#NoXML: Eliminating XML in Spring Projects

Matt Raible • http://raibledesigns.com

Photos by McGinity Photo

RAIBLE DESIGNS
ENTERPRISE OPEN SOURCE CONSULTING

Who is **Matt Raible**?

Montanan, Father, Husband, Skier, Mountain Biker, Whitewater Rafter

Open Source Connoisseur

Web Developer and UI Architect

Bus Lover

Blogger on raibledesigns.com

What about **You**?

How long have you been programming in Java?

Did you read Rod Johnson's first book?

Did you ever use Struts?

What about Acegi Security?

What's your most memorable development nightmare?

#NoXML

ROCKY MOUNTAIN
MARAVIA
NRS

“Parallelize”

RAIBLE DESIGNS
ENTERPRISE OPEN SOURCE CONSULTING

« First Avalanche Game | Main | Free Wireless along... »

Friday April 14, 2006

Tips for Productivity and Happiness at Work

Every so often, I get asked what my strategy is for "getting things done". This morning, I had a short session of mass productivity, and on my ride into work, got inspired to jot down a few tips productivity tips. Keep in mind that I grew up in the back woods of Montana with no electricity and I'm mostly Irish.

- **Write stuff down**. The first and most important means to getting things done is to keep a todo list. I keep mine on Ta-da List and update it daily, if not hourly. My two most important lists are "When I have time" and "This Week". I regularly re-order "This Week" for the current day.
- **Quit reading e-mail and blogs**. One of the ways I can tell I'm in uber-productive mode is my unread (or starred) mail piles up and I haven't read any blog posts (or blogged myself) in a couple days. I like GMail because I can easily star an e-mail after briefly skimming it. For some reason, flagging e-mail in Thunderbird or Mail.app doesn't work for me - I rarely go back and find those e-mails. With GMail, I'm always conscious I need to revisit starred messages.
- **Work on open source late at night, with a beer on your desk**. While I do get the opportunity to work on open source at my day job, I still find that I'm most productive at night. Maybe this is because no one bugs me via e-mail or IM, or maybe it's just because the world is asleep. The strange thing is I often find myself motivated at 3 p.m. for my 11 p.m. workload. However, when I get to 11 p.m., I'm not motivated to work on anything. I've found that cracking open a beer at 11 when I start helps me focus and quit worrying about all the other computer-related tasks I need to do. Also, on beer #2 or 3, you'll start to forget what time it is and *really* start getting things done. NOTE: this isn't for rookies. If you're a lightweight and get hammered on two beers - just go to bed when you start coding with one eye shut. It helps to sleep in the next day after doing this. If you finish off a six-pack before going to bed, it's probably best not to check your code in - you're probably going to spend the next day fixing it anyway. Regardless, this is a great way to get started on a new feature because you're less concerned about the details and more concerned about the big picture. I've also found that "bug fixing juice" can be great for fixing bugs - it gives you a different perspective on the problem. I told you I was Irish didn't I? ;-)
- **Work disconnected**. To further facilitate not checking e-mail or reading blogs, I've found that going to a coffee shop w/o connectivity is my most productive environment. They have liquid motivation in the form of coffee, and you can feed your brain with breakfast/lunch or some kind of snack. My most productive days are the ones where I show up at my local Einstein's (bagel shop) at 6

