No More Ugly

Using Twitter Bootstrap with Grails to make beautiful applications
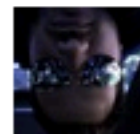
Twitter Bootstrap

- v1.0 - August 2011

- v2.0 - January 2012

- From the Twitter design team
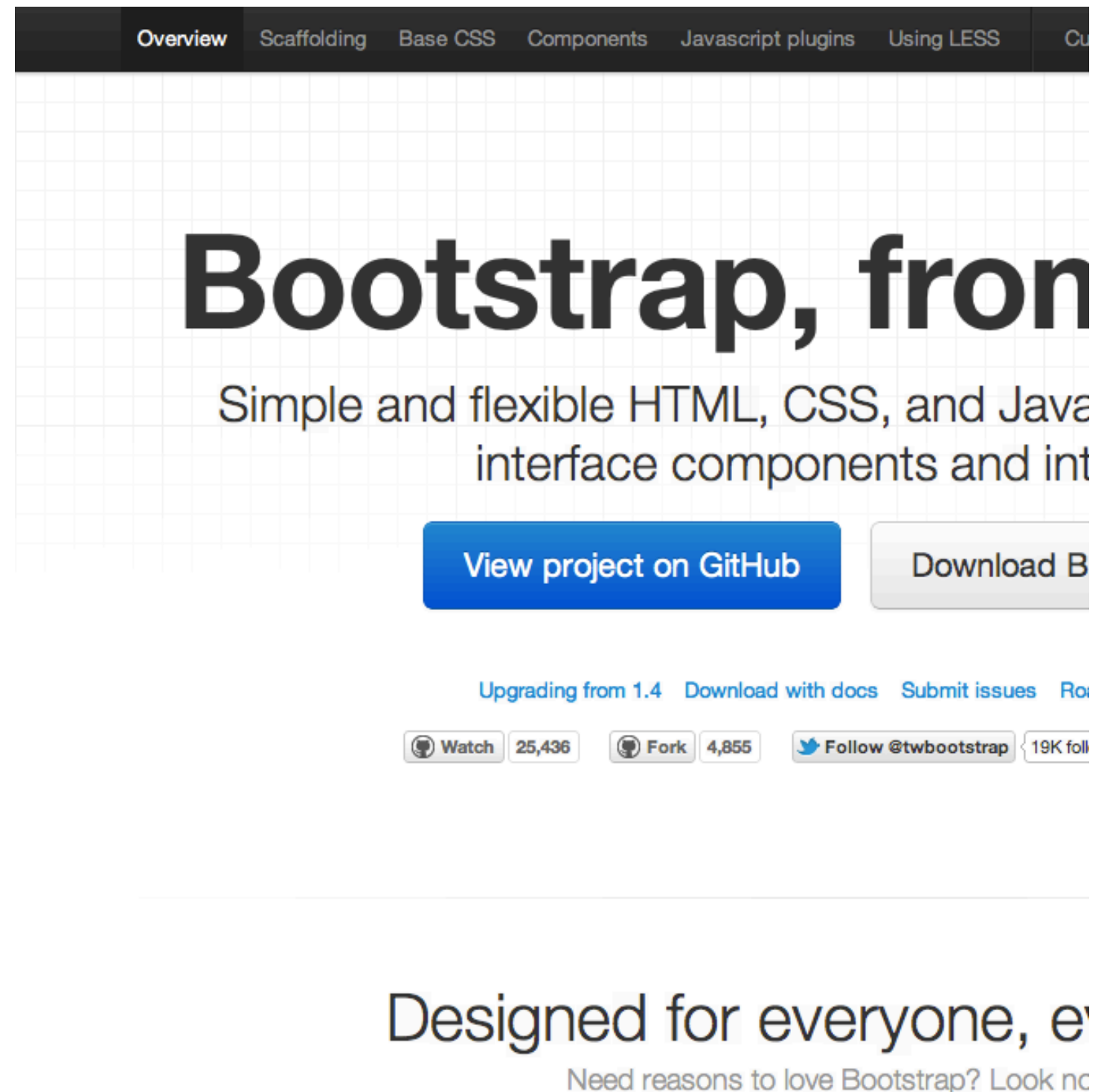
Best practices in a can

