Building Beautiful Grails Apps with Twitter Bootstrap

Presented by Jack Frosch

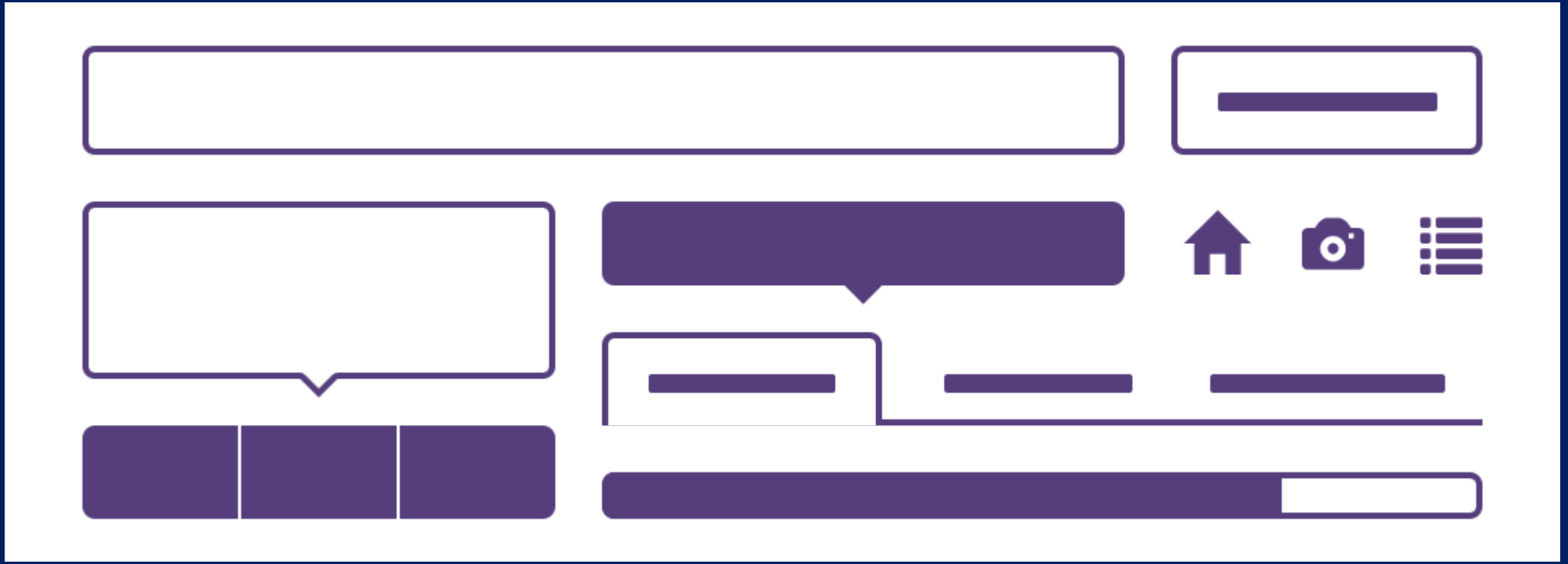
Wha's Up?

- Beautiful Bootstrap
- Start with Grails, Mix In Bootstrap
- Start with Bootstrap, Mix In Grails

Beautiful Bootstrap

- What's Bootstrap?
- Getting Bootstrap
- Bootstrap's approach to layout
- Components
- Mobile-ready with built-in responsiveness

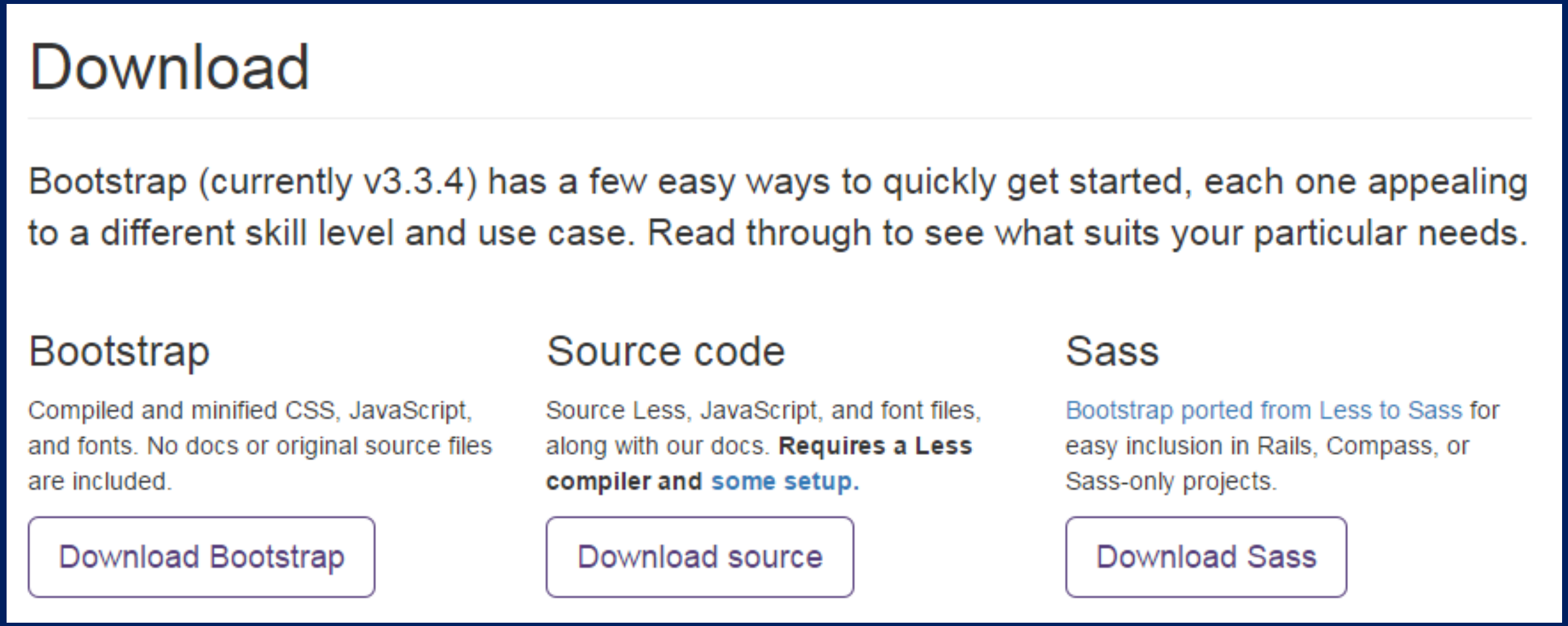
Bootstrap is a HTML/CSS/JS framework donated by Twitter
It's one framework for rendering on every device
i.e. You get built-in responsiveness on mobile devices



Bootstrap is loaded with features including Components, Glyph Icons, Behaviors

The cool thing is, almost all of these are rendered by simply applying CSS classes

Bootstrap ships with basic CSS, but supports preprocessors



If you want to download Bootstrap, you can:

http://getbootstrap.com/getting-started/#download

Bootstrap CDN

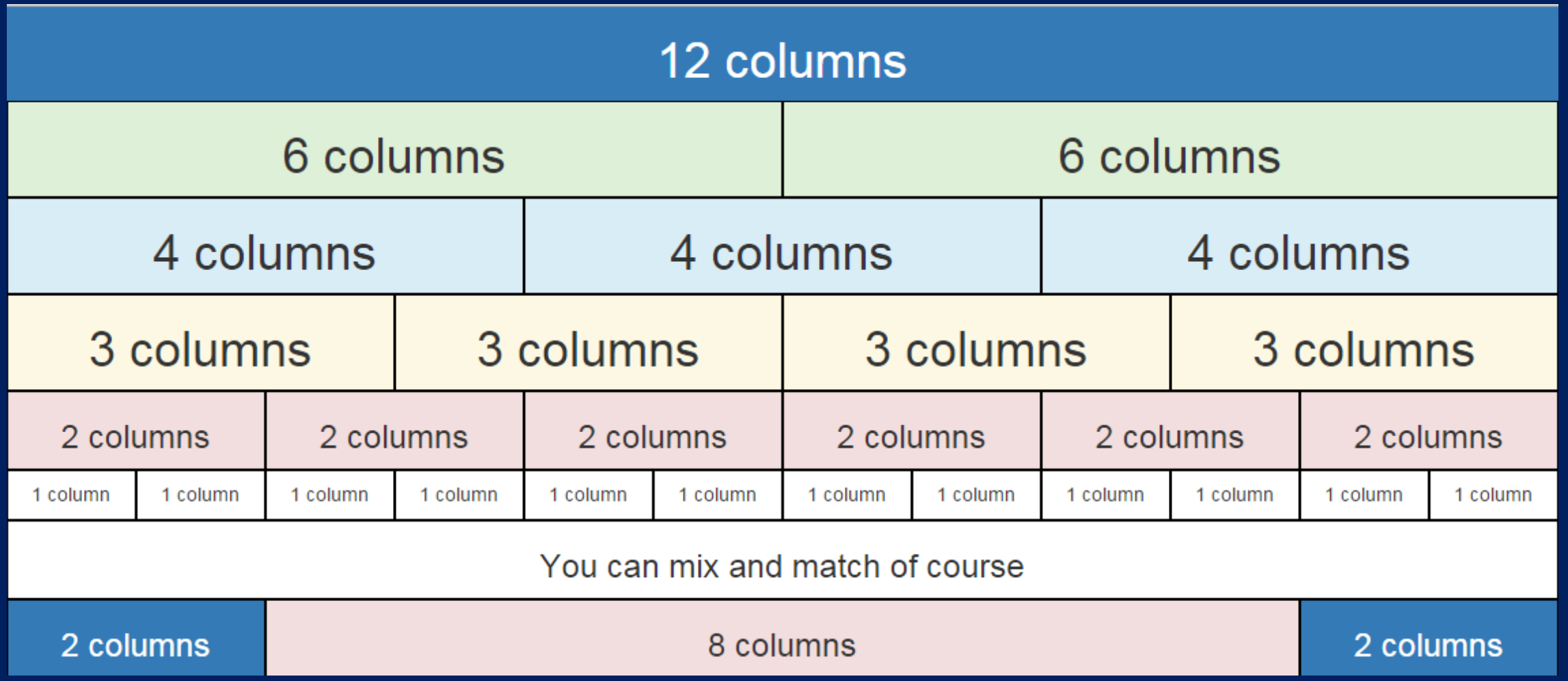
The folks over at MaxCDN graciously provide CDN support for Bootstrap's CSS and JavaScript. Just use these Bootstrap CDN links.