The JHipster Mini-Book

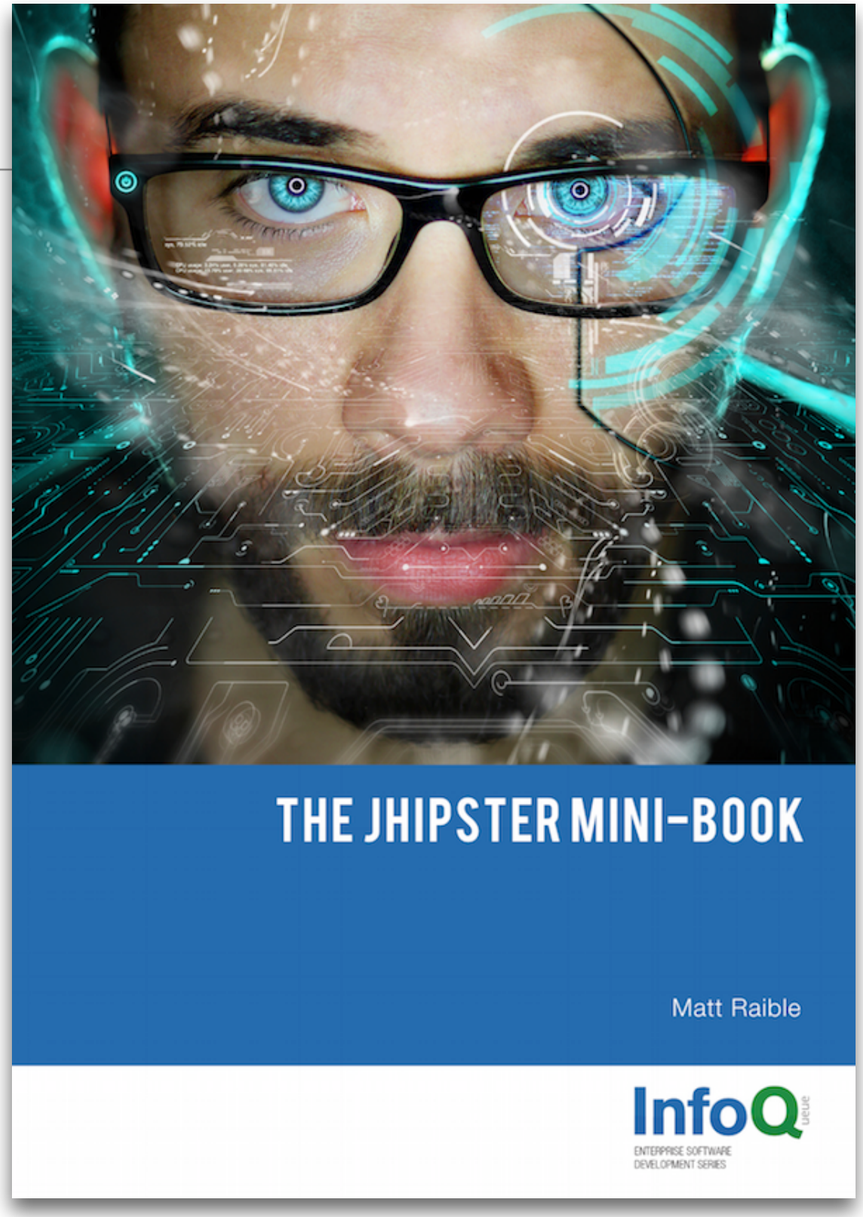
Written by yours truly

Quick and to the point

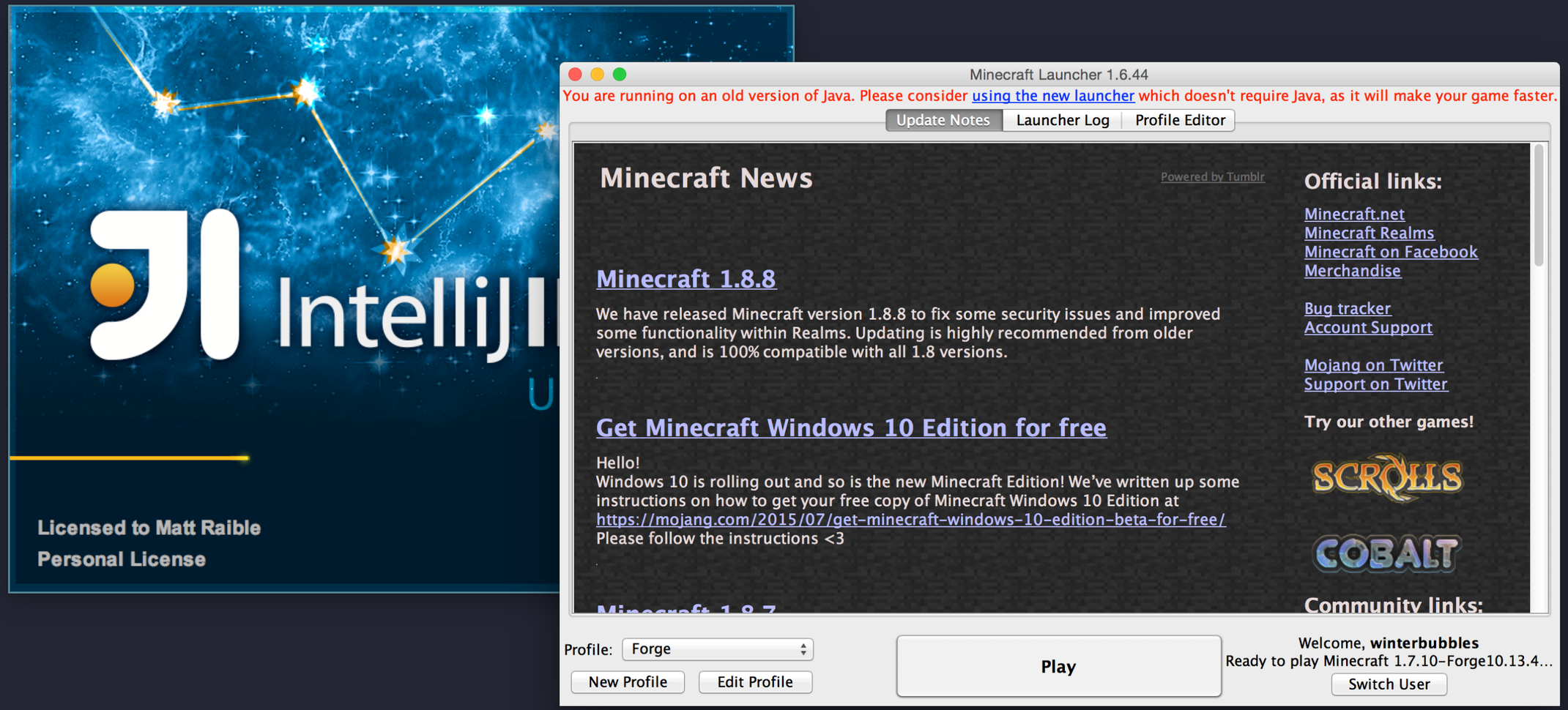
Less than 150 pages

Developing a Real World App

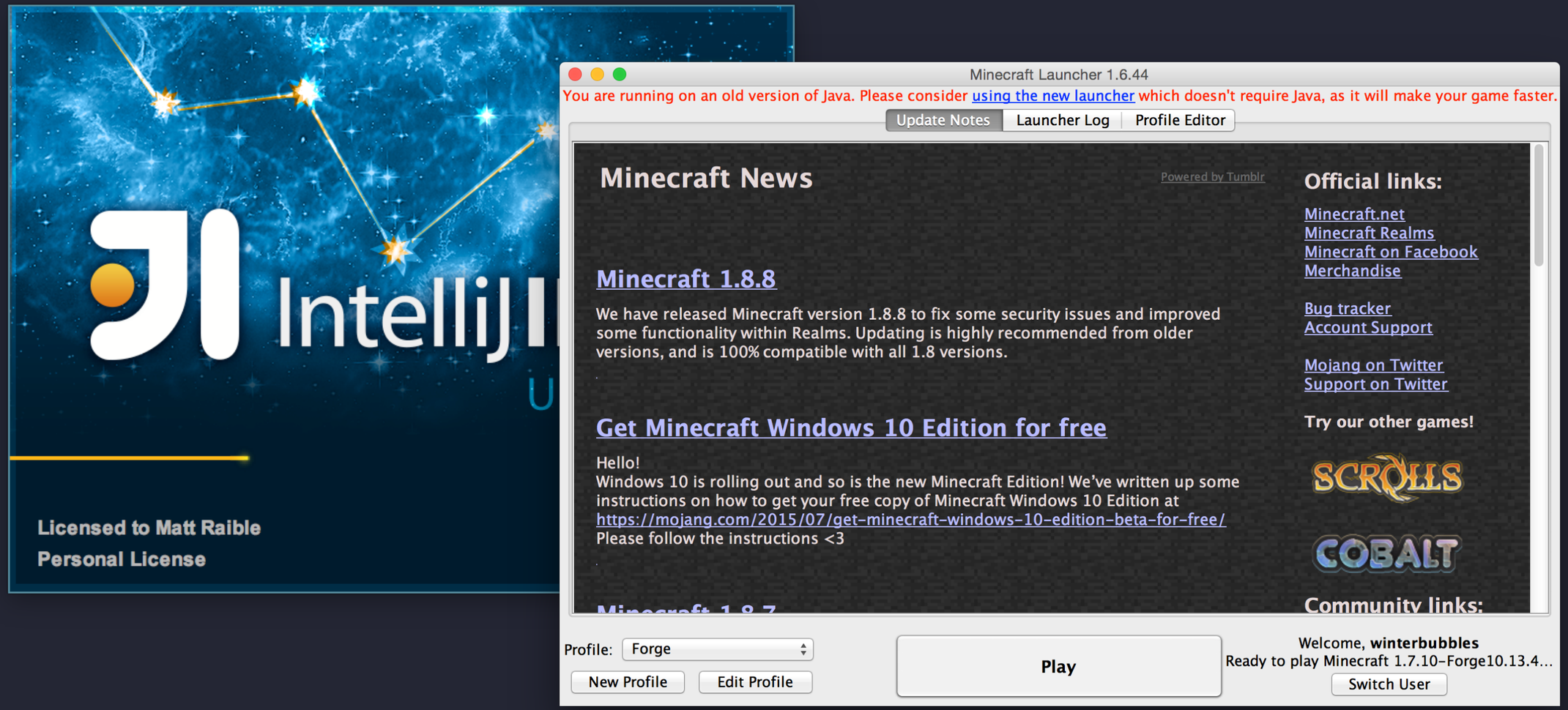
Free Download from InfoQ



Java on the Desktop



Java on the Desktop



APIs on the JVM

"When web companies grow up they turn into Java shops"

— *James Governor*
JAX London, October 29, 2013

Good Things in JavaLand

Big Data: Hadoop, Storm, Kafka, Cassandra and Apache Giraph

Groovy and Grails

Scala and Play

Spring Boot

Ratpack

Good Things in JavaLand

Big Data: Hadoop, Storm, Kafka, Cassandra and Apache Giraph

Groovy and Grails

Scala and Play

Spring Boot

Dropwizard

Spring Timeline

Spring Framework 1.0 Released

March 24, 2004

Spring 1.2.6 wins Jolt Productivity Award

2006

Spring 2.0

October 2006

Spring 2.5

November 2007

Spring Timeline

Spring 3.0

December 2009

Spring 4.0

December 2013

Spring Boot 1.0

April 1, 2014

Spring 4.2

July 2015

spring-projects / **spring-boot**

Watch 490 | Star 2,881 | Fork 2,997

September 7, 2015 – September 14, 2015

Period: **1 week**

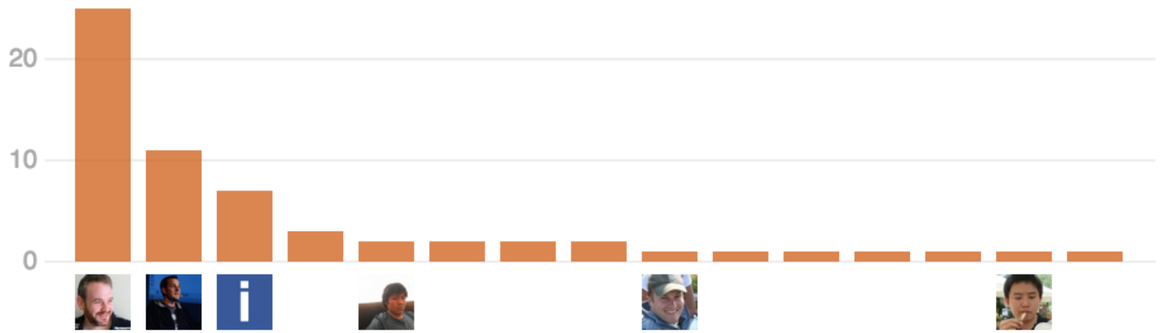
Overview

6 Active Pull Requests

38 Active Issues

0	**6**	**26**	**12**
Merged Pull Requests	Proposed Pull Requests	Closed Issues	New Issues

Excluding merges, **16 authors** have pushed **43 commits** to master and **62 commits** to all branches. On master, **811 files** have changed and there have been **3,243 additions** and **2,047 deletions**.



SPRING INITIALIZR @ start.spring.io

1. bash

```
~75% [mraible:~] $ curl start.spring.io


  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/

:: Spring Initializr ::  https://start.spring.io

This service generates quickstart projects that can be easily customized.
Possible customizations include a project's dependencies, Java version, and
build system or build structure. See below for further details.

The services uses a HAL based hypermedia format to expose a set of resources
to interact with. If you access this root resource requesting application/json
as media type the response will contain the following links:
+-----------------+----------------------------------------+
| Rel             | Description                            |
+-----------------+----------------------------------------+
| gradle-build    | Generate a Gradle build file           |
| gradle-project  | Generate a Gradle based project archive |
| maven-build     | Generate a Maven pom.xml               |
```

JHipster

http://jhipster.github.io/

Greetings, Java Hipster!

JHipster is a
Yeoman generator,

used to create a
Spring Boot + **AngularJS**
project.