- CSS Framework

- Grid System

- Widgets

- JavaScript Coolness

- Based on LESS

- Hosted on GitHub

Customize to fit

- Full custom download

- Select components

- Set colors

- Minified files

Shrink to fit

- Responsive design

- One page, many devices

LESS is more

- http://lesscss.org/
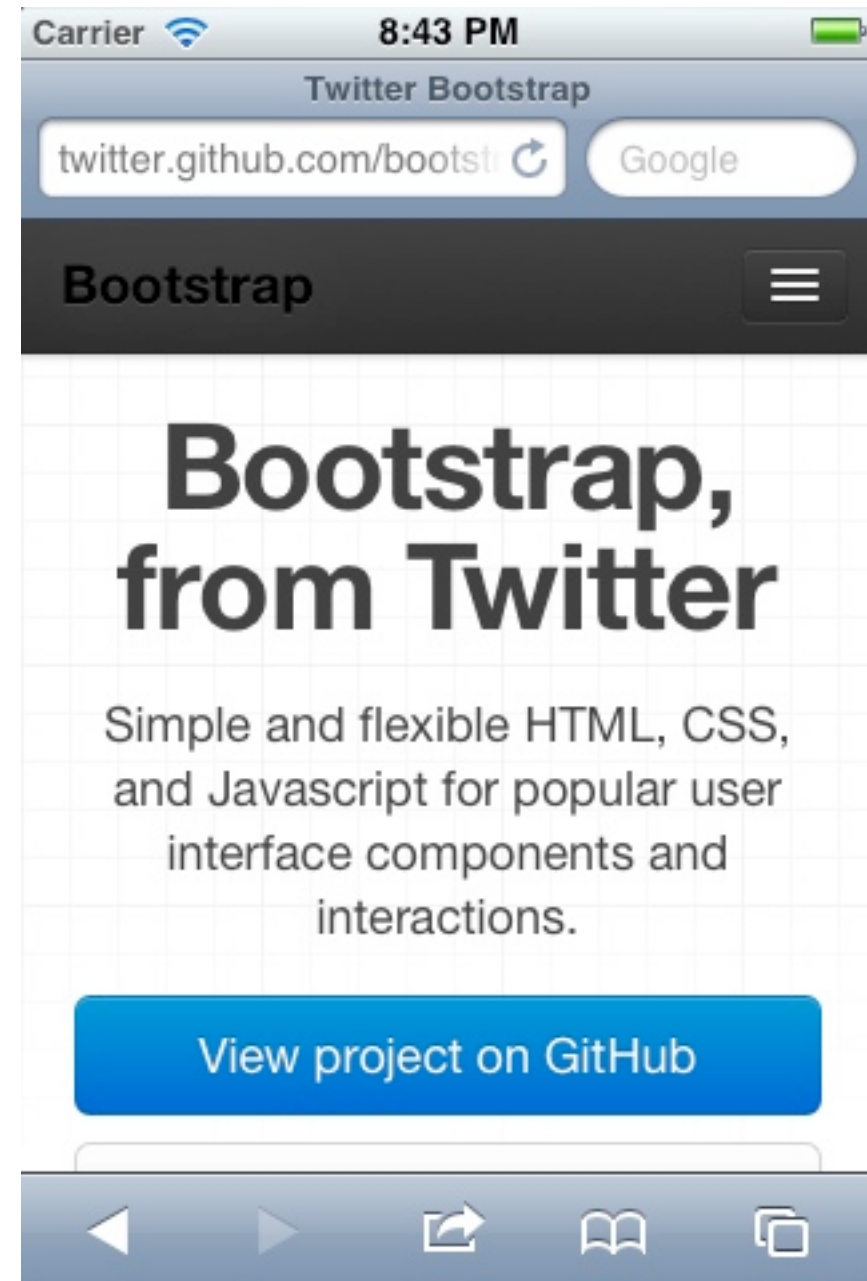
- Meta-language on top of CSS

- Feels "code-y"

