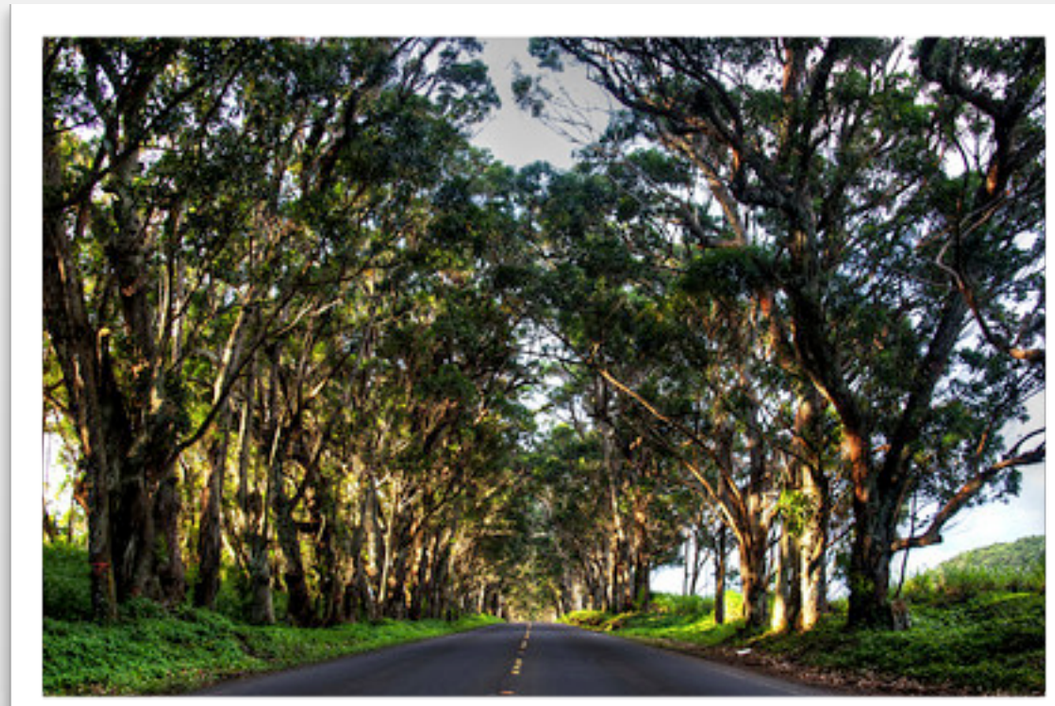
Use **JavaScript** to write your code

Access **Native Features**

© 2012 Raible Designs

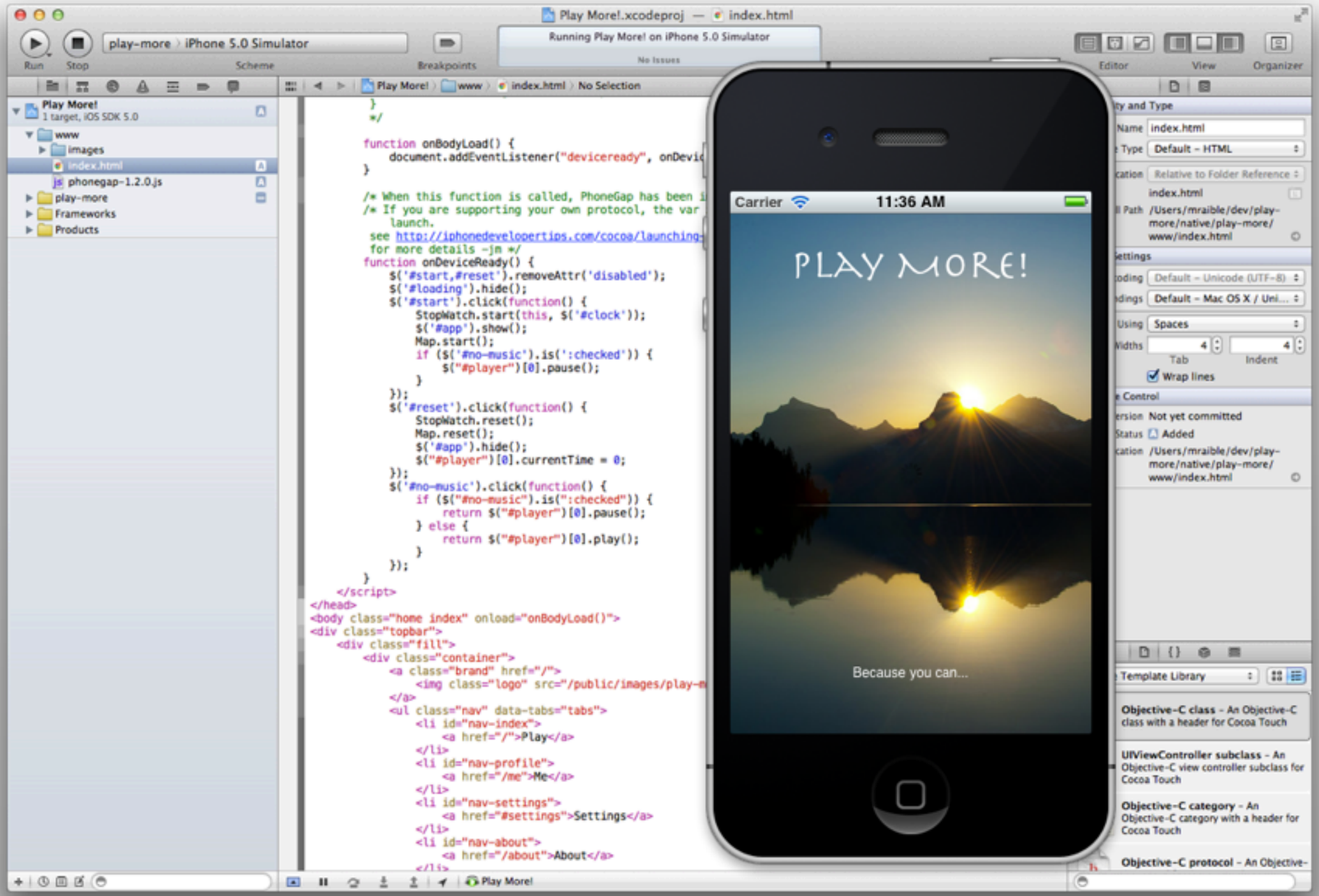
Requirements

- Intel-based computer with Mac OS X Snow Leopard (10.6)
- Xcode
- PhoneGap
- Necessary for Installation:
  - An Apple iOS Device
  - iOS Developer Certification