Programmer to Programmer
Rod Johnson
Rod Johnson is an enterprise Java architect specializing in scalable web applications. He has worked with both Java and J2EE since their release, and he is a member of JSR 154 Expert Group defining the Servlet 2.4 specification.
expert one-on-one
J2EE Design and Development
wrox
Updates, source code, and Wrox technical support at www.wrox.com

Spring Boot

Create stand-alone Spring applications

Embeds Tomcat, Jetty or Undertow directly

Provides opinionated 'starter' POMs to simplify your Maven configuration

Also supports/promotes using Gradle

Spring Boot

Automatically configures Spring whenever possible

Provides production-ready features such as metrics, health checks and externalized configuration

Absolutely no code generation and no requirement for XML configuration

Grails 3.0 is based on Spring Boot

Spring Boot

Automatically configures Spring whenever possible

Provides production-ready features such as metrics, health checks and externalized configuration

Absolutely no code generation and **no requirement for XML configuration**

Grails 3.0 is based on Spring Boot

JHipster

Foundational Frameworks

- Spring Boot
- Spring Security
- AngularJS
- Bootstrap
- Bower
- Metrics

Project Options

- Java 7 or Java 8
- Maven or Gradle
- Authentication Type: cookie-based or OAuth2
- Type of Database: SQL or NoSQL
- Caching: EhCache or Hazelcast
- Grunt or Gulp.js

Yeoman

http://yeoman.io/

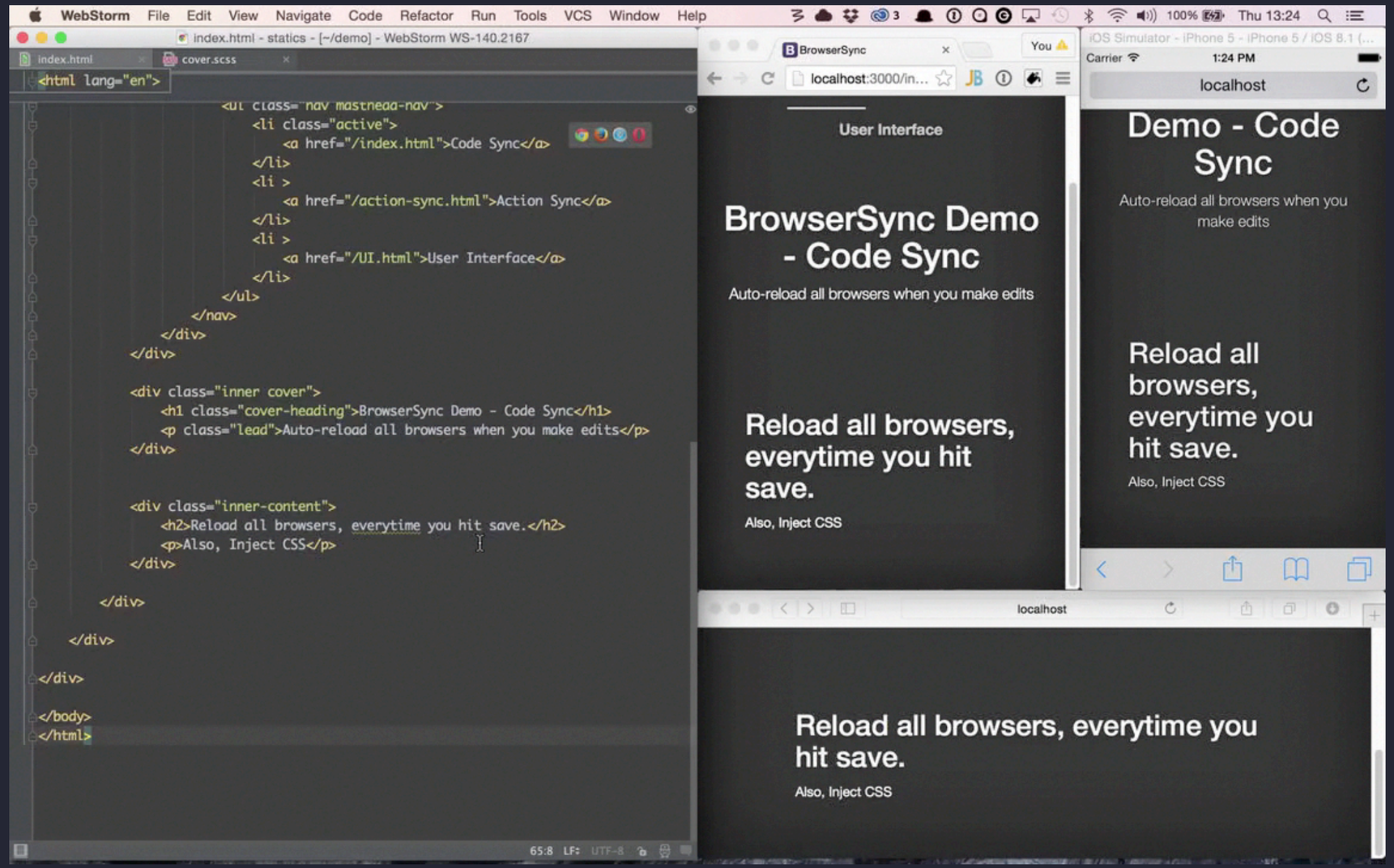
The web's scaffolding tool for modern webapps

Helps you kickstart new projects

Promotes the Yeoman workflow

BrowserSync

http://www.browsersync.io/



“Parallelize”

#NoXML

struts-config.xml

```xml
<?xml version="1.0" encoding="ISO-8859-1" ?>

<!DOCTYPE struts-config PUBLIC
  "-//Apache Software Foundation//DTD Struts Configuration 1.1//EN"
  "http://jakarta.apache.org/struts/dtds/struts-config_1_1.dtd">

<struts-config>

  <form-beans>
    <form-bean
      name="templateForm"
      type="org.appfuse.webapp.form.TemplateForm"
    />
    <form-bean
      name="resumeSkillForm"
      type="org.appfuse.webapp.form.ResumeSkillForm"
    />
    <form-bean
      name="resumeForm"
      type="org.appfuse.webapp.form.ResumeForm"
    />
    <form-bean
      name="roleForm"
      type="org.appfuse.webapp.form.RoleForm"
    />
```

build.xml

```
<!DOCTYPE project [
    <!ENTITY properties SYSTEM "file:./properties.xml">
    <!ENTITY app-settings SYSTEM "file:./app-settings.xml">
]>

<project name="appfuse" basedir="." default="package-web">

    <!-- import project-wide properties (classpath, jar file locations) -->
    &properties;
    <!-- import application settings for xdoclet -->
    &app-settings;

    <target name="define-tasks" description="defines custom tasks">
        <!-- Taskdefs -->
        <taskdef resource="net/sf/antcontrib/antcontrib.properties"
            classpath="${ant-contrib.jar}"/>

        <taskdef name="xdoclet"
            classname="xdoclet.DocletTask"
            classpathref="xdoclet.classpath"/>

        <taskdef name="todo"
            classname="xdoclet.modules.doc.DocumentDocletTask"
            classpathref="xdoclet.classpath"/>
```

