

KEEPING IT CLEAN

LESSONS LEARNED FROM OPEN SOURCE ON HOW TO POLISH CODE

```
$ git log -i --grep=polish --pretty=format:"%an" \  
| sort | uniq -c | sort -nr
```

410	Juergen Hoeller
192	Sam Brannen
182	Keith Donald
126	Chris Beams
119	Rossen Stoyanchev
67	Phillip Webb
23	Stephane Nicoll
8	Thomas Risberg
7	Arjen Poutsma
5	Brian Clozel
3	Sebastien Deleuze
3	Andy Clement
2	Scott Andrews
2	Rob Winch
2	Mark Pollack
1	Vasyl Tretiakov
1	Costin Leau
1	Ben Hale
1	Andy Wilkinson

Oliver Gierke on Twitter: "Yay, new working clothes arrived... #jürgenized #SpringSource" <https://twitter.com/olivergierke/>

Home Notifications Messages Discover Search Twitter

Oliver Gierke @olivergierke Following

Yay, new working clothes arrived...
#jürgenized #SpringSource

Mayence, Rhineland-Palatinate



RETWEETS FAVORITES
6 1

2:54 PM - 18 Aug 2012

SORRY
TO EVERYONE WHO'S CODE I'VE MEDDLED WITH

WHY CLEAN CODE?

CRAFTSMANSHIP?



NOPE!

THREE KINDS OF CODE

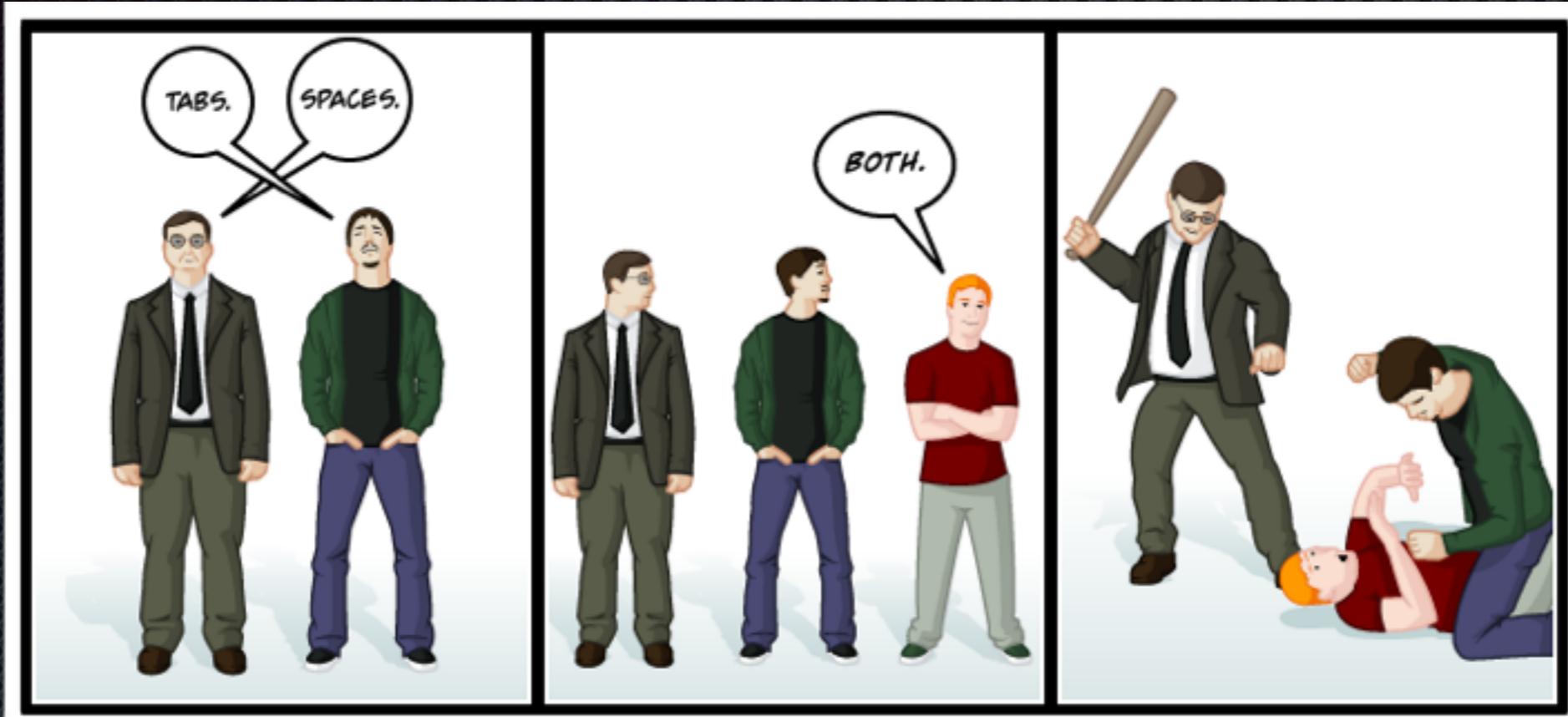
- NEVER USED
- CONTAINS BUGS
- NEEDS TO EVOLVE