Copy

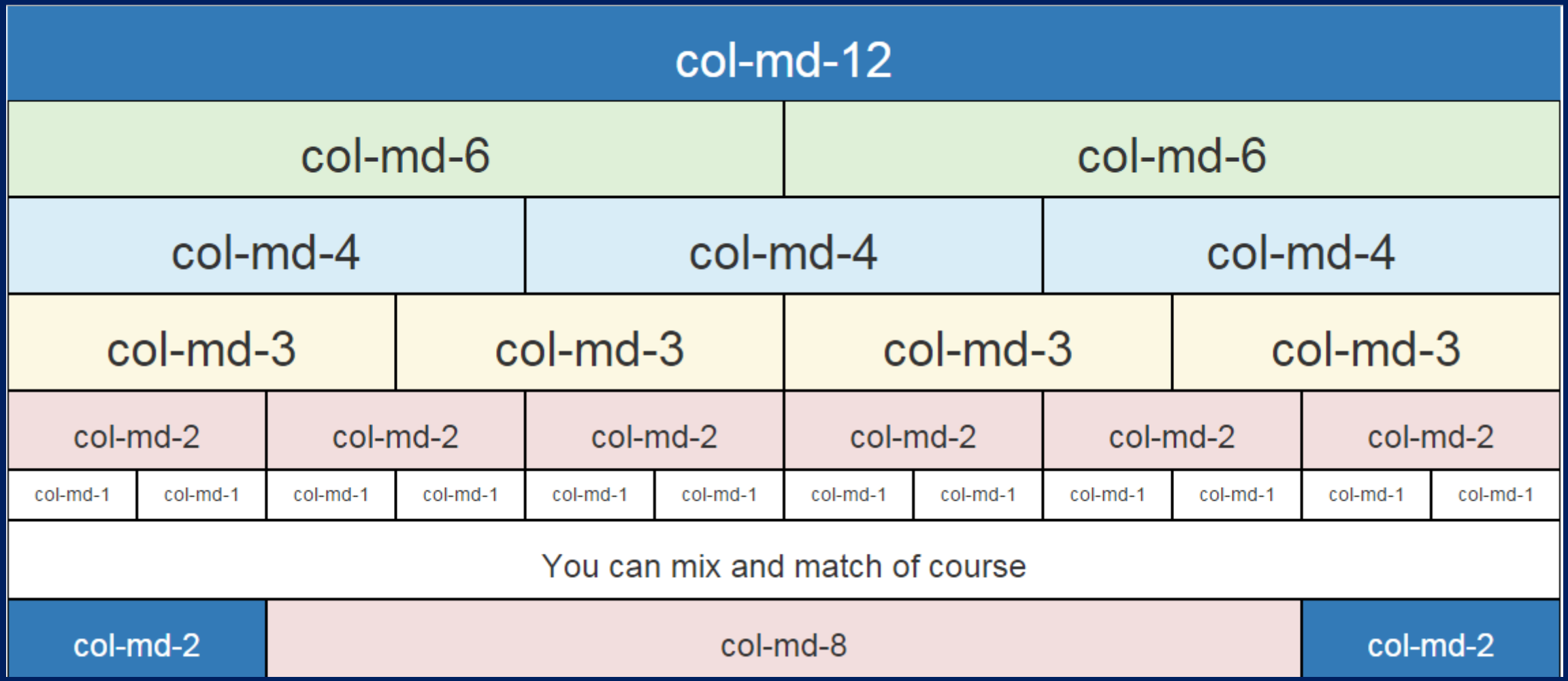
```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css">

<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap-theme.min.css">

<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"></script>
```

You can also serve it fresh from a CDN

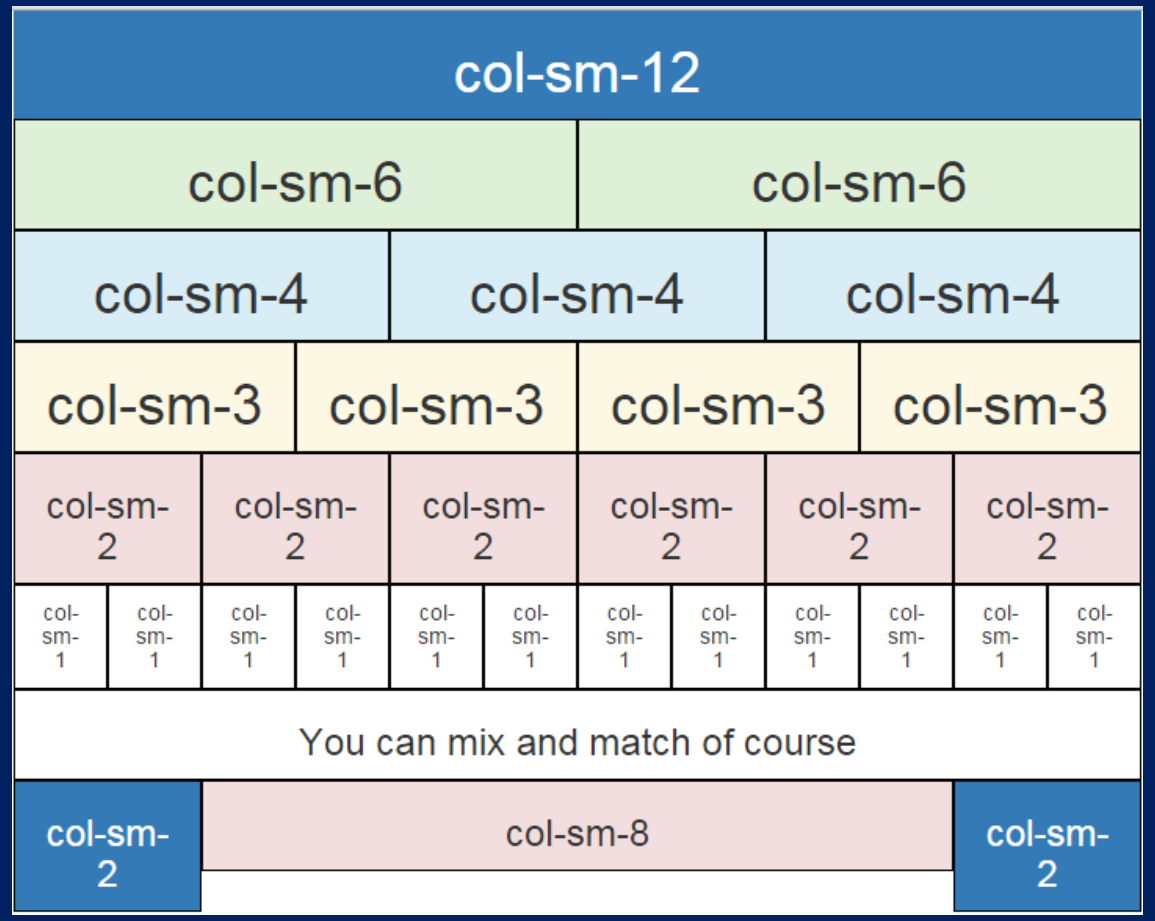
(As we'll see, we can also load it with a Grails plugin)