3. bash

```
⌂64% [mraible:~/Downloads/appfuse-APPFUSE-1_4] $ ant
Buildfile: /Users/mraible/Downloads/appfuse-APPFUSE-1_4/build.xml
Warning: 'file:./properties.xml' in /Users/mraible/Downloads/appfuse-APPFUSE-1_4/build.xml should be expressed simply as
 './properties.xml' for compliance with other XML tools
Warning: 'file:./app-settings.xml' in /Users/mraible/Downloads/appfuse-APPFUSE-1_4/build.xml should be expressed simply
as './app-settings.xml' for compliance with other XML tools

build.properties.missing:

define-tasks:

init:

BUILD FAILED
/Users/mraible/Downloads/appfuse-APPFUSE-1_4/build.xml:52: j2ee.jar (/Users/mraible/Downloads/appfuse-APPFUSE-1_4/lib/j2
sdkee1.4/lib/j2ee.jar) is not present.
          Please download the J2EE SDK and set your J2EE_HOME.

Total time: 0 seconds
⌂64% [mraible:~/Downloads/appfuse-APPFUSE-1_4] 1 $
```

applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
    "http://www.springframework.org/dtd/spring-beans.dtd"
    [<!ENTITY database SYSTEM "WEB-INF/applicationContext-database.xml">
     <!ENTITY dao SYSTEM "WEB-INF/applicationContext-@DAO-TYPE@.xml">
     <!ENTITY service SYSTEM "WEB-INF/applicationContext-service.xml">]>

<beans>

  <!-- import database settings -->
  &database;

  <!-- import dao definitions -->
  &dao;

  <!-- import service definitions -->
  &service;

</beans>
```

Acegi Security

Acegi Security in one hour

www.javaworld.com/article/2077780/java-web-development/acegi-security-in-one-hour.html

TRENDING: Learn Java · Java App Dev · Build Automation · Internet of Things · Java Toolbox · Open Source Java · Java 101 · Resources/White Papers

JAVAWORLD

Most read:

logentries 10 Best Practices for Log Management MORE

Home > Enterprise Java >

HOW-TO

Acegi Security in one hour

A concise guide to securing your Java Web applications

By ShriKant Vashishtha

JavaWorld | Oct 18, 2007 1:00 AM PT

RELATED TOPICS

Java Security

Enterprise Java

Java App Dev

Spring framework

82 COMMENTS

Acegi Security has been generating some serious positive buzz among Java enterprise developers, so you might be wondering how it works. In this article, ShriKant Vashishtha walks you through all the steps of a hands-on Acegi Security implementation. First you'll set up form-based authentication and authorization services for a Java-based Web application, then you'll customize Acegi Security for dynamic authorization, as well as integration with proprietary authentication implementations such as LDAP.

MORE LIKE THIS

Adopting Struts 2.0

Acegi security for JSF applications

Simplify directory access with Spring LDAP

IntelliJ IDEA 14.1 by JetBrains

Get it now

applicationContext-security.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">
<beans>
    <!-- ======================== FILTER CHAIN ======================== -->
    <bean id="filterChainProxy"
        class="org.acegisecurity.util.FilterChainProxy">
        <property name="filterInvocationDefinitionSource">
            <value>
                CONVERT_URL_TO_LOWERCASE_BEFORE_COMPARISON
                PATTERN_TYPE_APACHE_ANT
                /j_acegi_security_check*=httpSessionContextIntegrationFilter,authenticationProcessingFilter
                /**/*=httpSessionContextIntegrationFilter,logoutFilter,authenticationProcessingFilter,
                      securityContextHolderAwareRequestFilter,anonymousProcessingFilter,
                      exceptionTranslationFilter,filterInvocationInterceptor
            </value>
        </property>
    </bean>
```

AppFuse

```
mvn archetype:generate -B -DarchetypeGroupId=org.appfuse.archetypes \
-DarchetypeArtifactId=appfuse-basic-spring-archetype -DarchetypeVersion=3.5.0 \
-DgroupId=com.raibledesigns -DartifactId=noxml \
-DarchetypeRepository=https://oss.sonatype.org/content/repositories/appfuse
```

AppFuse QuickStart

Created by Matt Raible, last modified by Ivan Garcia Sainz-Aja on Feb 18, 2015

To start developing web applications with AppFuse, please following the instructions below:

Create a project

1. Setup your Development Environment - or follow the steps below if you're a veteran.
    a. Download and install JDK 7+ (make sure your JAVA_HOME environment variable points to the JDK, not a JRE).
    b. Download and install MySQL 5.5+ (or use a different database see here).
    c. Setup a local SMTP server or change `mail.properties` (in *src/main/resources*) to use a different host name - it defaults to "localhost".
    d. Download and install Maven 3.1.0+.
2. From the command line, cd into your "Source" directory (c:\Source on Windows, ~/dev on Unix) run the Maven command you see below after choosing your web framework and other options.

Choose Application: (•) **AppFuse** ( ) AppFuse Light

GroupId: com.mycompany (?)

ArtifactId: myproject (?)

Version: 3.5.0 (?)

Web Framework: Spring MVC (?)

Multi-Module Project? [ ] (?)