- Client or Server Side compile
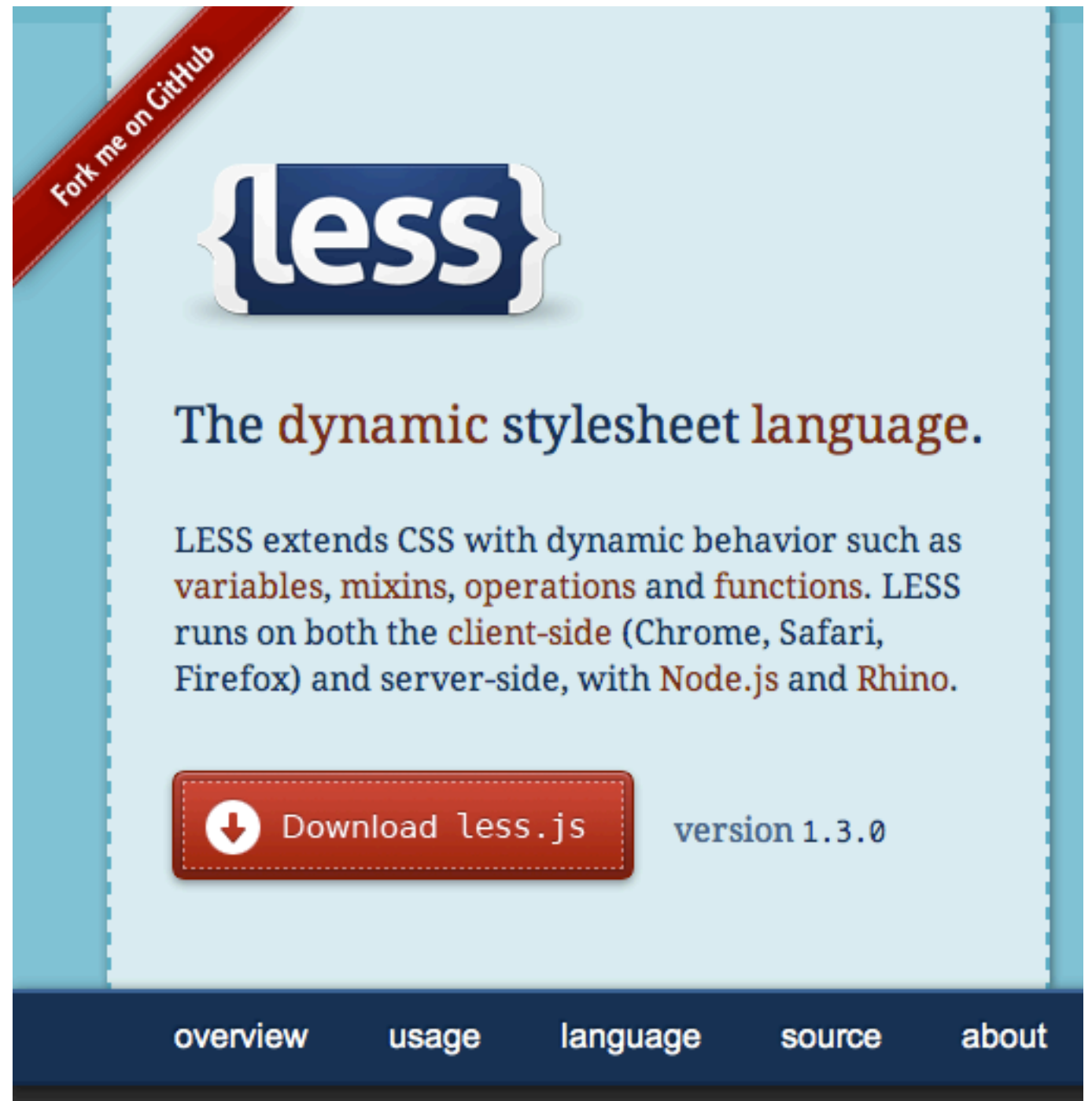
Mixing it up

```
// LESS

@color: #4D926F;

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

```
/* Compiled CSS */

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

```
// LESS

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

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

```
/* 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;
}
```

Getting on the grid

| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| 4 | 4 | 4 |

| 4 | 8 |

| 6 | 6 |

| 12 |
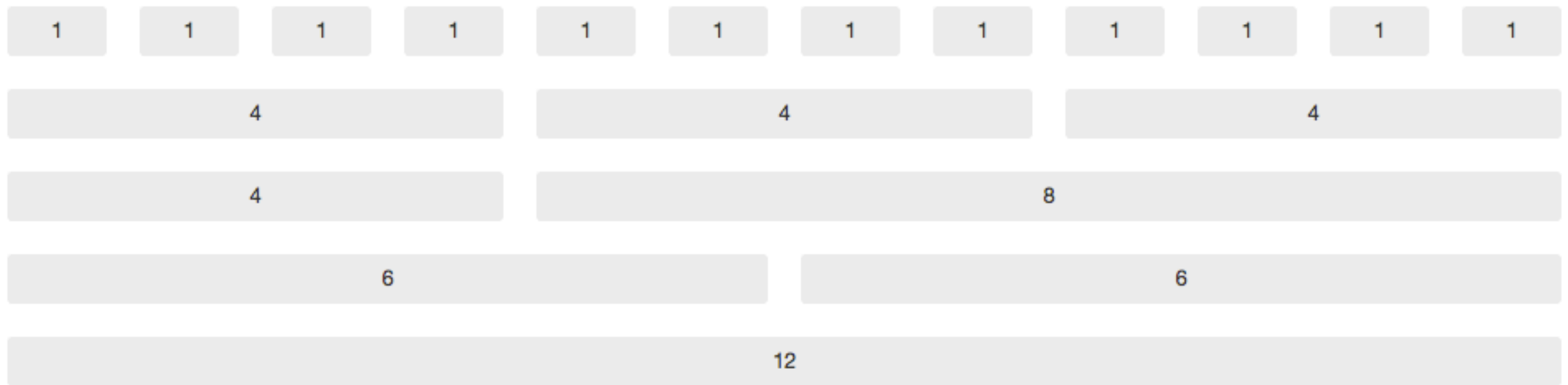
The default grid system provided as part of Bootstrap is a **940px-wide, 12-column grid**.

It also has four responsive variations for various devices and resolutions: phone, tablet portrait, table landscape and small desktops, and large widescreen desktops.