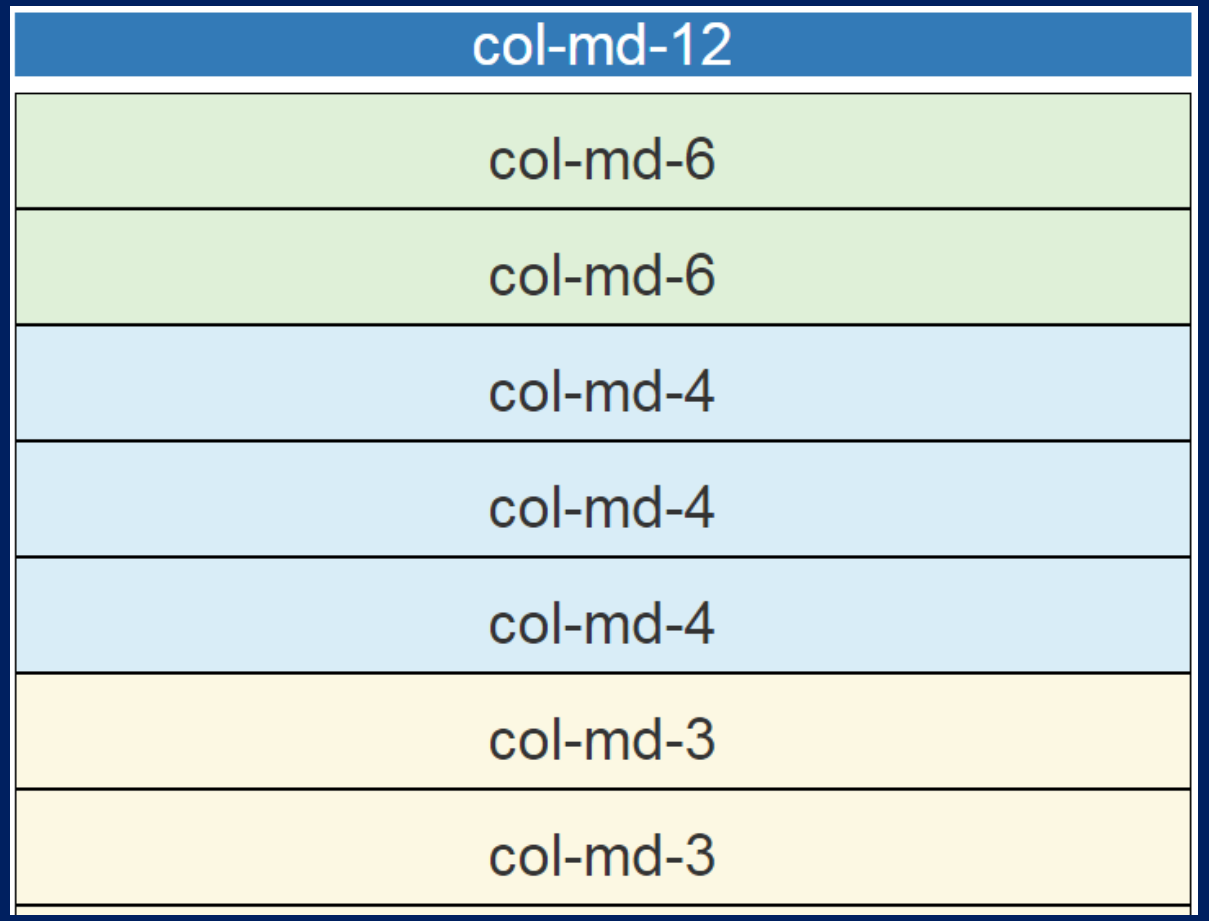


Bootstrap layout is based on 12 columns



Number of columns dictated by CSS class

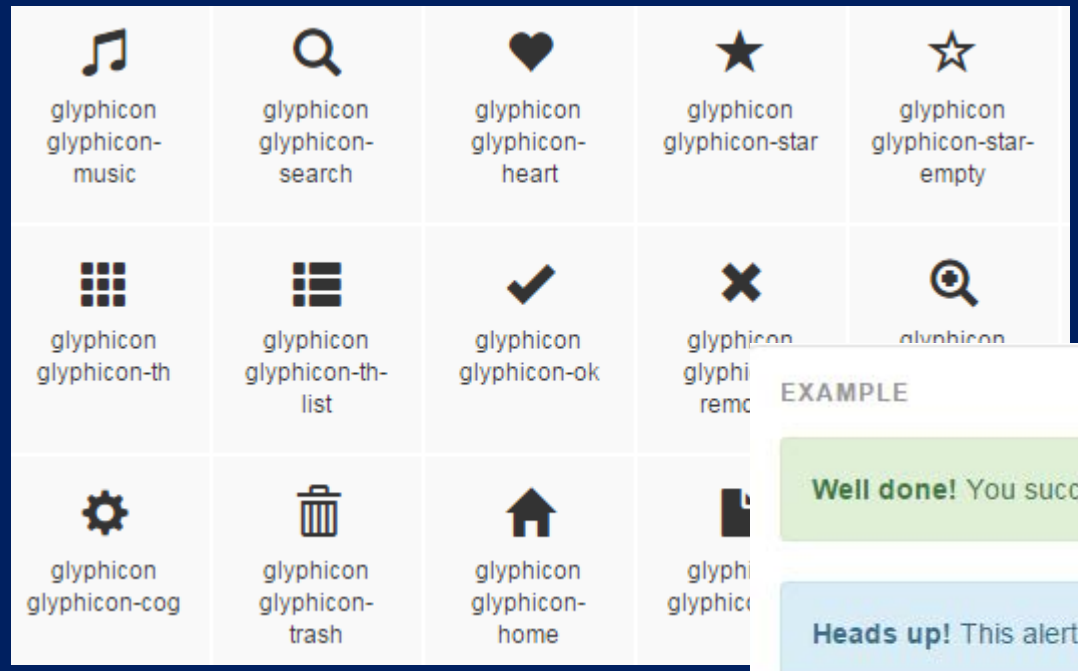




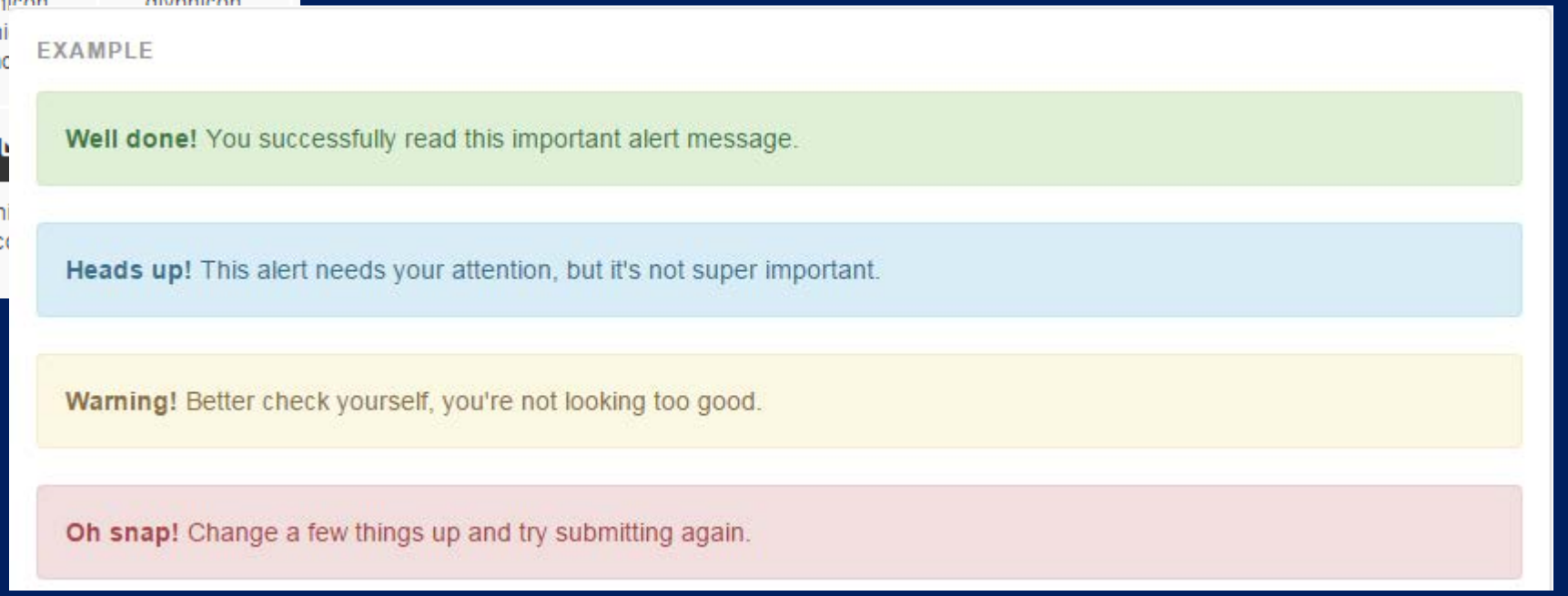
Bootstrap defines 4 device tiers:

xs (phones); sm (tablets); md (desktops); lg (large desktops)
Smaller tiers scale up when viewport grows
Larger tiers stack when viewport shrinks

Bootstrap defines 5 colors defined by in CSS classes; i.e.
btn-danger → red button
bg-primary → dark blue background