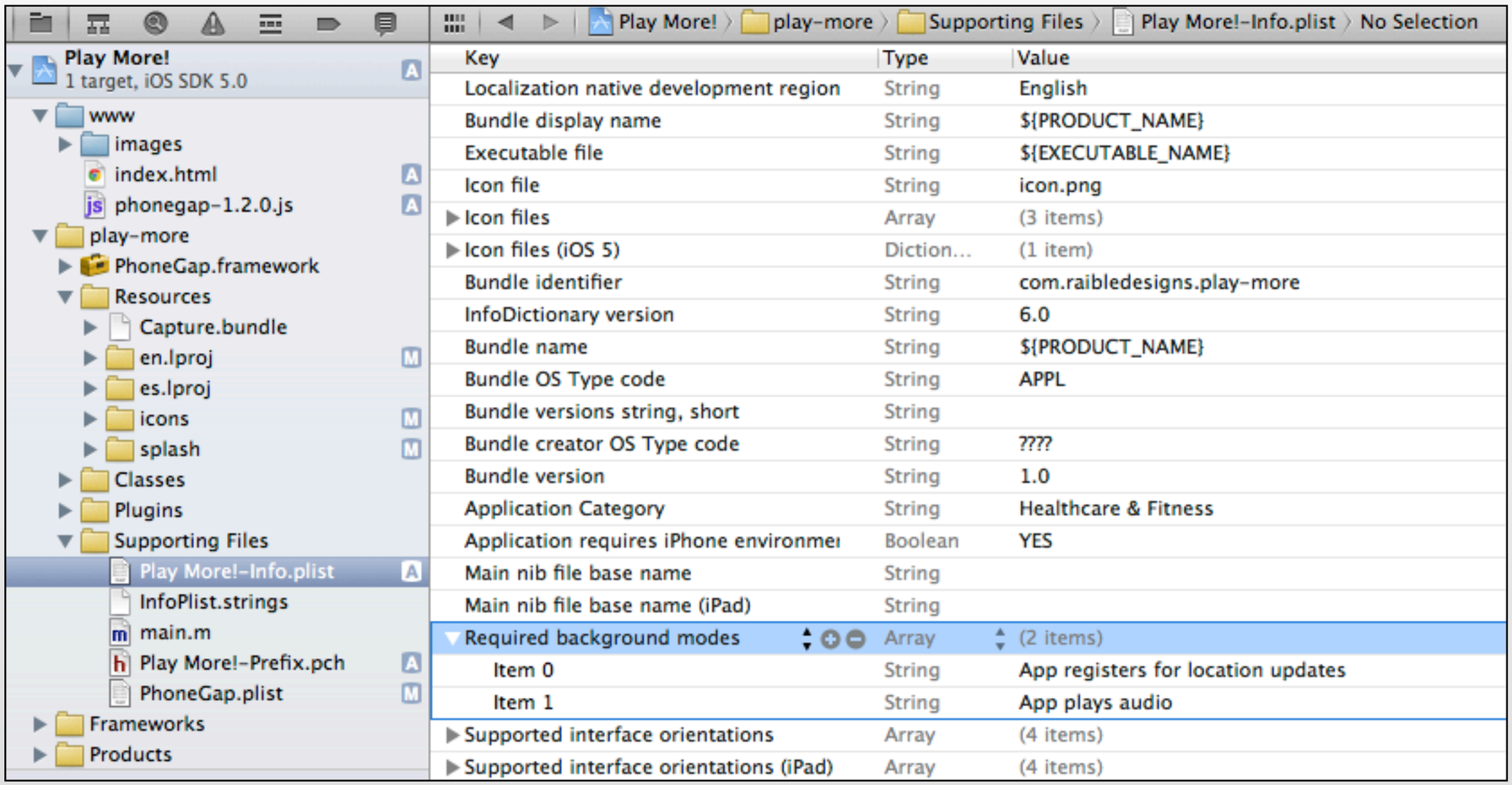
© 2012 Raible Designs

Icons and Splash Screen



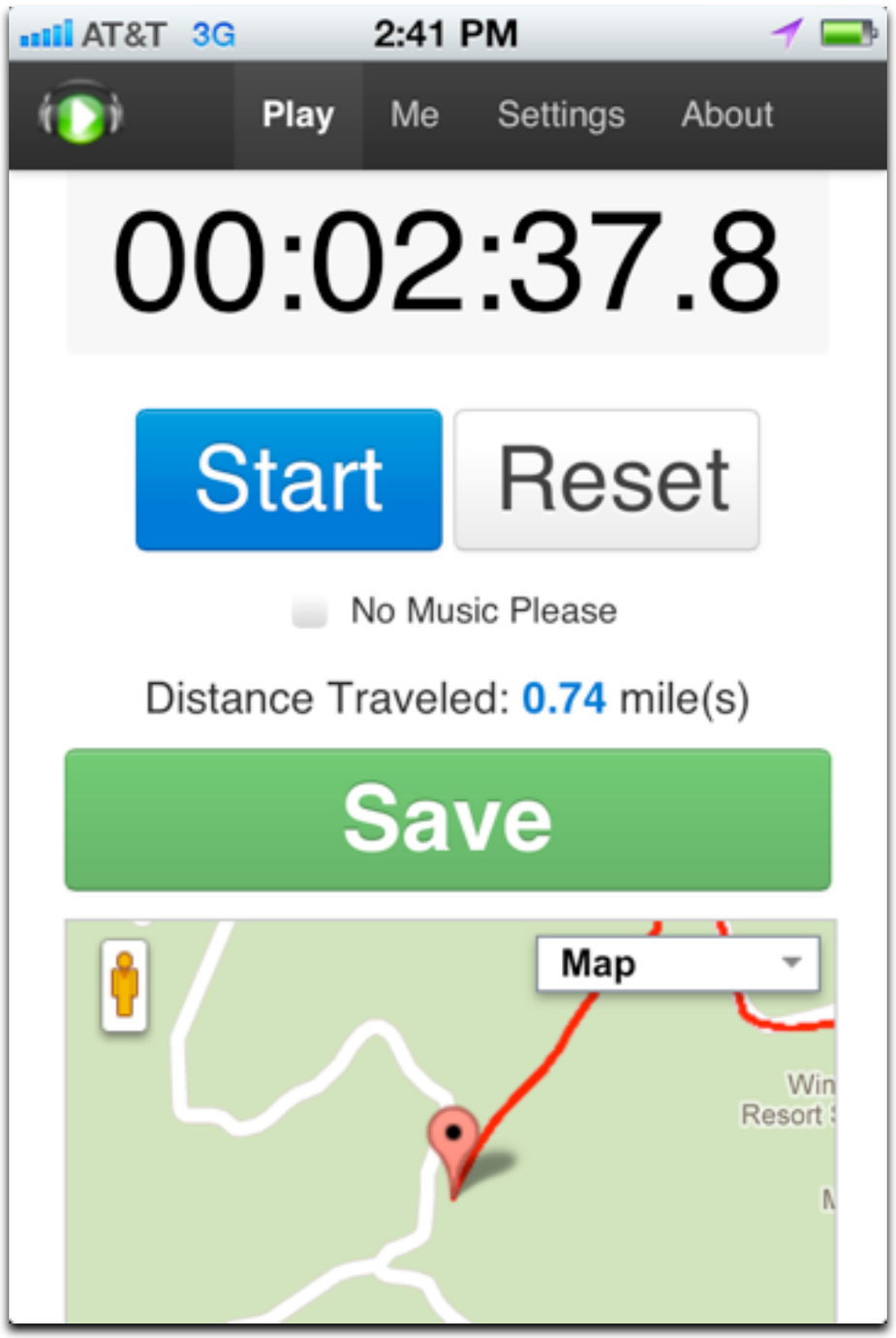
© 2012 Raible Designs

Background Modes



© 2012 Raible Designs

Success!



© 2012 Raible Designs

PhoneGap Writeup

Monday November 14, 2011

PhoneGap to the Rescue!