```
1.  <div class="row">
2.    <div class="span4">...</div>
3.    <div class="span8">...</div>
4.  </div>
```

As shown here, a basic layout can be created with two "columns," each spanning a number of the 12 foundational columns we defined as part of our grid system.

Layout made easy

Fixed layout

The default and simple 940px-wide, centered layout for just about any website or page provided by a single `<div class="container">`.
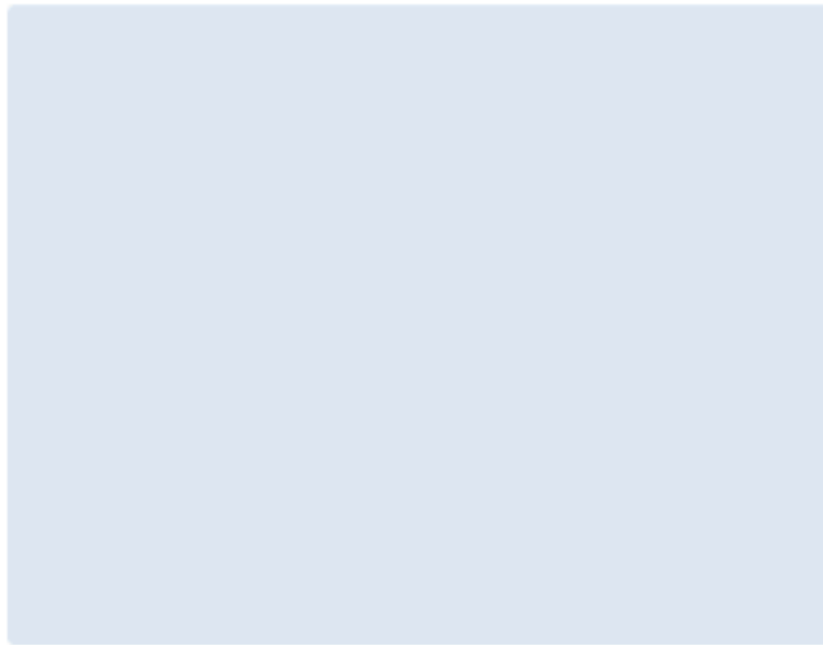
```
1.  <body>
2.    <div class="container">
3.      ...
4.    </div>
5.  </body>
```

Fluid layout

`<div class="container-fluid">` gives flexible page structure, min- and max-widths, and a left-hand sidebar. It's great for apps and docs.
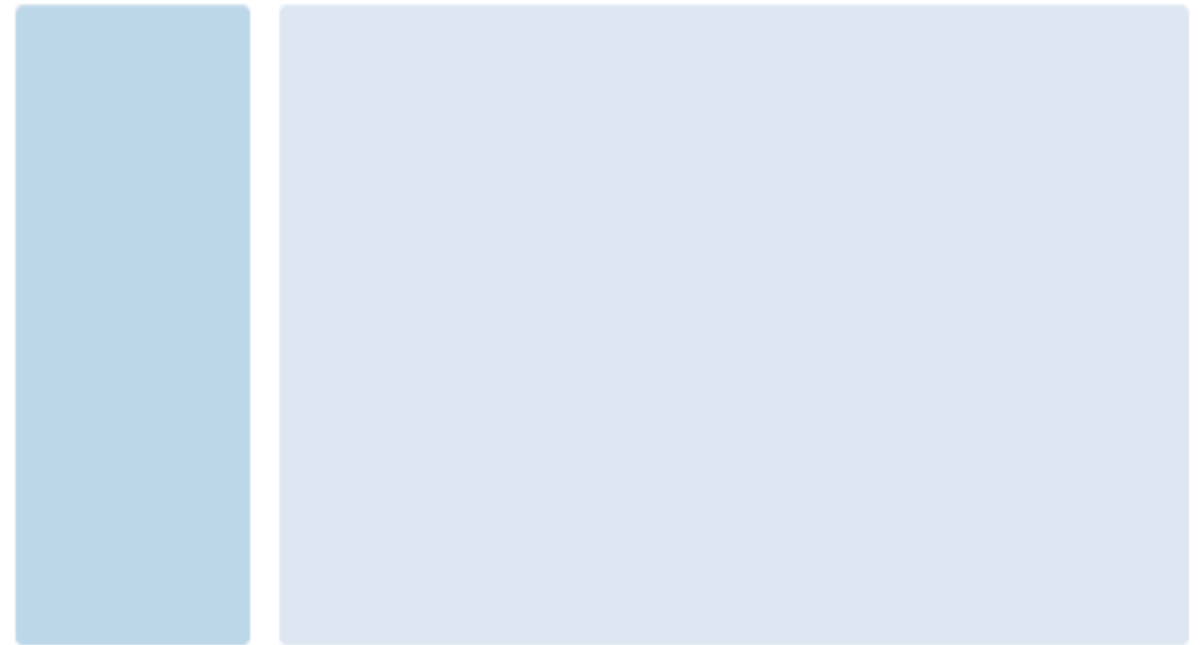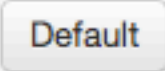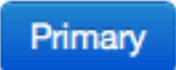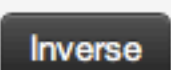
```
1.  <div class="container-fluid">
2.    <div class="row-fluid">
3.      <div class="span2">
4.        <!--Sidebar content-->
5.      </div>
6.      <div class="span10">
7.        <!--Body content-->
8.      </div>
9.    </div>
10. </div>
```

Typography for dummies

- Bootstrap provides correctly formatted versions of all key HTML elements

  - Headers (h1 - h6)

  - strong, em, address, abbr, block quotes

  - Lists : ul, ol, li

  - code and pre

  - tables and forms