Visit getbootstrap.com/components for details



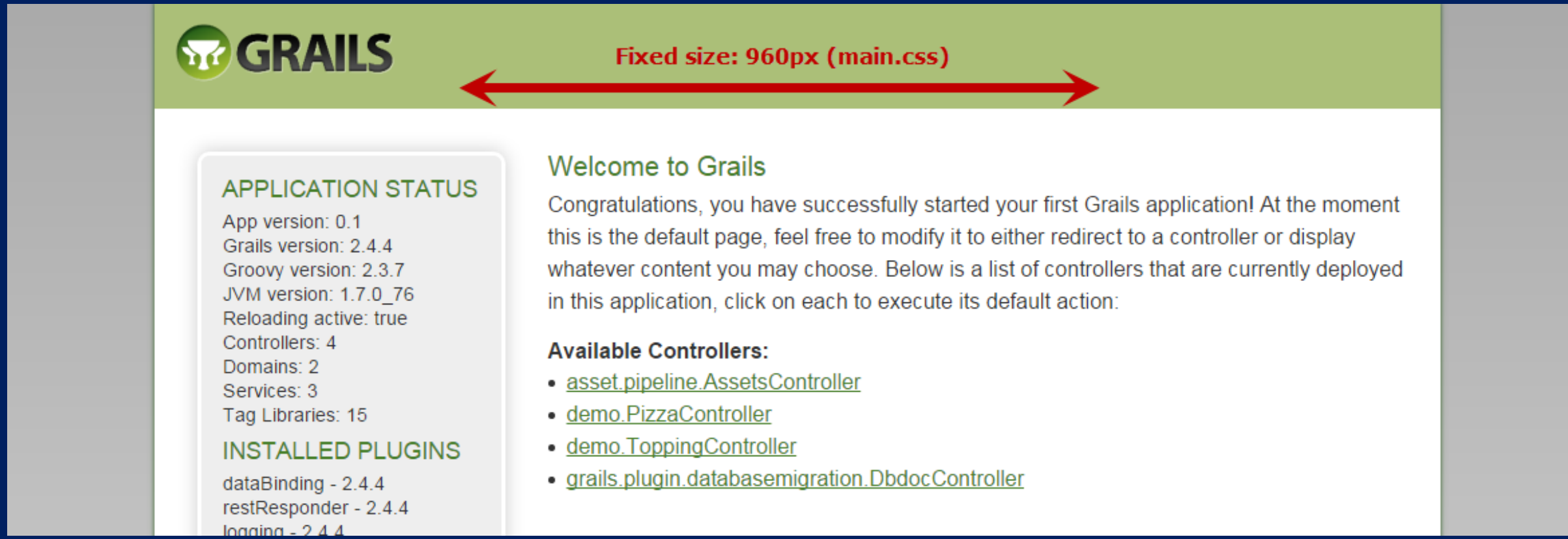
Bootstrap defines many components; i.e.

250+ glyphs in font format
Navs (tabs, pills); Navbar; BreadCrumbs; Pagination;
Dropdown menus with headers, dividers, etc.;
Label; Badges; Thumbnails; Alerts; Jumbotron; and more!

Bootstrap Theme Demo

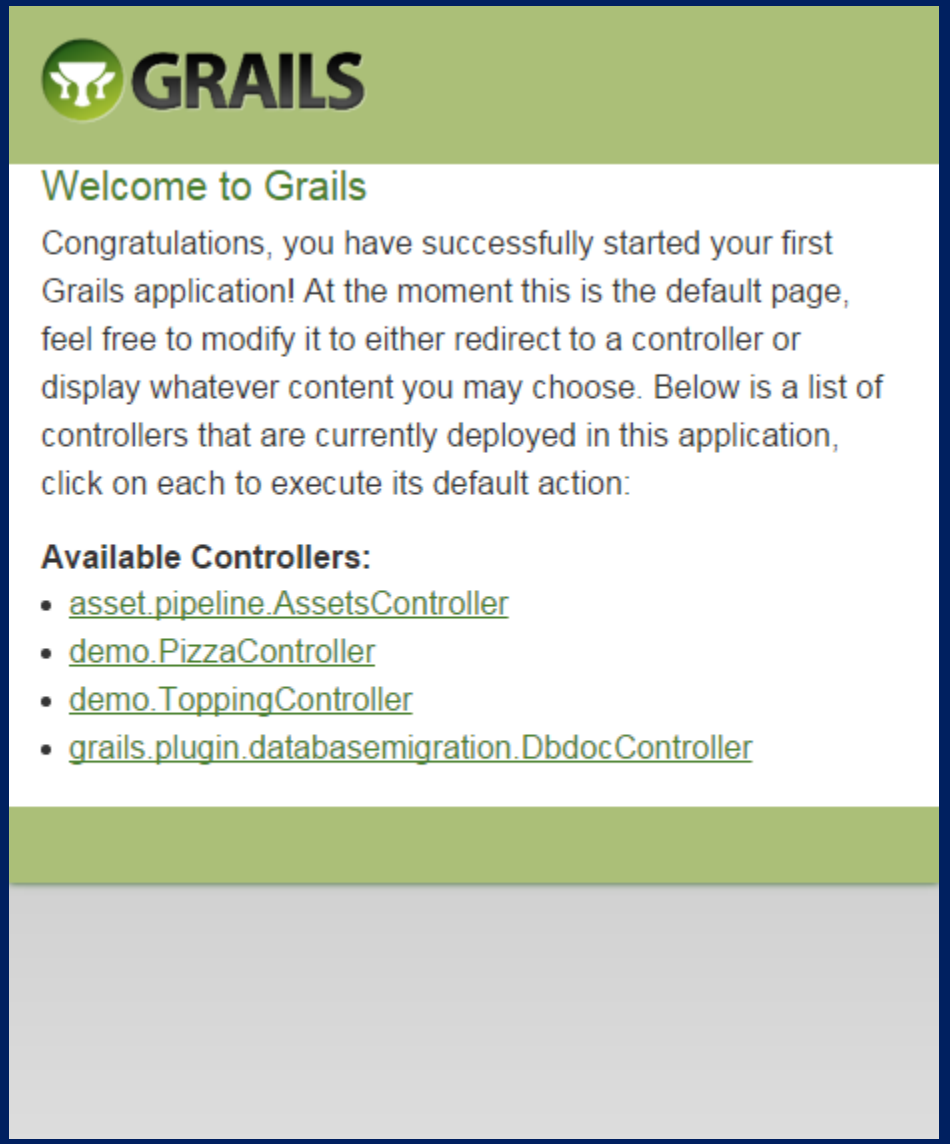
Start with Grails, Mix In Bootstrap

- Grails Scaffolding without Bootstrap
- The Twitter Bootstrap plugin
- Applying Bootstrap to Grails Scaffolding templates
- Applying Bootstrap to your pages



Basic Grails Scaffolding home page

We lost our application status and plugin list



Grails scaffolding templates use the main.css & mobile.css stylesheets. When viewport width drops below 480px, the mobile styles apply

Grails Scaffolding home page on a phone
(Bootstrap XS device tier)

Here's the application status and plugin div:

```
<div id="status" role="complementary">
    <h1>Application Status</h1>
    <ul>
        <li>App version: <g:meta name="app.version"/></li>
        <li>Grails version: <g:meta name="app.grails.version"/></li>
        <li>Groovy version: ${GroovySystem.getVersion()}</li>
        <li>JVM version: ${System.getProperty('java.version')}</li>
        <li>Reloading active: ${grails.util.Environment.reloadingAgentEnabled}</li>
        <li>Controllers: ${grailsApplication.controllerClasses.size()}</li>
        <li>Domains: ${grailsApplication.domainClasses.size()}</li>
        <li>Services: ${grailsApplication.serviceClasses.size()}</li>
        <li>Tag Libraries: ${grailsApplication.tagLibClasses.size()}</li>
    </ul>
    <h1>Installed Plugins</h1>
    <ul>
        <g:each var="plugin" in="${applicationContext.getBean('pluginManager').allPlugins}">
            <li>${plugin.name} - ${plugin.version}</li>
        </g:each>
    </ul>
</div>
```

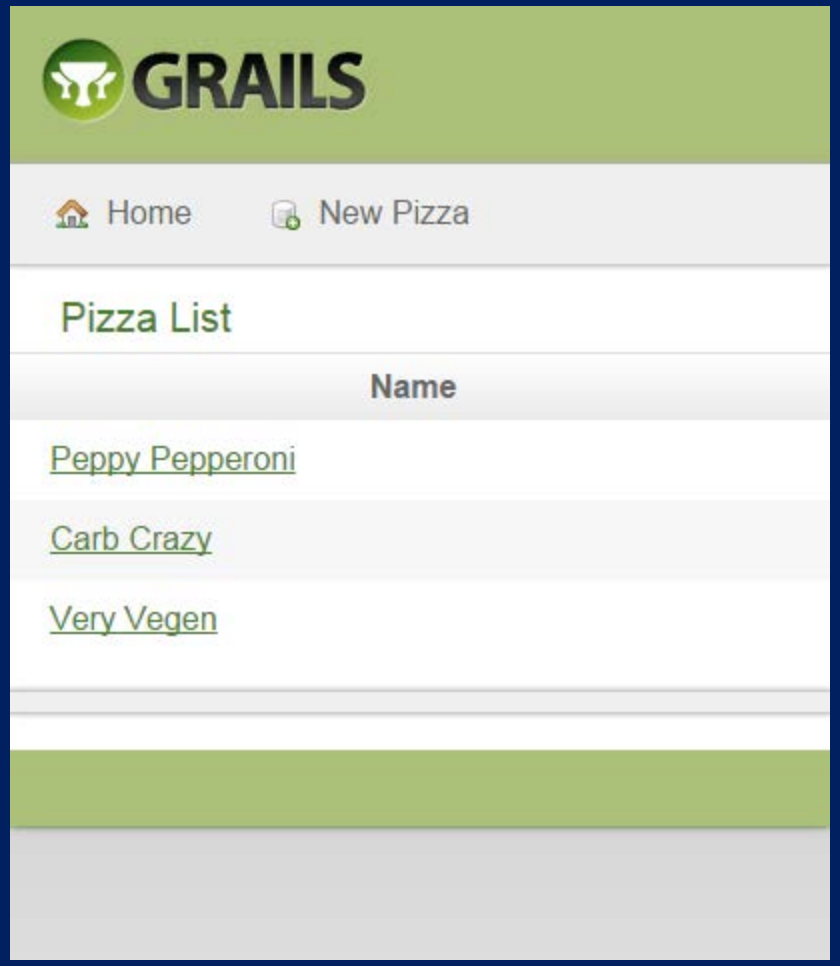
Style defined on index.gsp:

```
@media screen and (max-width: 480px) {
    #status {
        display: none;
    }

    #page-body {
        margin: 0 1em 1em;
    }

    #page-body h1 {
        margin-top: 0;
    }
}
```