This is the 7th article in a series about my adventures developing a web application with HTML5, Play Scala, CoffeeScript and Jade. Previous articles can be found at:

1. **Integrating Scalate and Jade with Play 1.2.3**
2. **Trying to make CoffeeScript work with Scalate and Play**
3. **Integrating HTML5 Boilerplate with Scalate and Play**
4. **Developing with HTML5, CoffeeScript and Twitter's Bootstrap**
5. **Play Scala's Anorm, Heroku and PostgreSQL Issues**
6. **More Scalate Goodness for Play**

A few weeks ago, I wrote about **Developing a Stopwatch and Trip Meter with HTML5**. I mentioned I'd run into a major issue when I discovered HTML5 Geo's `watchPosition()` feature didn't run in the background. From that article:

> I tried out the trip meter that night evening on a bike ride and noticed it said I'd traveled 3 miles when I'd really gone 6. I quickly figured out it was only calculating start point to end point and not taking into account all the turns in between. To view what was happening, I integrated my odometer.coffee with my map using **Google Maps Polylines**. Upon finishing the integration, I discovered two things, 1) HTML5 geolocation was highly inaccurate and 2) **geolocation doesn't run in the background**.

At the time, I opted to ignore this issue and use my app by setting Auto-Lock to never. This worked, but if I happened to bump my phone while exercising, the app would get closed. Not to mention it really drained the battery and seemed to crash every-so-often.

http://raibledesigns.com/rd/entry/phonegap_to_the_rescue

© 2012 Raible Designs

Was it worth it?

- Development Hours: $$$
- play-more.com domain: $180
- GoPro Helmet Cam: $239
- iOS Certified Developer: $100
- Free Trip to Devoxx: ***Priceless***

© 2012 Raible Designs

Since Devoxx

- Tried to upgrade to Play 2.0
- Integrated RESTful Services
- Integrated Secure Social for Authentication
- Added ability to save, edit and delete workouts

© 2012 Raible Designs

Upgrading to Play 2.0

```
import play.jobs._
import play.Play

@OnApplicationStart
class BootStrap extends Job {

  override def doJob() {

    import models._
    import play.test._

    // Import initial data if the database is
    if (Athlete.count().single() == 0) {
      Yaml[List[Any]]("initial-data.yml").fore
        _ match {
          case a: Athlete => Athlete.create(a)
          case w: Workout => Workout.create(w)
          case c: Comment => Comment.create(c)
        }
      }
    }
  }
}
```

```
import play.mvc.{Scope, Http}

trait Scalate {

  def render(args: (Symbol, Any)*) = {
    var template = Scope.RenderArgs.current().get("template")
    if (template == null) {
      template = Http.Request.current().action.replace(".", "/")
    }

    renderTemplate(template.toString, args: _*);
  }

  def renderTemplate(template: String, args: (Symbol, Any)*) = {
    ScalateTemplate(template).render(args: _*);
```

[2.0][scala] Anyone succeeded in running a Play20 + postgres server on Heroku?