Command Line:

```
mvn archetype:generate -B -DarchetypeGroupId=org.appfuse.archetypes -
DarchetypeArtifactId=appfuse-basic-spring-archetype -
DarchetypeVersion=3.5.0 -DgroupId=com.mycompany -
DartifactId=myproject -
DarchetypeRepository=https://oss.sonatype.org/content/repositories/appfu
se
```

2. bash

```
[mraible:~] $ mvn archetype:generate -B -DarchetypeGroupId=org.appfuse.archetypes -DarchetypeArtifactId=appfuse-basic-spring-archetype -DarchetypeVer
sion=3.5.0 -DgroupId=com.raibledesigns -DartifactId=noxml -DarchetypeRepository=https://oss.sonatype.org/content/repositories/appfuse
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building Maven Stub Project (No POM) 1
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] >>> maven-archetype-plugin:2.2:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
[INFO] <<< maven-archetype-plugin:2.2:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO] --- maven-archetype-plugin:2.2:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[INFO] Archetype defined by properties
[INFO] ----------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Archetype: appfuse-basic-spring-archetype:3.5.0
[INFO] ----------------------------------------------------------------------------
[INFO] Parameter: groupId, Value: com.raibledesigns
[INFO] Parameter: artifactId, Value: noxml
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.raibledesigns
[INFO] Parameter: packageInPathFormat, Value: com/raibledesigns
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.raibledesigns
[INFO] Parameter: groupId, Value: com.raibledesigns
[INFO] Parameter: artifactId, Value: noxml
[INFO] project created from Archetype in dir: /Users/mraible/noxml
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 0.925 s
[INFO] Finished at: 2015-09-10T23:57:18-06:00
[INFO] Final Memory: 10M/310M
[INFO] ------------------------------------------------------------------------
[mraible:~] 10s $
```

2. bash

```
[mraible:~/noxml] $ cloc .
     128 text files.
     126 unique files.
      40 files ignored.

http://cloc.sourceforge.net v 1.62  T=0.33 s (264.4 files/s, 22563.7 lines/s)
-------------------------------------------------------------------------------
Language                             files          blank        comment           code
-------------------------------------------------------------------------------
Java                                    33            567            755           2394
XML                                     25            217            138           1992
JSP                                     20             85              8            773
CSS                                      1             59             22            210
Maven                                    1             10              1             86
Velocity Template Language               5             16              0             79
Javascript                               2             14              4             76
HTML                                     1              1              0              2
-------------------------------------------------------------------------------
SUM:                                    88            969            928           5612
-------------------------------------------------------------------------------
[mraible:~/noxml] $
```

AppFuse Features

Authentication and Security

User Management

Bootstrap and jQuery Integrated

REST API with CXF

Hibernate and JPA Support

CRUD Generation

Internationalization

JHipster Features

Authentication and Security

Spring Boot Metrics Dashboard

Bootstrap and AngularJS Integrated

REST API with Spring MVC

JPA Support and NoSQL Support

CRUD Generation

Internationalization

JHipster's API Building Blocks

Spring Boot

Maven vs. Gradle

IDE Support: Running, Debugging and Profiling

Security

JPA vs. MongoDB vs. Cassandra

Liquibase

Elasticsearch

Spring WebSockets

Security

Spring Security 3.2.5

Remember Me

HttpSession Authentication

ZAP Tested

Improved Remember Me

Cookie theft protection

CSRF protection

Authentication

- HTTP Session
- Token-based
- OAuth2

JPA

Hibernate

Generic DAO

persistence.xml

Hibernate

Spring Data

Repositories

Dynamic Finders

Schema Generation and Population

Hibernate4 Plugin

DbUnit

Liquibase

CSV

Search

Hibernate Search

Elasticsearch

Hibernate Search

```
@Indexed
@XmlRootElement
public class User extends BaseObject implements Serializable, UserDetails {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @DocumentId
    public Long getId() {
        return id;
    }

    @Column(nullable = false, length = 50, unique = true)
    @Field
    public String getUsername() {
        return username;
    }
```

Hibernate Search

```
@Indexed
@XmlRootElement
public class User extends BaseObject implements Serializable, UserDetails {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @DocumentId
    public Long getId() {
        return id;
    }

    @Column(nullable = false, length = 50, unique = true)
    @Field
    public String getUsername() {
        return username;
    }
```

Elasticsearch

```
@Entity
@Table(name = "JHI_USER")
@Cache(usage = CacheConcurrencyStrategy.NONSTRICT_READ_WRITE)
@Document(indexName="user")
public class User extends AbstractAuditingEntity implements Serializable {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    @NotNull
    @Pattern(regexp = "^[a-z0-9]*$")
    @Size(min = 1, max = 50)
    @Column(length = 50, unique = true, nullable = false)
    private String login;
```

Elasticsearch

```
@Entity
@Table(name = "JHI_USER")
@Cache(usage = CacheConcurrencyStrategy.NONSTRICT_READ_WRITE)
@Document(indexName="user")
public class User extends AbstractAuditingEntity implements Serializable {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    @NotNull
    @Pattern(regexp = "^[a-z0-9]*$")
    @Size(min = 1, max = 50)
    @Column(length = 50, unique = true, nullable = false)
    private String login;
```

WebSockets

None

Spring WebSockets

Stomp

SockJS

WebSockets

```
@Configuration
@EnableWebSocketMessageBroker
public class WebsocketConfiguration extends AbstractWebSocketMessageBrokerConfigurer {

    @Override
    public void configureMessageBroker(MessageBrokerRegistry config) {
        config.enableSimpleBroker("/topic");
    }

    @Override
    public void registerStompEndpoints(StompEndpointRegistry registry) {
        registry.addEndpoint("/websocket/tracker")
            .setHandshakeHandler(new DefaultHandshakeHandler() {
                //
            })
            .withSockJS()
            .setInterceptors(httpSessionHandshakeInterceptor());
    }
```

yo jhipster

Installing JHipster

Local installation vs. Docker

We provide 2 ways of working with JHipster:

- You can do a "local installation", which is the classical way of working with JHipster. Everything is installed on your machine, which can be a little complex to set up, but that's how most people usually work. In case of doubt, choose this installation.
- You can use a "Docker" container, which brings you a lightweight, virtualized container with everything already installed. You don't have to worry about installing everything. But Docker is a rather young tool, and you need to know how to use Linux quite well.

Local installation (recommended for normal users)

1. Install Java from the Oracle website.
2. Install Maven if you want to use Maven as the build tool. If you prefer to use Gradle instead, don't install it, as JHipster ships with the Gradle Wrapper.
3. Install Git from git-scm.com. We recommend you also use a tool like SourceTree if you are starting with Git.
4. Install Node.js from the Node.js website. This will also install `npm`, which is the node package manager we are using in the next commands.
5. Install Yeoman: `npm install -g yo`
6. Install Bower: `npm install -g bower`
7. Depending on your preferences, install either Grunt (recommended) with `npm install -g grunt-cli` or Gulp.js with `npm install -g gulp`.
8. Install JHipster: `npm install -g generator-jhipster`

To find more information, tips and help, please have a look at the Yeoman "getting starting" guide and at the NPM documentation before submitting a bug.