For the love of buttons!

Button	class=""	Description
Default	`btn`	Standard gray button with gradient
Primary	`btn btn-primary`	Provides extra visual weight and identifies the primary action in a set of buttons
Info	`btn btn-info`	Used as an alternate to the default styles
Success	`btn btn-success`	Indicates a successful or positive action
Warning	`btn btn-warning`	Indicates caution should be taken with this action
Danger	`btn btn-danger`	Indicates a dangerous or potentially negative action
Inverse	`btn btn-inverse`	Alternate dark gray button, not tied to a semantic action or use
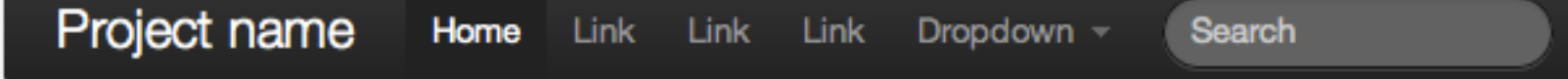
- Add a css class to any existing button or anchor tag

- Also available in other sizes and with icons
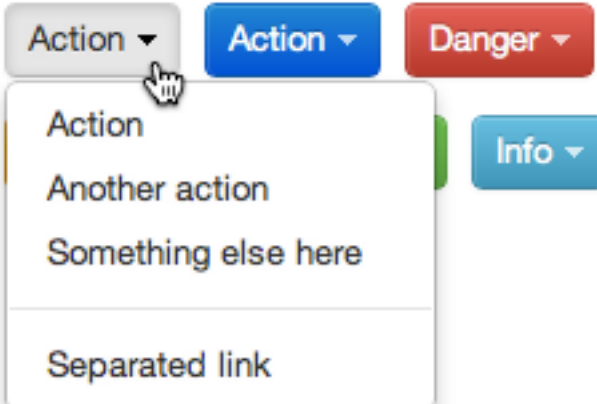
Parts are parts

- Bootstrap provides most all typical widgets via CSS and JS plugins

- Button Groups :

- Tabs:

- Navbar:

- Dropdown Buttons:

Adding to Grails

- Download Bootstrap and extract the zip file

- Copy assets to corresponding directories under web-app

Name ▲
bootstrap-responsive.css
bootstrap-responsive.min.css
bootstrap.css
bootstrap.min.css

Name ▲
glyphicons-halflings-white.png
glyphicons-halflings.png

Name ▲
bootstrap.js
bootstrap.min.js

Configure ApplicationResources.groovy
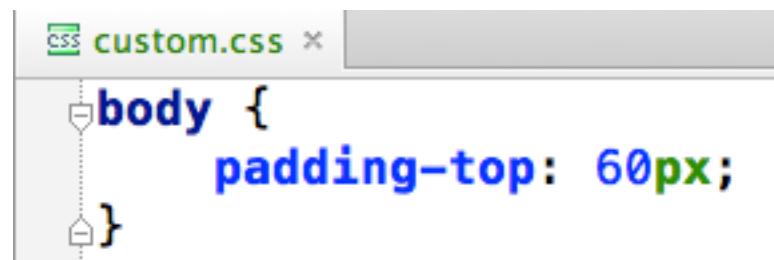
```groovy
ApplicationResources.groovy ×
modules = {