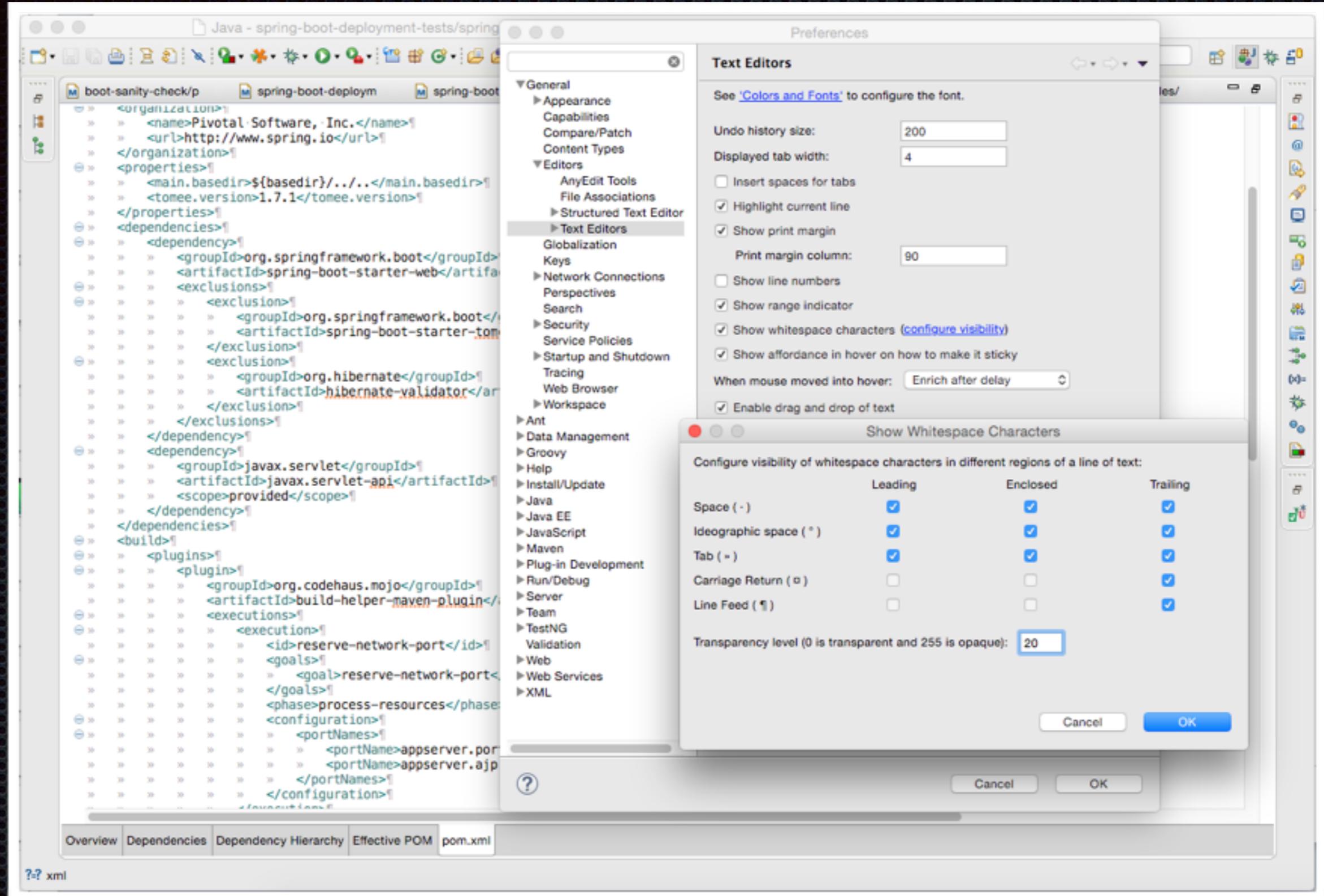
reduce cognitive
overhead so that code
can be understood
easily .