12 messages - Collapse all - Report discussion as spam

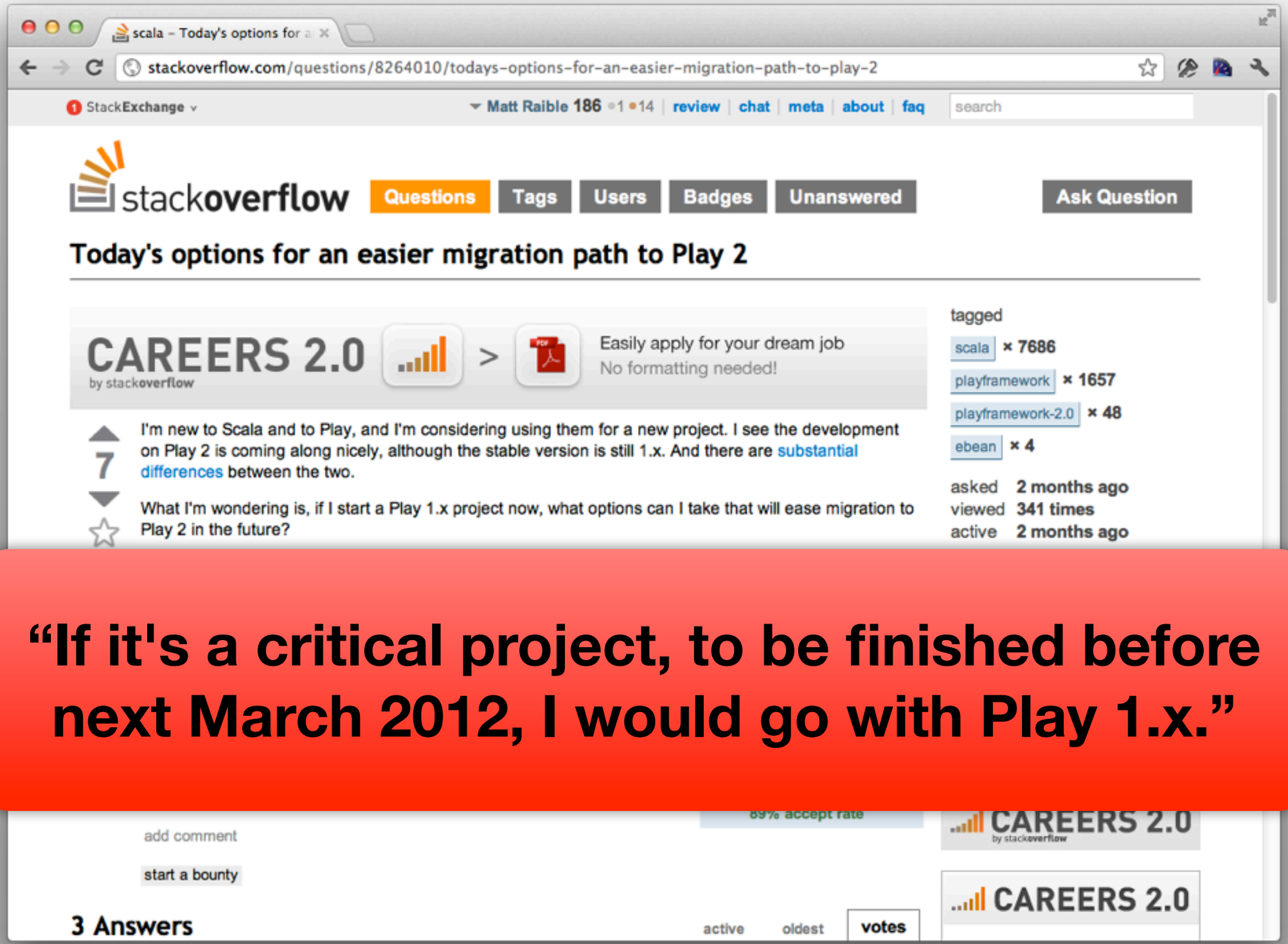
Pascal Voitot Dev View profile

Hello,
I can deploy on Heroku and run the app but I get the following error:

2012-01-31T09:38:58+00:00 app[web.1]: Caused by: java.sql.SQLException: No suitable driver found for postgres://rsfrdzpvpe:pR1XloicSbtp-dbT4...@ec2-107-21-110-231.compute-1.amazonaws.com/rsfrdzpvpe
2012-01-31T09:38:58+00:00 app[web.1]: at play.core.server.NettyServer.main(NettyServer.scala)
2012-01-31T09:38:58+00:00 app[web.1]: at

© 2012 Raible Designs

Upgrading to Play 2.0



© 2012 Raible Designs

JSON Services

```
package controllers.api

import play.mvc.Controller
import models._
import com.codahale.jerkson.Json._

object WorkoutService extends Controller {

  def workouts = {
    response.setContentTypeIfNotSet("application/json")
    generate(Workout.find().list())
  }

  def edit(id: Long) = {
    generate(Workout.byIdWithAthleteAndComments(id))
  }

  def create() = {
    var workout = params.get("workout", classOf[Workout])
    Workout.create(workout)
  }

  def save(id: Option[Long]) = {
    var workout = params.get("workout", classOf[Workout])
    Workout.update(workout)
  }

  def delete(id: Long) = {
    Workout.delete("id={id}").on("id" -> id).executeUpdate()
  }

}
```

© 2012 Raible Designs

API Tests

```
import play.test.FunctionalTest
import play.test.FunctionalTest._
import org.junit._

class ApiTests extends FunctionalTest {

    @Test
    def testGetWorkouts() {
        var response = GET("/api/workouts");
        assertStatus(200, response);
        assertContentType("application/json", response)
        println(response.out)
    }

}
```