2. node

```
Welcome to the JHipster Generator v2.20.0

? (1/15) What is the base name of your application? noxml
? (2/15) What is your default Java package name? com.raibledesigns
? (3/15) Do you want to use Java 8? Yes (use Java 8)
? (4/15) Which *type* of authentication would you like to use? HTTP Session Authentication (stateful, default Spring Sec
urity mechanism)
? (5/15) Which *type* of database would you like to use? SQL (H2, MySQL, PostgreSQL, Oracle)
? (6/15) Which *production* database would you like to use? (Use arrow keys)
? (6/15) Which *production* database would you like to use? MySQL
? (7/15) Which *development* database would you like to use? H2 in-memory with Web console
? (8/15) Do you want to use Hibernate 2nd level cache? Yes, with ehcache (local cache, for a single node)
? (9/15) Do you want to use a search engine in your application? Yes, with ElasticSearch
? (10/15) Do you want to use clustered HTTP sessions? No
? (11/15) Do you want to use WebSockets? No
? (12/15) Would you like to use Maven or Gradle for building the backend? Gradle
? (13/15) Would you like to use Grunt or Gulp.js for building the frontend? Grunt (recommended)
? (14/15) Would you like to use the LibSass stylesheet preprocessor for your CSS? No
? (15/15) Would you like to enable translation support with Angular Translate? Yes
```

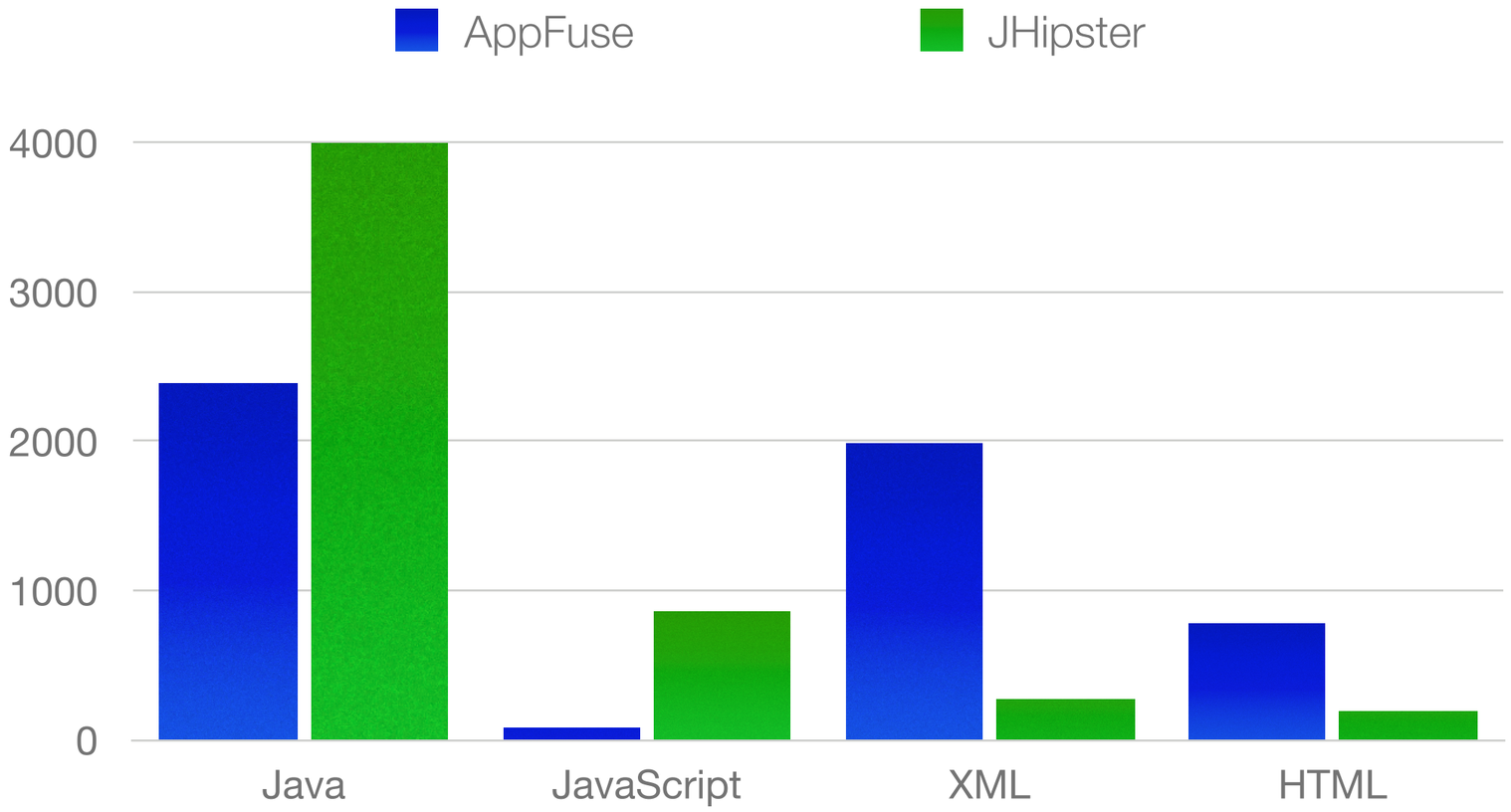
2. bash

```
Running "ngconstant:dev" (ngconstant) task
Creating module noxmlApp at src/main/webapp/scripts/app/app.constants.js...OK

Running "wiredep:app" (wiredep) task

Running "wiredep:test" (wiredep) task

Done, without errors.


Execution Time (2015-09-11 06:55:40 UTC)
loading tasks     3ms  ▌ 3%
ngconstant:dev    9ms  ▌ 8%
wiredep:app      95ms  ▌79%
wiredep:test     12ms  ▌10%
Total 120ms

[mraible:~/noxml] 2m56s $
```