OVERVIEW

1. SIMPLE STUFF
2. HARDER STUFF

1. SIMPLE STUFF

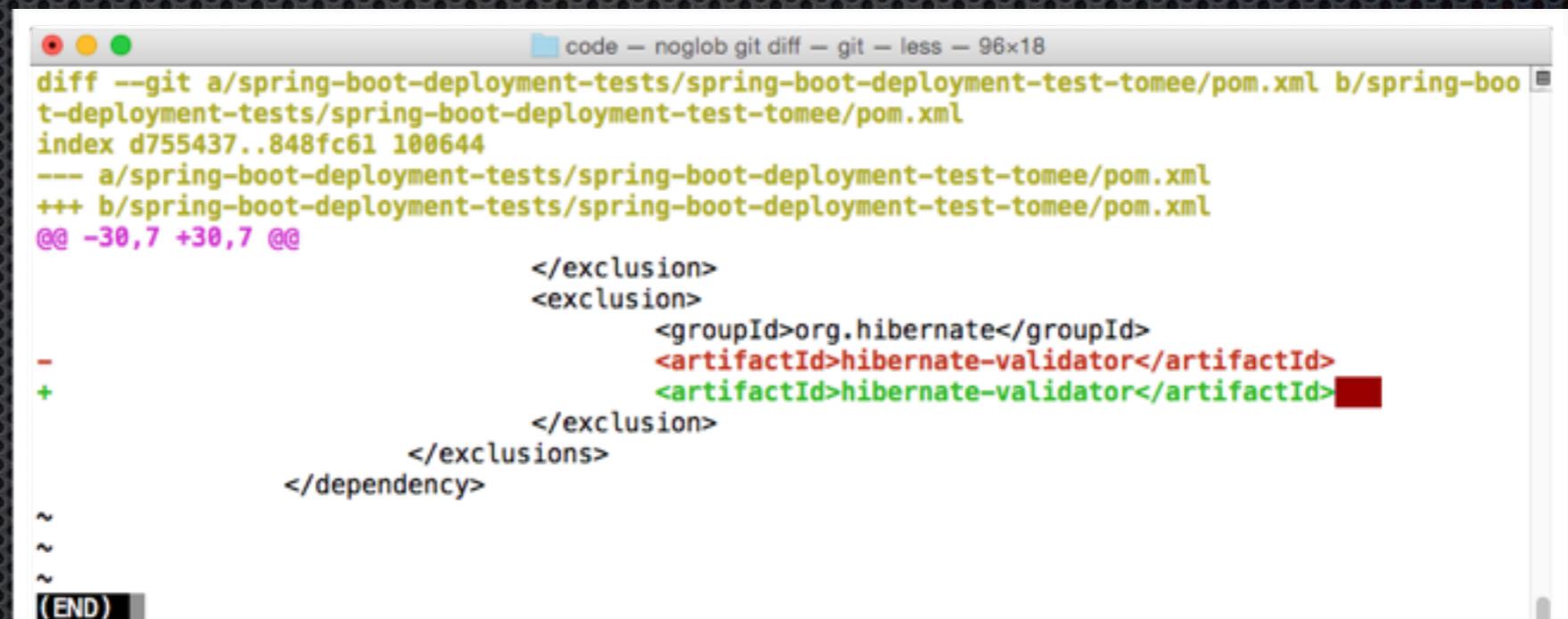




GIT COLOR

~/.gitconfig

```
[color]
  ui = auto
```



A screenshot of a terminal window titled "code — noglob git diff — git — less — 96x18". The window displays a git diff output between two XML files: pom.xml from branches 'a' and 'b'. The diff shows several changes, including the addition of a dependency exclusion for 'hibernate-validator'. The terminal interface includes standard window controls (red, yellow, green circles) and a scroll bar.

```
diff --git a/spring-boot-deployment-tests/spring-boot-deployment-test-tomee/pom.xml b/spring-boot-deployment-tests/spring-boot-deployment-test-tomee/pom.xml
index d755437..848fc61 100644
--- a/spring-boot-deployment-tests/spring-boot-deployment-test-tomee/pom.xml
+++ b/spring-boot-deployment-tests/spring-boot-deployment-test-tomee/pom.xml
@@ -30,7 +30,7 @@
                               </exclusion>
                               <exclusion>
                                   <groupId>org.hibernate</groupId>
-                                  <artifactId>hibernate-validator</artifactId>
+                                  <artifactId>hibernate-validator</artifactId>
                               </exclusion>
                           </exclusions>
                       </dependency>
~
```

(END)