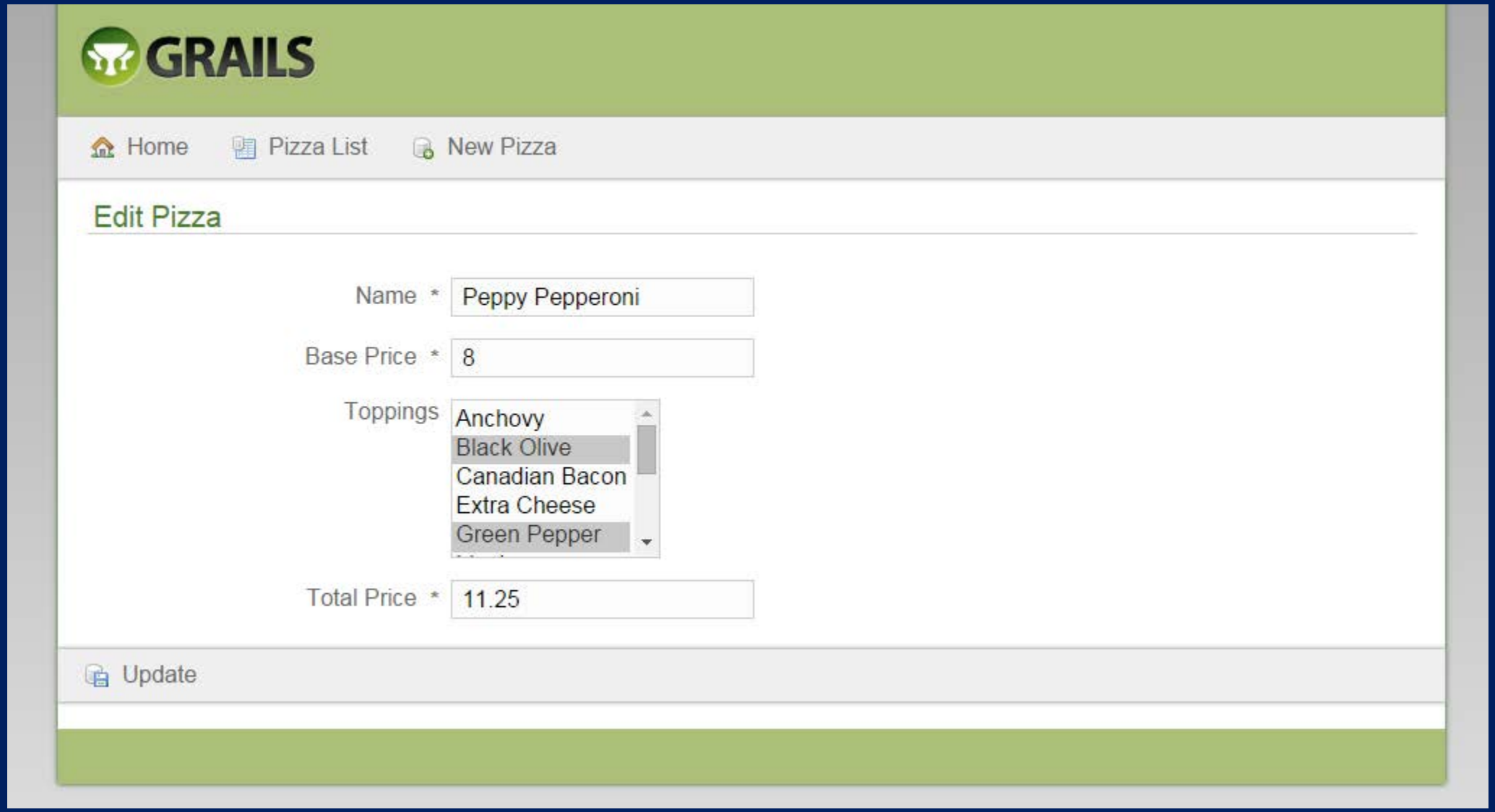
Grails Scaffolding home page markup

Basic Grails Scaffolding list page

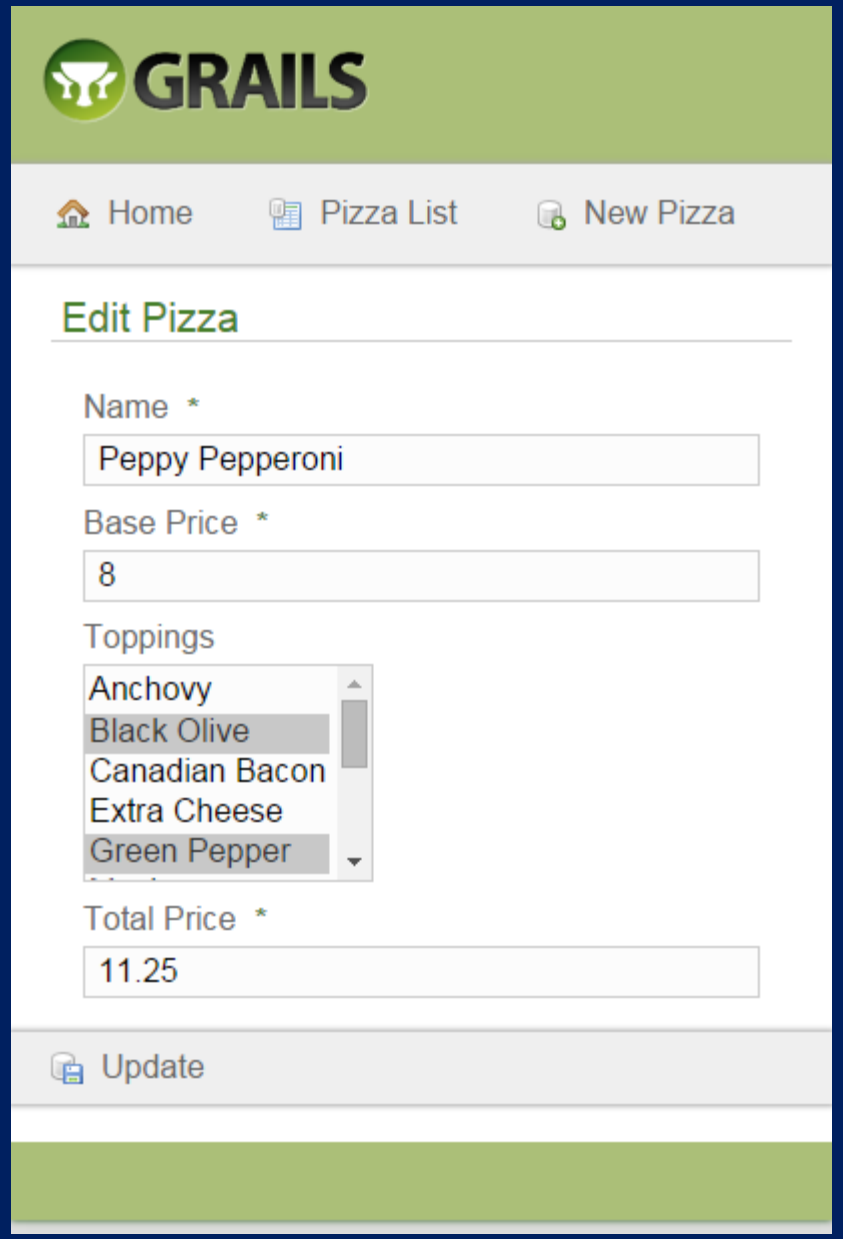
We lost our base and total price data



Grails Scaffolding list page on a phone (Bootstrap XS device tier)



Basic Grails Scaffolding edit page



Our edit labels and fields stack nicely!

Grails Scaffolding edit page on a phone (Bootstrap XS device tier)

Basic Scaffolding Demo

https://grails.org/plugin/twitter-bootstrap

The docs are just slightly confusing!

Plugin Bootstrap CSS framework resource files

Tags: css, resources, framework, bootstrap, grails2.4, assets | Latest: **3.3.4** | Last Updated: **16 March 2015** | Grails version: 1.3.7 > *

Authors: Karol Balejko, Nicholas Wittstruck, Michael Legart | License: Apache License 2.0 | Organization: GroovyDev

Dependency:

```
compile ":twitter-bootstrap:3.3.4"
```

Installation

Add the following to your grails-app/conf/BuildConfig

```
…
plugins {
…
    runtime ':twitter-bootstrap:3.3.1'
…
}
```

```
plugins {
    // plugins for the build system only
    build ":tomcat:7.0.55"

    // plugins for the compile step
    compile ":scaffolding:2.1.2"
    compile ':cache:1.1.8'
    compile ":asset-pipeline:1.9.9"


    // plugins needed at runtime but not for compilation
    runtime ":hibernate4:4.3.6.1" // or ":hibernate:3.6.10.18"
    runtime ":database-migration:1.4.0"
    runtime ":jquery:1.11.1"

    runtime ":twitter-bootstrap:3.3.4"
```

Installing Grails Twitter Bootstrap Plugin

Including the resources with Asset-Pipeline plugin

The recommended way is to use the asset-pipeline plugin (minimal 0.1.7). It is the default since Grails 2.4.0 and replaces the resources plugin.