2. bash

```
[mraible:~/noxml] $ cloc .
     152 text files.
     149 unique files.
      26 files ignored.

http://cloc.sourceforge.net v 1.62  T=0.50 s (246.0 files/s, 15584.1 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
Java                            95            905            630           4000
Javascript                       9            115            100            861
XML                              6             51             26            271
YAML                             5             27              4            209
HTML                             3             22              0            188
JSON                             3              0              0            128
Bourne Again Shell               1             20             21            123
DOS Batch                        1             24              2             64
-------------------------------------------------------------------------------
SUM:                           123           1164            783           5844
-------------------------------------------------------------------------------
[mraible:~/noxml] $
```

Lines of Code Comparison



2. bash

```
[mraible:~/noxml] $ cd ~/noxml.appfuse/
[mraible:~/noxml.appfuse] $ find . -name '*.xml'
./pom.xml
./src/main/resources/applicationContext-resources.xml
./src/main/resources/default-data.xml
./src/main/resources/ehcache.xml
./src/main/resources/hibernate.cfg.xml
./src/main/resources/log4j2.xml
./src/main/resources/META-INF/persistence.xml
./src/main/webapp/WEB-INF/applicationContext-validation.xml
./src/main/webapp/WEB-INF/applicationContext.xml
./src/main/webapp/WEB-INF/cxf-servlet.xml
./src/main/webapp/WEB-INF/decorators.xml
./src/main/webapp/WEB-INF/dispatcher-servlet.xml
./src/main/webapp/WEB-INF/dwr.xml
./src/main/webapp/WEB-INF/menu-config.xml
./src/main/webapp/WEB-INF/resin-web.xml
./src/main/webapp/WEB-INF/security.xml
./src/main/webapp/WEB-INF/sitemesh.xml
./src/main/webapp/WEB-INF/urlrewrite.xml
./src/main/webapp/WEB-INF/validation.xml
./src/main/webapp/WEB-INF/validator-rules-custom.xml
./src/main/webapp/WEB-INF/validator-rules.xml
./src/main/webapp/WEB-INF/web.xml
./src/main/webapp/WEB-INF/wro.xml
./src/site/site.xml
./src/test/resources/sample-data.xml
./src/test/resources/web-tests.xml
[mraible:~/noxml.appfuse] $
```

XML to Feature Map

Build

Spring Config

DbUnit

EhCache

Hibernate

Log4J2

JPA

Validation

CXF

SiteMesh

Spring MVC

DWR

Struts Menu

Resin Web Config

Spring Security

UrlRewrite

Commons Validator

web.xml

Wro4j

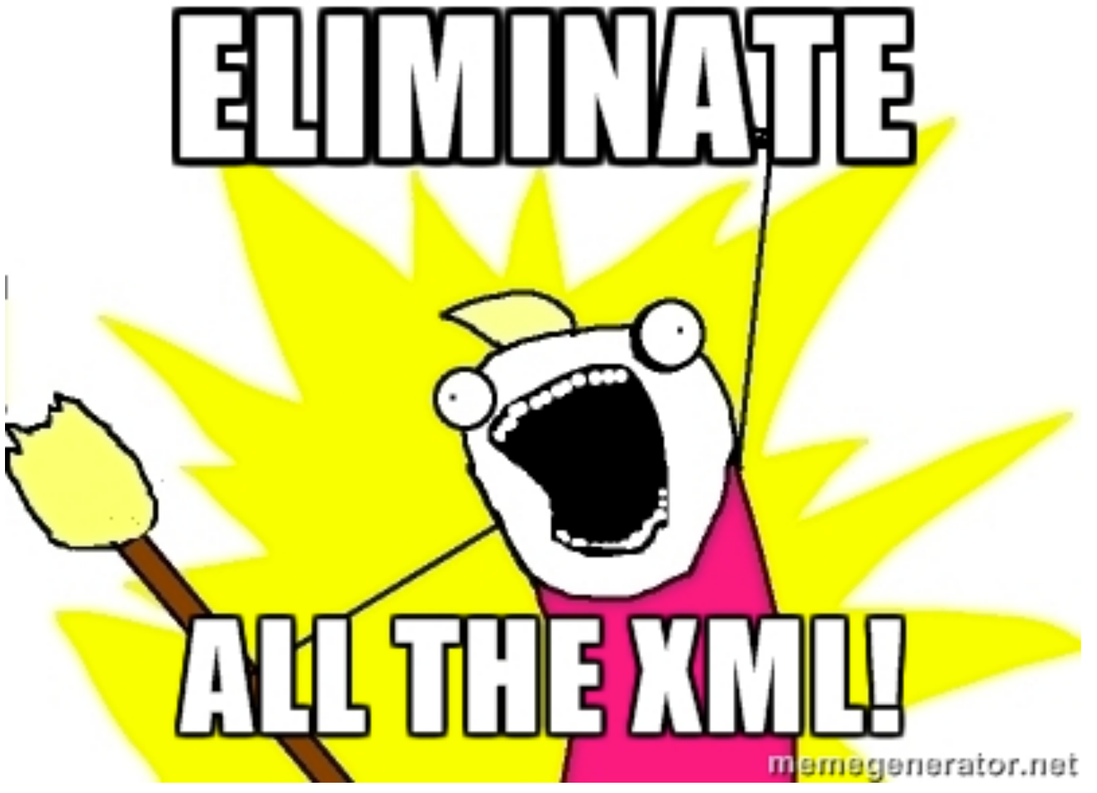
Generated documentation

Integration Tests

How do we get rid of these?

Maven	Gradle or Polyglot for Maven
Spring XML	JavaConfig
DbUnit	DataSetBuilder
EhCache	@EnableCaching
Hibernate	Spring Data
Log4J2	Configurator#initialize
Validation	JSR 303
web.xml	WebApplicationInitializer
Maven Site	README.md

ALL THE
THINGS!


ELIMINATE
ALL THE XML!
memegenerator.net

Live Demo

Recap

Spring XML to Java

Spring Security XML to Java

web.xml to WebApplicationInitializer

Spring MVC to Java

Migrated to Spring Boot

Greetings, Java Hipster!

JHipster is a
Yeoman generator,

used to create a
Spring Boot + **AngularJS**
project.

Discuss: How can things be even better?

Questions?

Stay hip by following me!

http://raibledesigns.com

@mraible

Presentations

http://slideshare.net/mraible

Code

https://github.com/mraible/appfuse-noxml