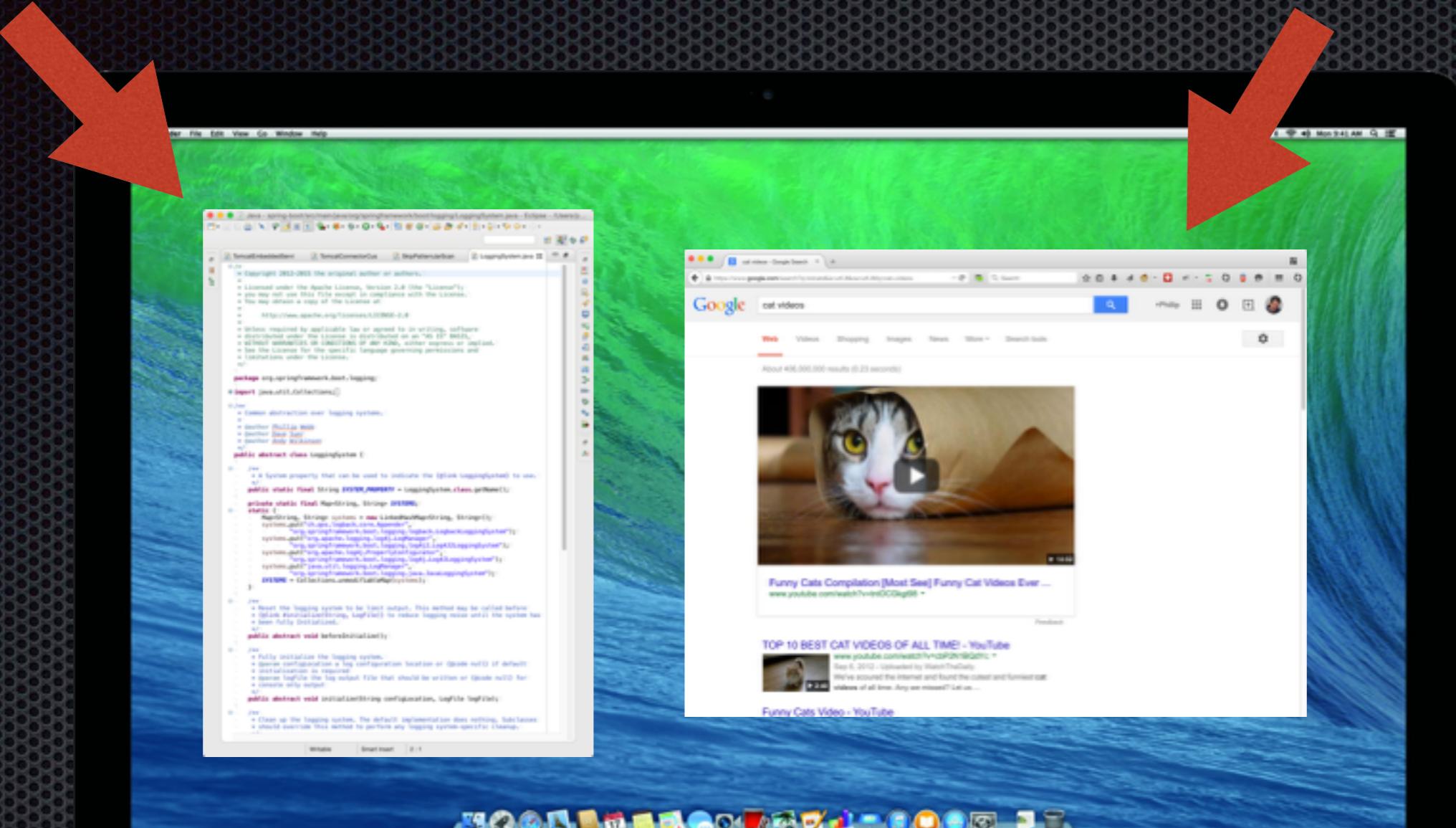
SPRING BOOT CONVENTIONS

- PRETTY SIMILAR TO THE “GOOGLE JAVA STYLE”
- TABS NOT SPACES
- NO ‘.*’ IMPORTS
- ALWAYS USE BRACES **(NO SINGLE LINE IFS)**
- ALWAYS USE ‘THIS.’ WHEN REFERENCING PRIVATE FIELDS
- WRAP AT 90 CHARS

WRAP AT 90 CHARS ?!!

CODE

CAT VIDEOS



CONSTRAINTS ARE GOOD

- IT LOOKS GOOD ON GITHUB
- IT'S READABLE ON A PHONE
- IT WORKS IN YOUR TERMINAL
- IT MAKES YOU WRITE BETTER CODE

AUTOMATIC FORMATTING

- SPRING FRAMEWORK IS MANUALLY FORMATTED
- SPRING BOOT IS AUTO-FORMATTED
- BOTH HAVE PROJECT SPECIFIC ECLIPSE CONFIGURATION

MAVEN

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-eclipse-plugin</artifactId>
  <configuration>
    <additionalConfig>
      <file>
        <name>.settings/org.eclipse.jdt.ui.prefs</name>
        <location>${main.basedir}/eclipse/org.eclipse.jdt.ui.prefs</location>
      </file>
      <file>
        <name>.settings/org.eclipse.jdt.core.prefs</name>
        <location>${main.basedir}/eclipse/org.eclipse.jdt.core.prefs</location>
      </file>
    </additionalConfig>
  </configuration>
</plugin>
```

GRADLE

```
// Include project specific settings
task eclipseSettings(type: Copy) {
    from rootProject.files(
        "src/eclipse/org.eclipse.jdt.ui.prefs",
        "src/eclipse/org.eclipse.wst.common.project.facet.core.xml")
    into project.file('.settings/')
    outputs.upToDateWhen { false }