To include your bootstrap resources add the following to your application's css or js file.

Javascript `grails-app/assets/javascripts/application.js`:

```
//= require bootstrap

console.log("My javascript goes here");
```

Stylesheet `grails-app/assets/javascripts/application.css`:

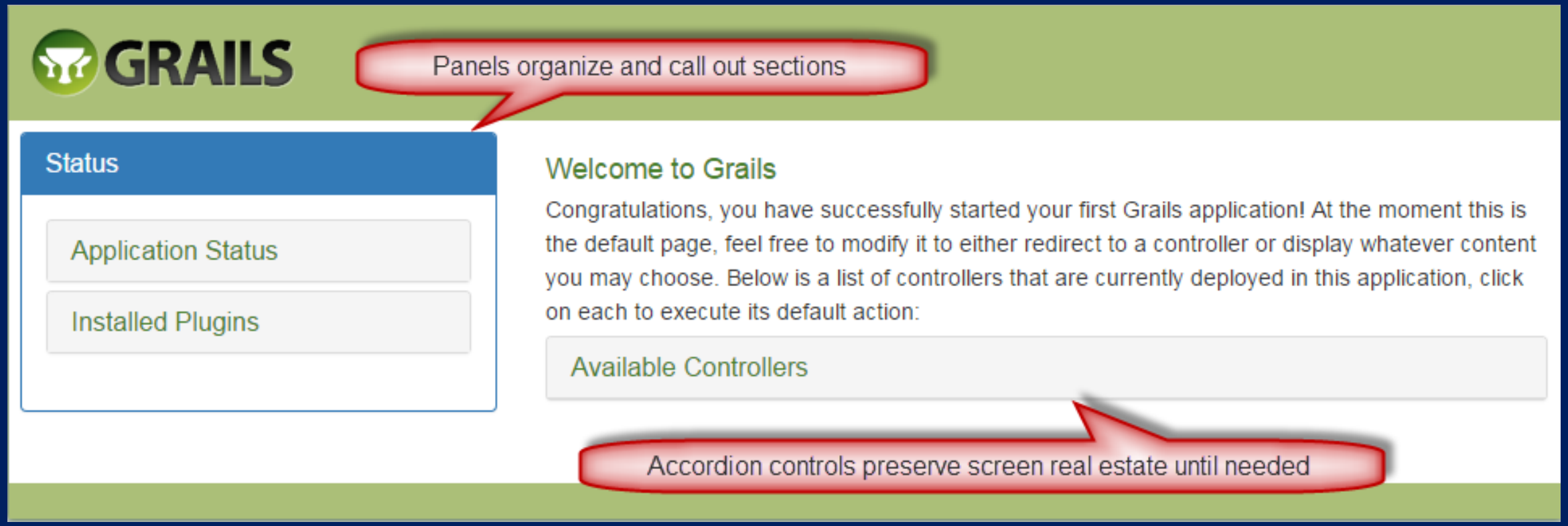
```
/*
*= require bootstrap
*/
```

Be sure to declare the bootstrap.js requirement after the jquery.js requirement!

Beware of conflicting CSS declarations!

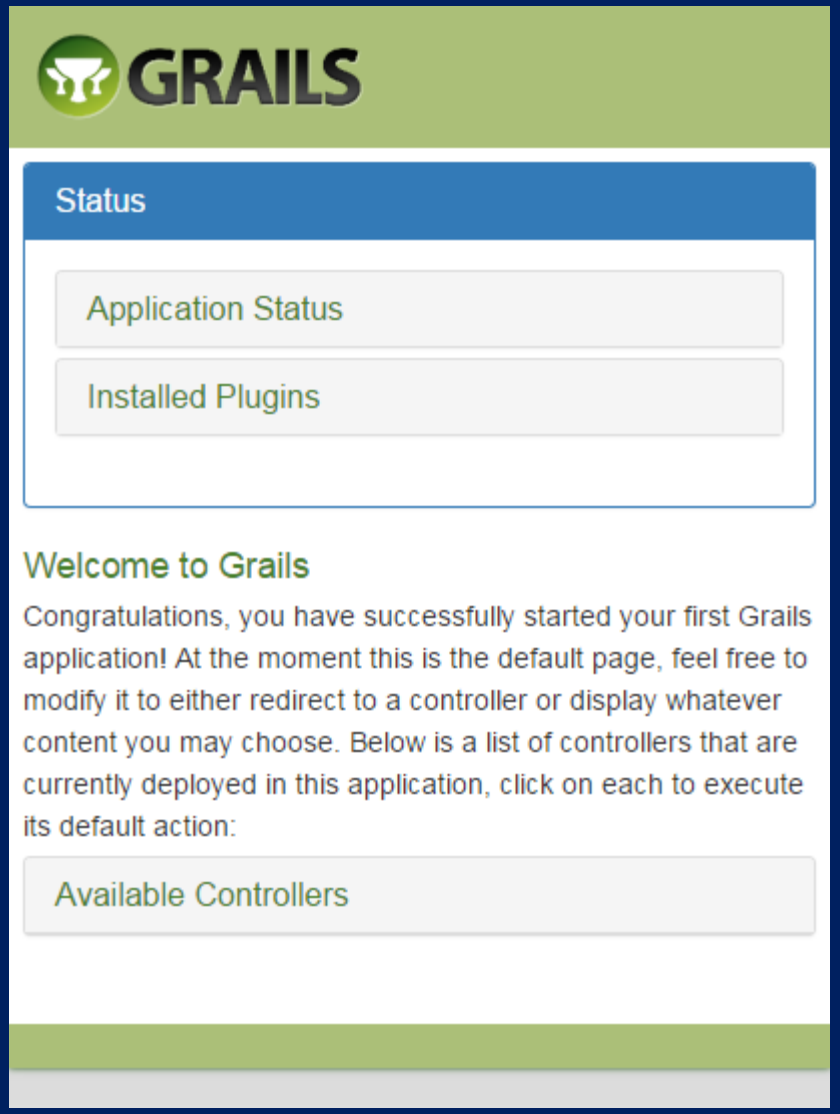
You'll want to remove or disable the mobile.css requirement

Configuring Grails to use Twitter Bootstrap



Bootstrap Enhanced to Grails Scaffolding home page

We kept our application status and plugin list – and they're nicely wrapped in a Bootstrap panel using panel-primary class for color accent



Accordion components allow us to show what must be shown and hide what may not be important at the moment

Bootstrap enhanced Grails Scaffolding home page on a phone (Bootstrap XS device tier)

Bootstrap enhanced Grails Scaffolding home page

Because we're not using mobile css, we retain our field values – and get a scrollbar courtesy of Bootstrap

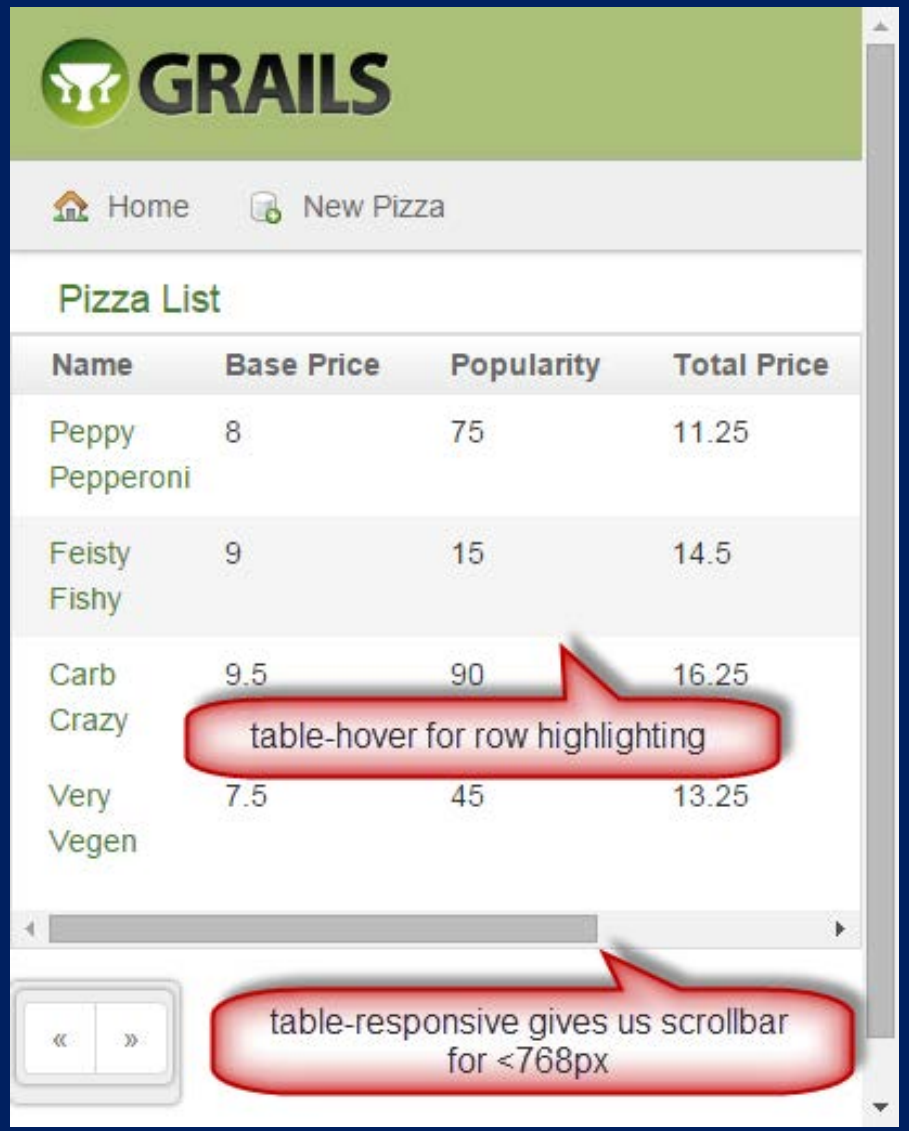


Table decoration options include table-striped, table-hover (shown here), table-bordered, and table-condensed for less cell padding