    bootstrap {
        dependsOn 'jquery'
        resource url: 'js/bootstrap.js'
        resource url: 'css/bootstrap.css'
        resource url: 'css/bootstrap-responsive.css'
        resource url: 'css/custom.css'
    }

    overrides {
        jquery {
            defaultBundle 'bootstrap'
        }
    }
}
```

Create custom.css

- Call it what you want, but you'll need it

- Add padding to the body so things don't mush together

- 40px or 60px are the norms

```css
css custom.css ×
body {
    padding-top: 60px;
}
```
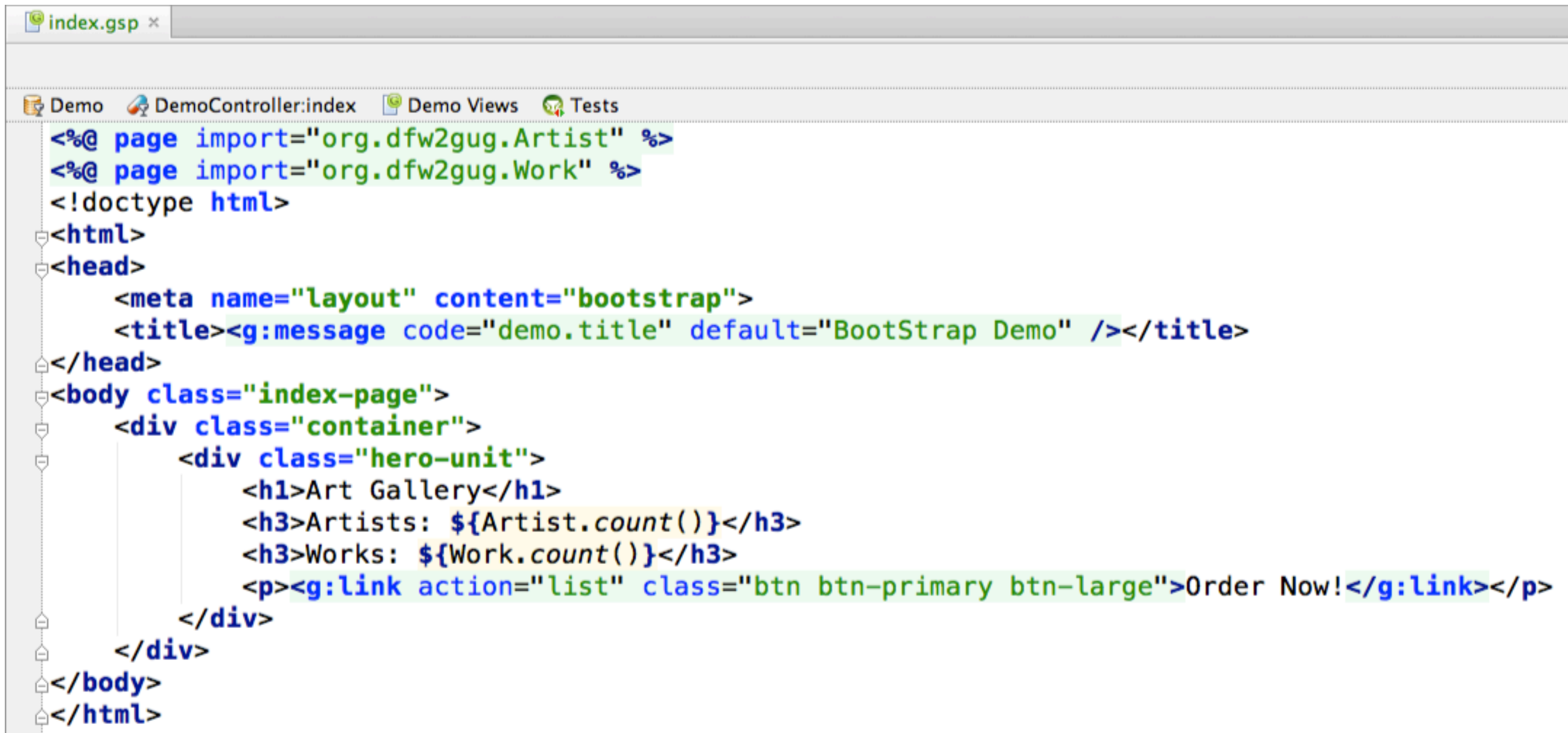
Update images URL in bootstrap.css