}

tasks["eclipse"].dependsOn(eclipseSettings)
```

DEMO

AUTO FORMATTING IN ECLIPSE

ORDERING

- CONSTANTS AT THE TOP OF THE FILE
- INNER CLASSES AT THE END
- PRIVATE VARIABLES IN ORDER OF PRIORITY
 - MATCH CONSTRUCTOR ARGUMENTS FIRST
 - OPTIONAL SETTERS LATER
- READ DOWN ORDERING FOR METHOD NAMES

EXAMPLE

UndertowEmbeddedServletContainer

// COMMENTS



Michael Kozierski
@nzkoz



Follow

Code comments



RETWEETS
4,605

FAVORITES
2,394



7:10 PM - 29 Nov 2014

Don't trust anyone that
tells you not to
comment your code!

CODE COMMENTS

```
/**  
 * This is probably good.  
 */
```

```
// This is probably bad!
```

COMMENTS

- DOCUMENT WHY IT EXISTS
- DOCUMENT WHAT IT DOES
- LEAVE BREADCRUMBS
- DON'T DOCUMENT HOW IT DOES IT

EXAMPLE

EmbeddedServletContainer

2. HARDER STUFF

TESTS

- TESTS GIVE YOU THE CONFIDENCE TO CHANGE CODE
- ADD TESTS BEFORE CHANGING CODE
- CHANGE CODE INDEPENDENTLY OF TESTS
- IMAGINE YOUR TESTS NAMES START ‘SHOULD’...
- USE GIVEN/WHEN/THEN FORMAT
- ADD THE CLASS UNDER TEST IN THE DOC-COMMENTS

TESTS

- USE @RULES
 - OutputCapture
 - TemporaryFolder
 - ExpectedException
- USE assertThat(...)
- USE BDD MOCKITO

EXAMPLE

TomcatEmbeddedServletContainerFactoryTests

CONSISTENCY

- EVEN WITH AUTO-FORMATTING IT CAN BE HARD TO KEEP CODE CONSISTENT
 - WHITESPACE, HEADERS, JAVADOC, NAMES
- IT'S OFTEN EASIER TO APPLY AFTER A CHANGE HAS BEEN MADE

RANDOM FILE GAME