Bootstrap enhanced Grails Scaffolding home page on a phone (Bootstrap XS device tier)

Enhanced Scaffolding Demo

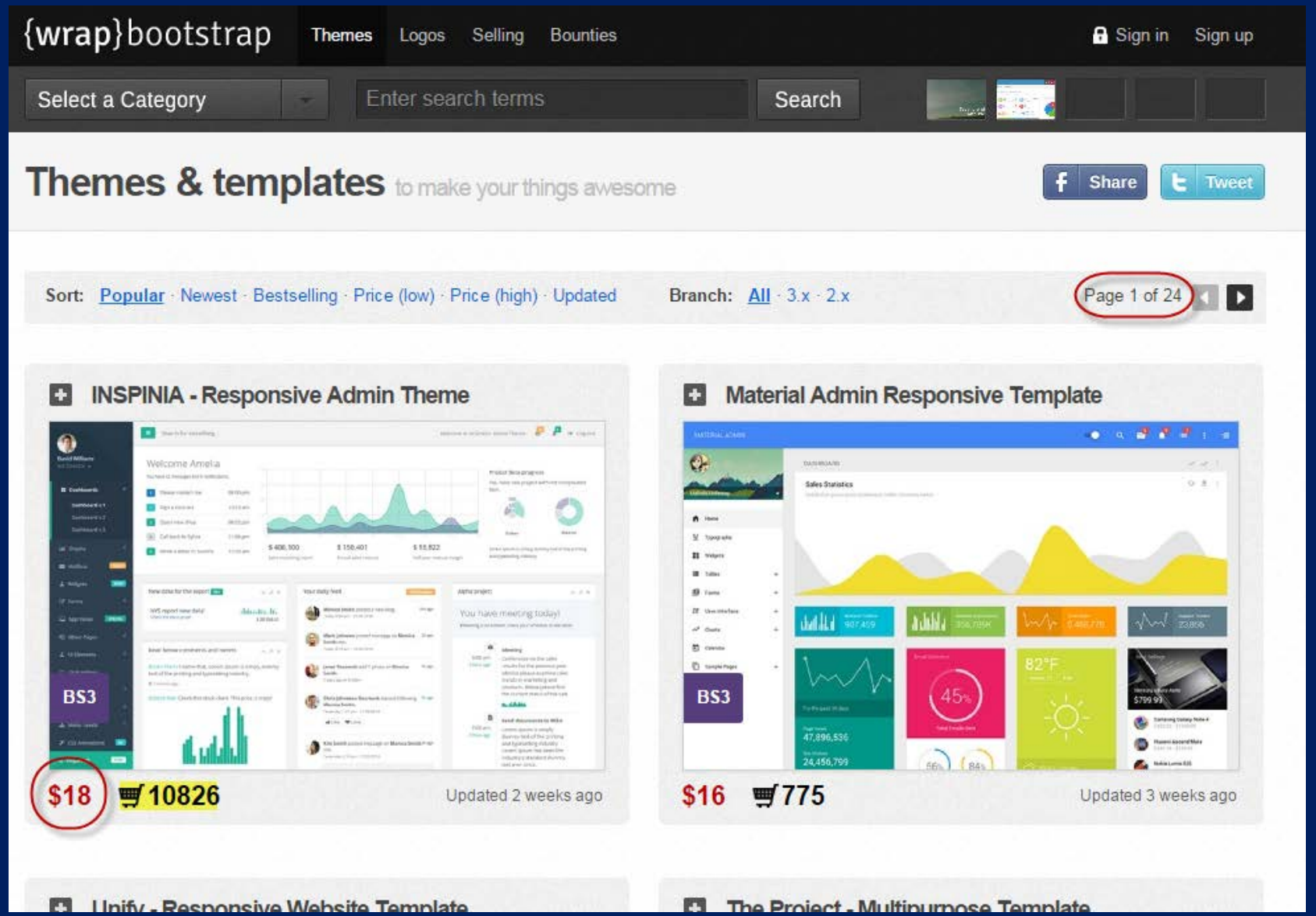
Final words about adding Bootstrap to Grails

- The dynamic nature of the scaffolding templates actually makes integrating Bootstrap features harder
- Your existing custom applications will likely be easier to integrate Bootstrap
  - Find the Bootstrap component you want to use
  - Find the place in your custom page you want to use it
  - Use it!

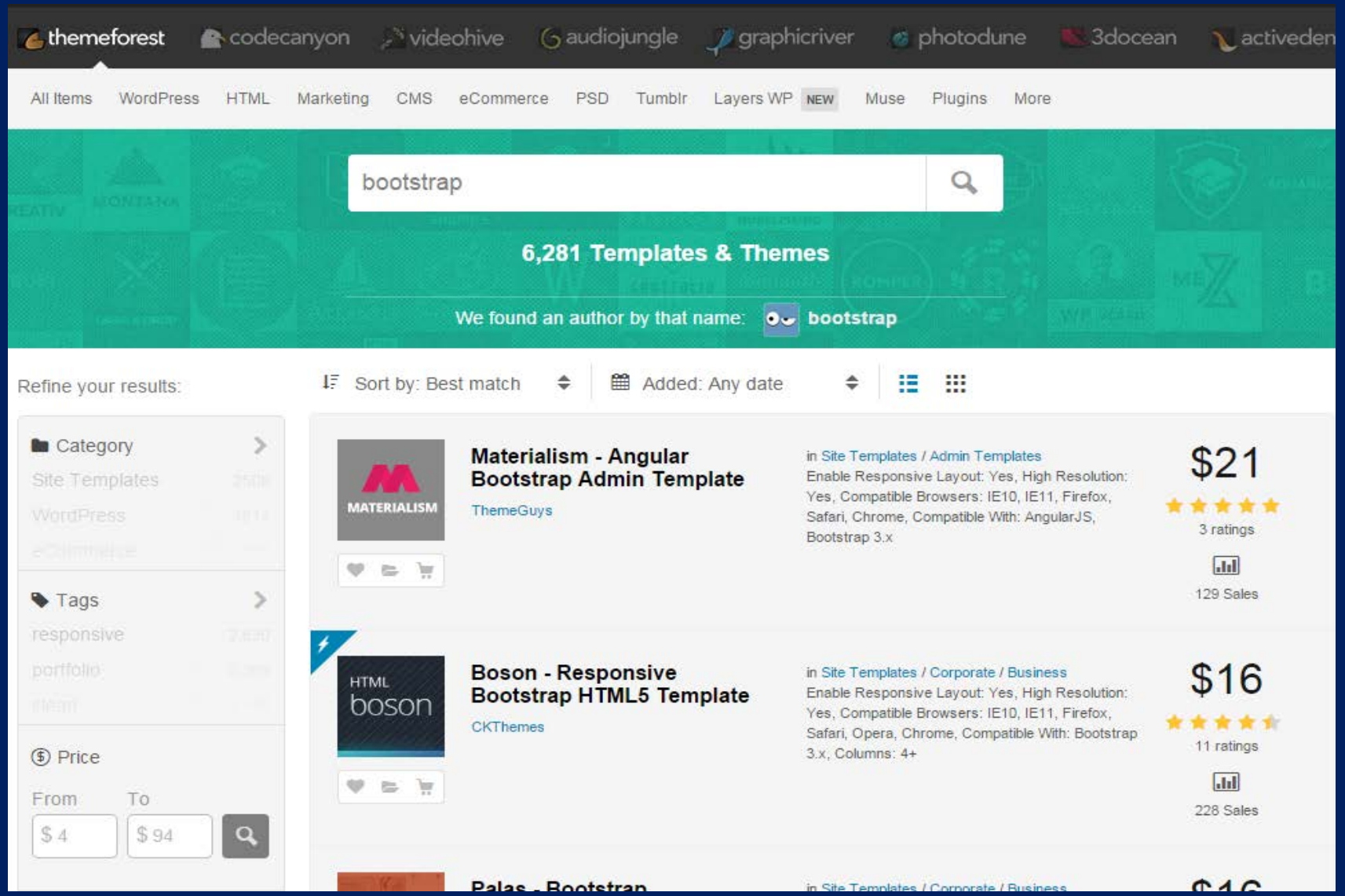
For a green field app, maybe the best approach is start with a beautiful design…