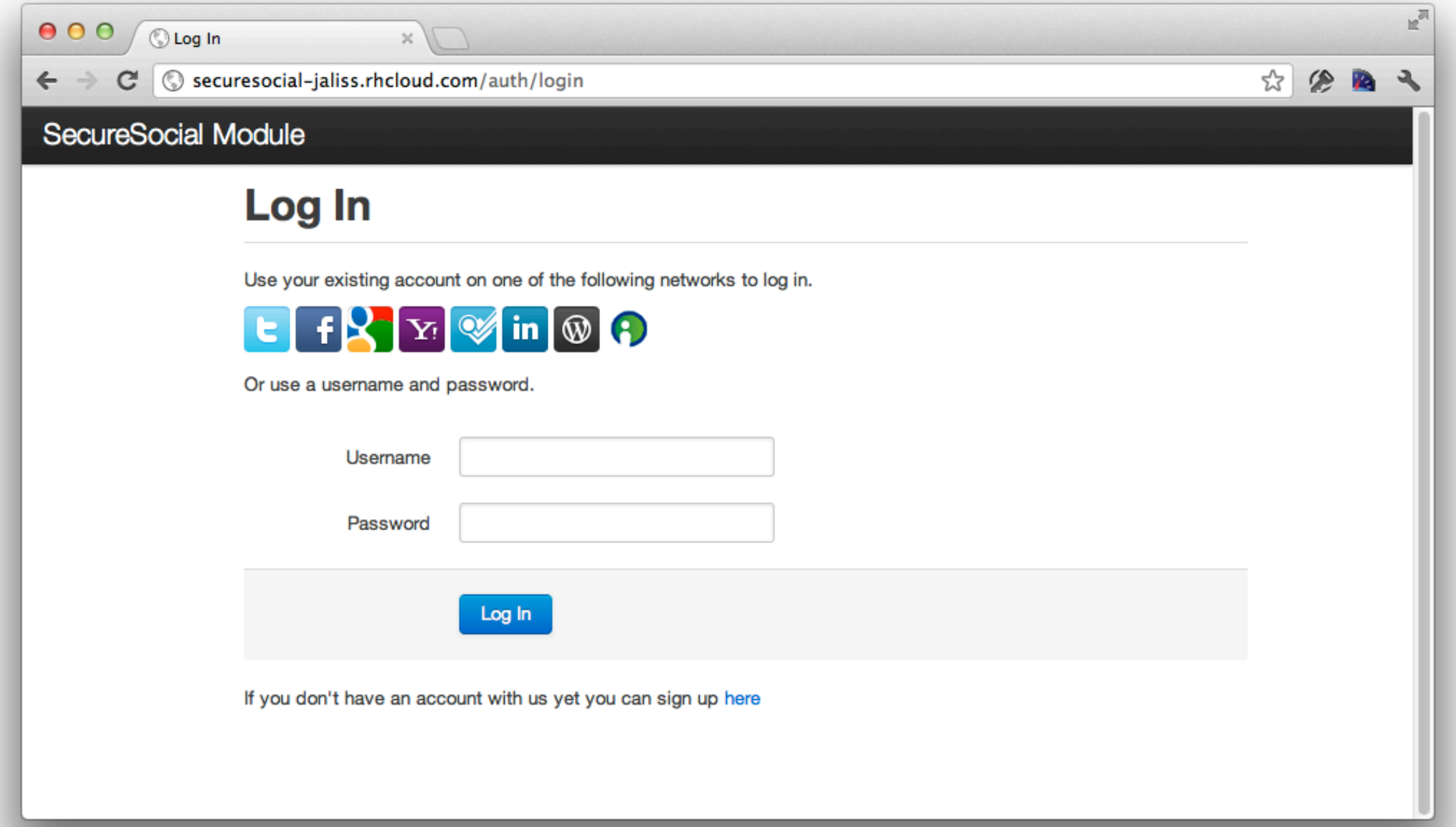
© 2012 Raible Designs

Secure Social

```scala
package controllers
import play.mvc._
import controllers.securesocial.SecureSocial

/*
 * @author Jorge Aliss <jaliss@gmail.com> of Secure Social fame.
 */
trait Secure {
    self: Controller =>

    @Before def checkAccess() {
        SecureSocial.DeadboltHelper.beforeRoleCheck()
    }

    def currentUser = {
        SecureSocial.getCurrentUser
    }
}
```

© 2012 Raible Designs

Secure Social



© 2012 Raible Designs

Secure Social

Sunday February 12, 2012

Secure JSON Services with Play Scala and SecureSocial

Last November, I **traveled to Antwerp to speak at Devoxx**. After my talk on HTML5 with Play Scala, **Mattias Karlsson** approached me and we had a chat about doing the same talk at **Jfokus** in Stockholm. I agreed and we began talking details after Trish and I returned to the US.

I wrote this article on a plane between Denver and Seattle and will be hopping over the North Pole to Stockholm via Iceland tonight. For the past couple of weeks, I've been updating my *Play More!* HTML5/mobile app to add some new features. Most notably, I wanted to upgrade to Play 2.0, create JSON services and add authentication.

Upgrading to Play 2.0

My attempt to upgrade to Play 2.0 involved **checking out the source from GitHub**, building and installing the RC1 snapshot. As I tried to upgrade my app and started getting failed imports, I turned to the internet (specifically StackOverflow) to **see if it was a good idea**. The first answer for that question suggested I stay with 1.x.

http://raibledesigns.com/rd/entry/secure_json_services_with_play

© 2012 Raible Designs

Was it worth it?

- Development Hours: $$$
- play-more.com domain: $180
- GoPro Helmet Cam: $239
- iOS Certified Developer: $100
- Free Trip to Jfokus: ***Awesome!***

© 2012 Raible Designs

Developing Play More

http://vimeo.com/36826202

© 2012 Raible Designs

Since Jfokus

- Upgraded to Play 2.0
- Rewrote iOS App
  - Integrated with jQTouch
  - Optimized for speed
  - Upgraded PhoneGap
  - Tested with iOS 6 Beta

© 2012 Raible Designs

Upgrading to Play 2

Tuesday June 05, 2012

Upgrading to Play 2: Anorm and Testing

This time last year, I decided I wanted to learn Scala. I chose the **Play Framework** as my vehicle for learning and I added **CoffeeScript** and **Jade** to the mix. I packaged it all up, learned a bunch and **presented it at Devoxx 2011**.

In January, I added SecureSocial, JSON Services and worked a bit on the mobile client. I presented my findings **at Jfokus** shortly after. As part of my aforementioned post, I wrote:

> Right before we left for Jfokus, I was able to get everything to work, but didn't spend as much time as I'd like working on the mobile client. If this talk gets accepted for Devoxx France, I plan on spending most of my time enhancing the mobile client.