```css
[class^="icon-"],
[class*=" icon-"] {
  display: inline-block;
  width: 14px;
  height: 14px;
  line-height: 14px;
  vertical-align: text-top;
  background-image: url("../img/glyphicons-halflings.png");
  background-position: 14px 14px;
  background-repeat: no-repeat;
  *margin-right: .3em;
}
[class^="icon-"]:last-child,
[class*=" icon-"]:last-child {
  *margin-left: 0;
}
.icon-white {
  background-image: url("../img/glyphicons-halflings-white.png");
}
```

Create the bootstrap.gsp layout

```html
bootstrap.gsp ×

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title><g:layoutTitle default="DFW Groovy Grails"/></title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <r:require module="bootstrap" />
    <g:layoutHead/>
    <r:layoutResources />
</head>
<body class="${pageProperty(name: 'body.class')}">

<div class="navbar navbar-fixed-top">
    <div class="navbar-inner">
        <div class="container">
            <g:link action="index" class="brand">Groovy!</g:link>
            <div class="nav-collapse">
                <ul class="nav">
                    <li id="homePage" class="active"><g:link controller="demo" action="index">Home</g:link></li>
                    <li id="tablePage"><g:link controller="demo" action="table">Table</g:link></li>
                    <li id="listPage"><g:link controller="demo" action="list">List</g:link></li>
                    <li id="formPage"><g:link controller="demo" action="form">Form</g:link></li>
                </ul>
            </div>
        </div>
    </div>
</div>