Start with Bootstrap, Mix In Grails

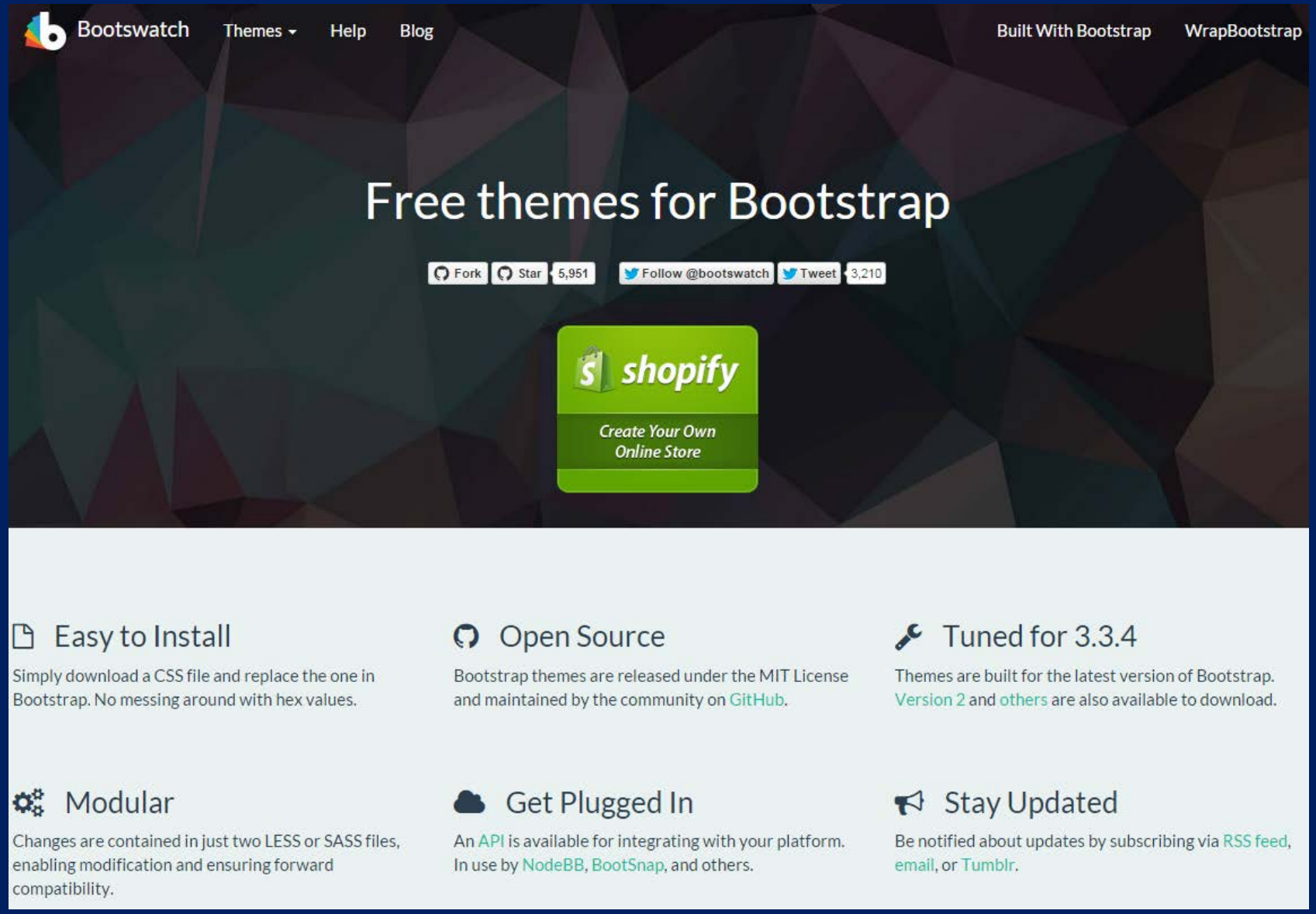
- Pick a theme, any theme
- HTML, CSS and JavaScript… Oh my!
- Coercing a commercial theme to work with Grails



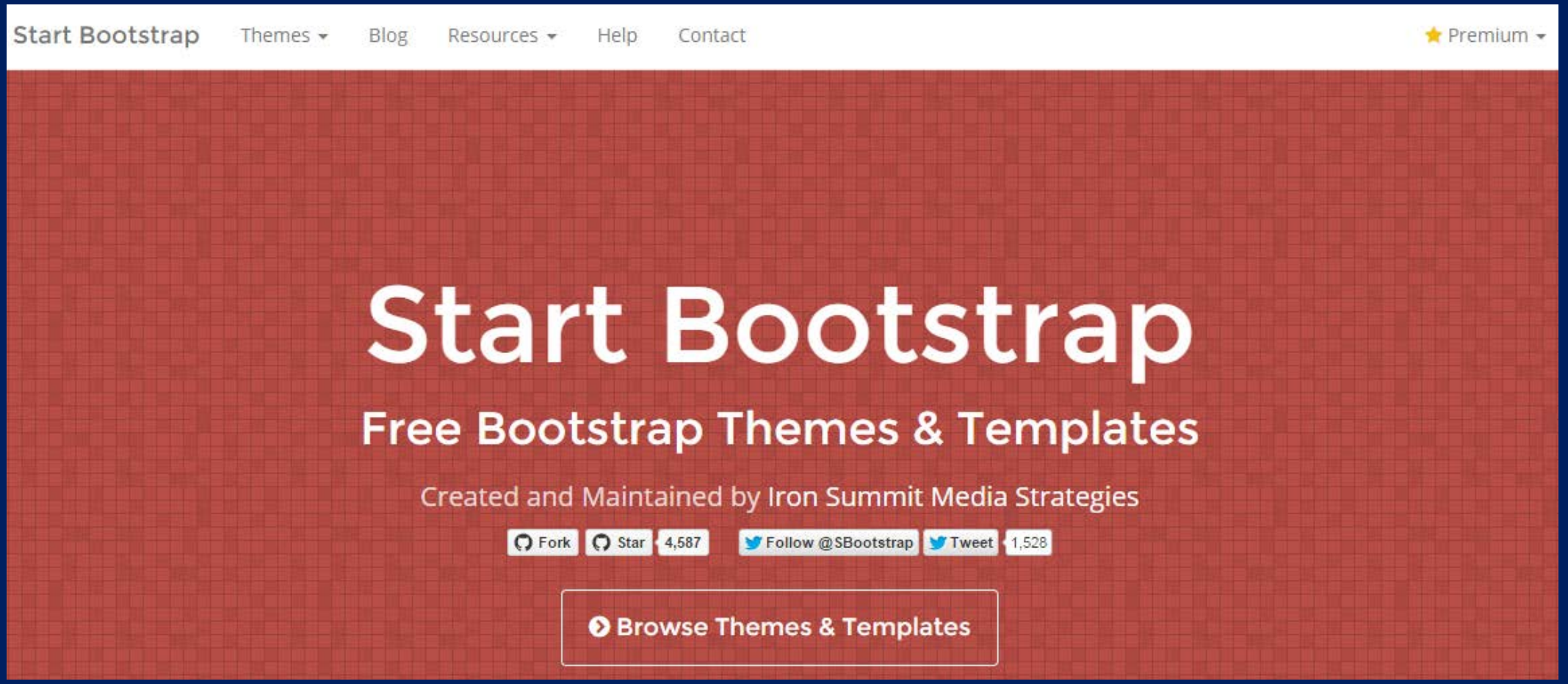
WrapBootstrap.com
Low cost, visually appealing, very robust themes



ThemeForest.net
Low cost, visually appealing, very robust themes
(Beware: Not just Bootstrap themes sold here!)



BootsWatch.com
Visually appealing themes that are free, but limited



StartBootstrap.com
Visually appealing themes that are free, but limited

Most themes have markup/JavaScript that’s not too well organized

WrapBootstrap.com Unity Theme Demo

Some steps to transforming a Bootstrap Theme to Grails – Order out of Chaos

- Identify common header, menu bar, footer, sidebar, etc. markup
- In each common block of markup, identify any context specific attributes or markup
  - A menu bar might have an 'active' attribute to signal which menu item corresponds to the current page or module
  - You'll need to keep track of that state somewhere
- Consolidate common, repeated markup
  - Extract the common markup into a layout
    - Replace link and JavaScript source references with asset pipeline tags
  - Extract context specific content into a separate page
    - Reference the layout in a meta tag in the head section
    - Where content is repeated, try to use a template
  - Beware of timing issues related to page rendering
    - Static markup is rendered as it's encountered, but I've seen templates get rendered out of sequence
    - This can cause page state to be incorrect for the template's needs

Some steps to transforming a Bootstrap Theme to Grails

...

- The next steps are optional depending on how close you want to stay to pure HTML rather than rely on a lot of Grails tags
  - There are good arguments for staying pure and equally good arguments for leveraging Grails tag productivity
  - If you stay with pure HTML, you'll need to ensure your security and url mappings align with the urls/uris used in the page. There's no referencing controllers and actions!
  - Replace all <a> tags with g:link (or not)
- Replace manual table markup with g:forEach wrappers around <tr> elements
- Forms
  - Replace HTML forms with g:forms
  - Replace form controls with Grails equivalents if desired

Custom Site Demo (time permitting)

Resources

- Get Grails: grails.org
- Get Bootstrap: getbootstrap.com
- Get some themes
  - Wrapbootstrap.com
  - Themeforest.com
- Get Jack
  - LinkedIn: jackfrosch
  - Twitter: @jackfrosch
  - Email: jackfrosch@gmail.com

Questions?