I had some complications (a.k.a. too much **vacation**) with Devoxx France and wasn't able to attend. To make up for it, I submitted the talk to **ÜberConf**. It got accepted and I started working on my app a couple weeks ago. So far, I've spent about 8 hours upgrading it to Play 2 and I hope to start re-writing the mobile client later this week. I plan on using **Cordova**, **jQTouch** and releasing it in the App Store sometime this month.

Upgrading to Play 2

When I heard about Play 2, I thought it was a great thing. The developers were re-writing the framework to use Scala at the core and I was already using Scala in my app. Then I learned they were going to throw backwards compatibility out the window and I groaned. "Really? Another web framework (like Tapestry of old) screwing its users and making them learn everything again?!", I thought. "Maybe they should call it *Run* instead of *Play*, leaving the old framework that everyone loves intact."

However, after hearing about it at **Devoxx** and **Jfokus**, I figured I should at least *try* to migrate. I downloaded Play 2.0.1, created a new project and went to work.

http://raibledesigns.com/rd/entry/upgrading_to_play_2_anorm

© 2012 Raible Designs

Tools

git --distributed-is-the-new-centralized

heroku

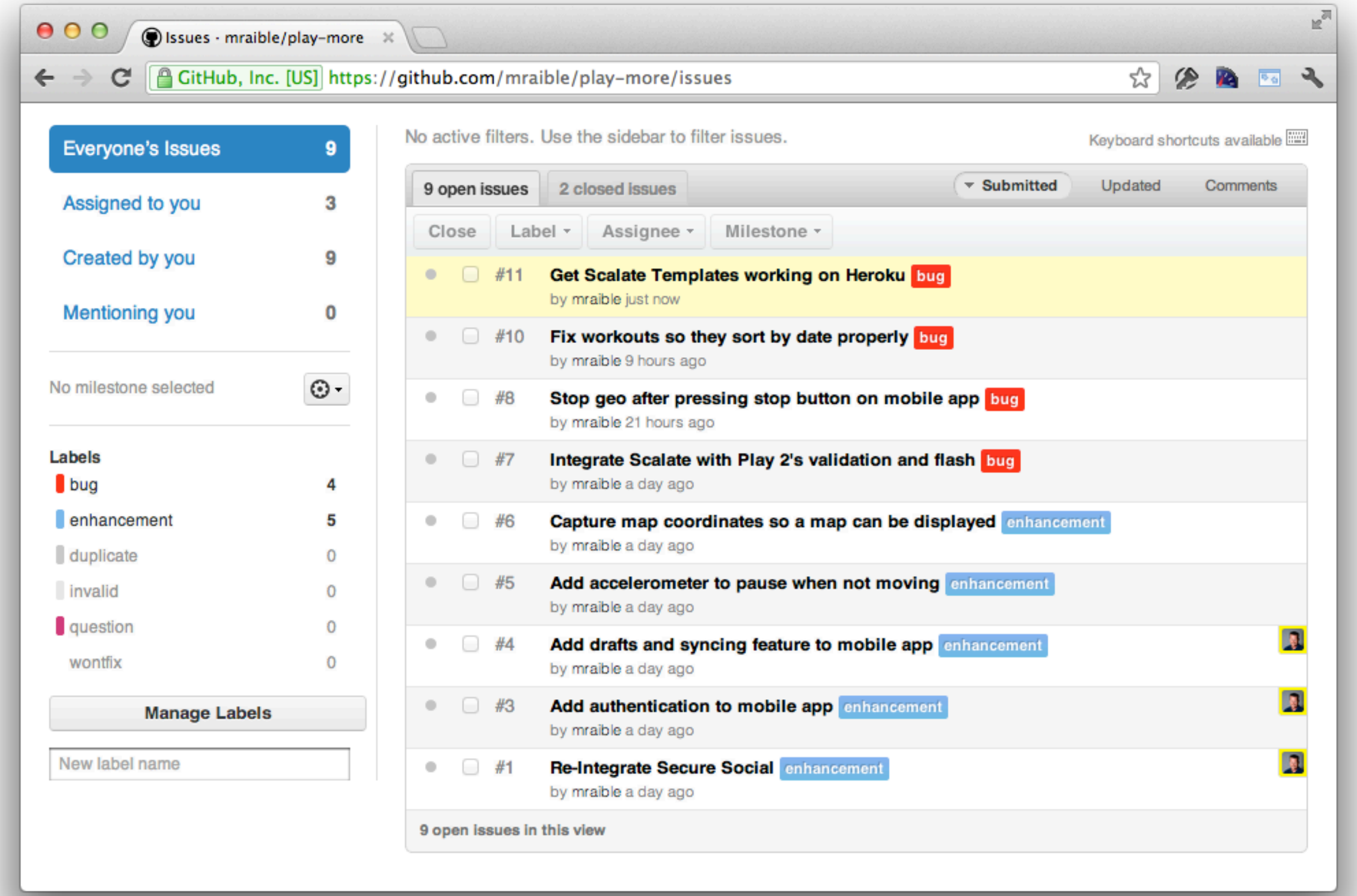
© 2012 Raible Designs

What didn't work?