<g:layoutBody/>
<r:layoutResources />
</body>
</html>
```

Create a content page

```
index.gsp ×

Demo    DemoController:index    Demo Views    Tests
<%@ page import="org.dfw2gug.Artist" %>
<%@ page import="org.dfw2gug.Work" %>
<!doctype html>
<html>
<head>
    <meta name="layout" content="bootstrap">
    <title><g:message code="demo.title" default="BootStrap Demo" /></title>
</head>
<body class="index-page">
    <div class="container">
        <div class="hero-unit">
            <h1>Art Gallery</h1>
            <h3>Artists: ${Artist.count()}</h3>
            <h3>Works: ${Work.count()}</h3>
            <p><g:link action="list" class="btn btn-primary btn-large">Order Now!</g:link></p>
        </div>
    </div>
</body>
</html>
```

Hello Bootstrap!

Art Gallery

Artists: 10

Works: 41

Order Now!

The code...

Suggestions

- Forget about centering; embrace the grid

- Use the joda-time plugin

- Use the less-resource plugin

- Bootstrap plugin is available

- "Another Bootstrap site"

Resources

http://twitter.github.com/bootstrap/

http://www.alistapart.com/articles/building-twitter-bootstrap/

Bootswatch

News   Gallery   Preview

Boots

Free themes fo

Watch 650   Fork   Follow
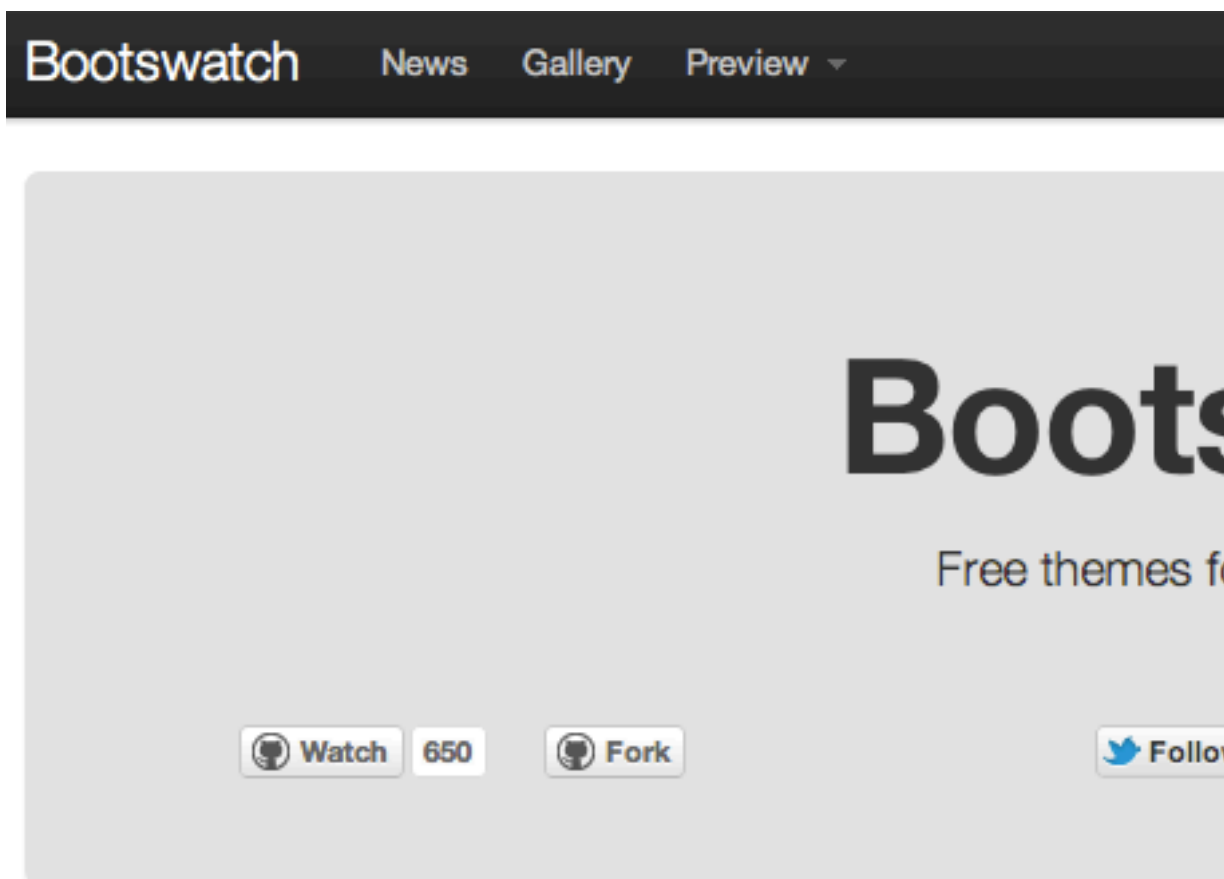
Easy to Install

Simply download the CSS file from the swatch of your choice and replace the one in Bootstrap. No messing around with hex values.

Whole New

We've all been there w buttons. See how a sp can transform the feel

Open Source

Licensed under Apache 2.0 and maintained by the community on GitHub.

Modular

Changes are contained enabling modification a compatibility.

win a free theme in our weekly raffle!

{wrap}bootstrap   Home   Themes   Start sellir

Themes for

Bootstrap is an HTML5 & CSS3 framework c sites. WrapBootstrap is a marketplace for pre and visitors while using a single, rock-solid fc

View all the themes »   Or make them! Sell your des

348 Followers   74 Followers   162 Subscribers

Hubble Business Theme

Hubble   HOME PORTFOLIO PRICING TABLES ABOUT BLOG CONTACT

http://bootswatch.com   http://wrapbootstrap.com

Thursday, April 5, 12

http://builtwithbootstrap.com

http://fortawesome.github.com/
Font-Awesome/

GORM Recipes
Tastes like chicken, acts like SQL

Project Setup getting the code

Overview

GORM recipes demonstrates how to effectively use the GORM persistence provider in a Grails 2.0 application. GORM makes it exceptionally easy to do the simple things, but for a new user, you can quickly hit a wall when you want to do something which would be simple to do in SQL. These recipes should help you get over that wall and help you resist that urge to drop to raw SQL to do things.

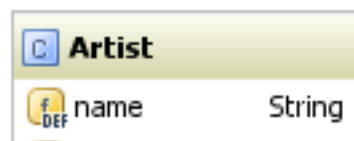The Grails 2.0 documentation should be a starting point for anyone jumping in to Grails. It is much improved over the 1.x branch and can probably answer your questions if you actually read it. The weakness is finding what you need or the right example of how a particular feature works. These recipes are an extension to the Querying with GORM section and are an attempt to help with these weaknesses.

The Git Stuff

These recipes use a simple Grails application hosted on GitHub as a testbed for the queries. The application defines two domain classes and populates them with sample data to query against. The two domain classes are in the files `Artist.groovy` and `Work.groovy`.

C Artist	
f DEF name	String

C Work	
f DEF title	String

http://timsporcic.github.com/GORM-Recipes/ | GORM Recipes

Thursday, April 5, 12

About Me

Tim Sporcic
Twitter: @TimSporcic
Blog: http://www.sporcic.org
Email: timsporcic@gmail.com
GitHub: https://github.com/timsporcic
LinkedIn: http://www.linkedin.com/in/timsporcic