- Wasn't able to get workouts sorting by date
- Didn't have time to implement Draft and Syncing
  - Local Storage or Web SQL Database?
- Geo doesn't stop after pressing stop button
- Scalate integration with errors and flash
- Secure Social doesn't work on Heroku
- Only Track When Moving not implemented

© 2012 Raible Designs

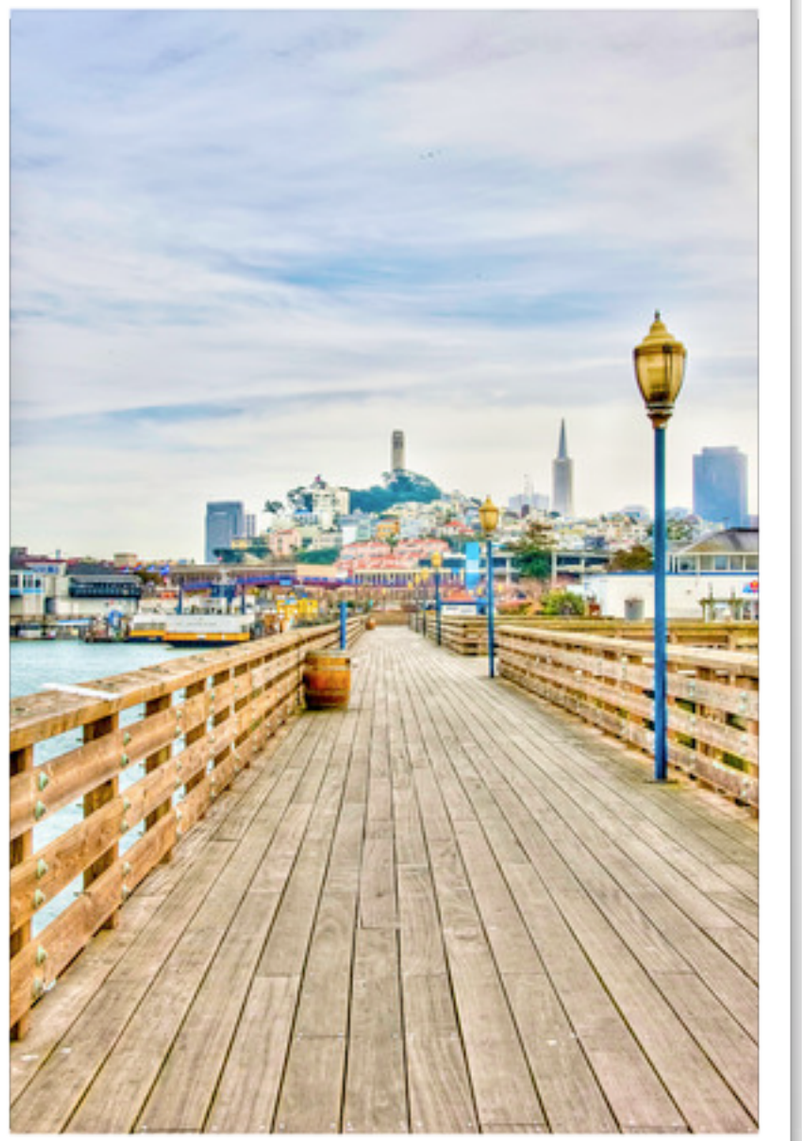
What didn't work?



© 2012 Raible Designs

Lessons Learned

- Develop Mobile Client first
- Develop Web Client as a one-page app
- Don’t rely on the internet for mobile
- Keep static assets local for faster startup
- Bleeding edge can be painful



© 2012 Raible Designs

Would I do it again?

- Learned more about Play and Scala than expected
- Play 1 was more productive than Play 2
- Anorm and Scalate were ***huge*** time sinks
- HTML5 and CoffeeScript were most enjoyable

© 2012 Raible Designs

HTML5 vs. Native

- If you need background services like geolocation or audio, native is necessary
- Can still write your apps in HTML5
  - Bridge the Gap with PhoneGap or Titanium
- If mobile is important, consider fully native with WebViews for common features - a.k.a. Hybrid

© 2012 Raible Designs

Questions?

Contact

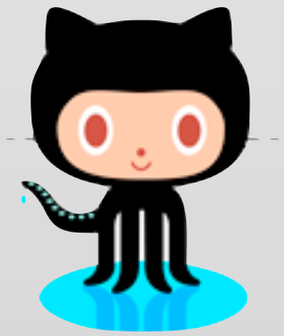
- http://raibledesigns.com
- @mraible

Presentation

- http://slideshare.net/mraible

Source

- https://github.com/mraible/play-more



© 2012 Raible Designs

Play More!

- Learn Something New
- Have Fun
- Get out there and *Play!*