```
$ brew install coreutils
```

```
$ find . -type f | grep '\.java' | gshuf | head -n1
```

NAMING

NAMING ADVICE...



**FROM THE ABSTRACT SINGLETON
~~PROXY~~ FACTORY BEAN PEOPLE?**

БЫКОМ ИНОДЬ ВЕНИКЪ БЫТЬ

IN DEFENSE OF ASPFB

- YES YES, HILARIOUS NAME SINCE **2006**, BUT...
- IT'S SOMEWHAT INTERNAL
- ALL IMPLEMENTATIONS OF FACTORYBEAN IN SPRING
HAVE A NAME ENDING '**FACTORYBEAN**'
- MOST ABSTRACT CLASSES THAT YOU EXTEND IN SPRING
START WITH '**ABSTRACT**'
- IT'S FOR CREATING SINGLETON-SCOPED PROXIES

WE'VE DONE WORSE

HasThisTypePatternTriedToSneakInSomeGenericOrParameterizedTypePatternMatchingStuffAnywhereVisitor

NAMING

- USE YOUR TESTS TO GUIDE YOU
- USE JAVADOC TO GUIDE YOU
- USE “**DUNNO**” WHILST YOU THINK OF A NAME
- CONSIDER THE CLASS NAME WHEN NAMING STATIC METHODS
- **E.G.** `Assert.assertEquals(...)` **RATHER THAN** `assertNotEquals(...)`

REFACTORING

EXTRACT METHOD

- USE “**EXTRACT METHOD**” TO AID READABILITY
 - LINE COMMENTS ARE OFTEN A SIGN
 - WHITESPACE IS OFTEN A SIGN
 - UNSET VARIABLES ARE OFTEN A SIGN
- ```
String foo;
if (this.x) {
 foo = "x";
} else {
 foo = "y";
}
...
```

# PARAMETER OBJECTS

- EXTRACT METHOD PARAMETERS IF
  - THERE ARE TOO MANY OF THEM
  - THEY ARE LIKELY TO GROW

# EXAMPLE

```
package org.springframework.context.annotation

public interface Condition {

 boolean matches(BeanDefinitionRegistry registry,
 ConfigurableListableBeanFactory beanFactory,
 Environment environment,
 ResourceLoader resourceLoader,
 ClassLoader classLoader,
 AnnotatedTypeMetadata metadata);

 // vs

 boolean matches(ConditionContext context,
 AnnotatedTypeMetadata metadata);
}
```

# EXTRACT INNER CLASS

- TOO MUCH “EXTRACT METHOD”
  - CLUSTER OF RELATED METHODS
  - THE SAME PARAMETERS AGAIN AND AGAIN
- ALSO CONSIDER CREATING A PACKAGE PRIVATE CLASS

ConfigFileApplicationListener from Spring Boot is a good example

# PACKAGE TANGLES

- SERIOUSLY HINDER FUTURE REFACTORING
- YOU WON'T FIND THEM UNTIL IT'S TOO LATE
- USE A TOOL
  - STRUCTURE 101
  - SONARQUBE

# THE HIDDEN DEBUGGER

## GIT HISTORY

# COMMIT LOGS

- COMMIT LOGS ARE AN AMAZING DEBUGGING TOOL
- ECLIPSE: TEAM/SHOW ANNOTATIONS
- git blame
- git log -- /some/file.java
- KEEP CONSISTENCY IN YOUR LOGS
- TELL A STORY

# COMMIT LOGS

- FIRST LINE CAPITALIZED, SHORT (50 CHARS OR LESS)
- SECOND LINE BLANK
- MESSAGE WRAPPED AT 72 CHARS
- IMPERATIVE FORM “FIX BUG” NOT “FIXED”/“FIXING”
- REFERENCE TO A ISSUE NUMBER

[tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html](http://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html)

[chris.beams.io/posts/git-commit](http://chris.beams.io/posts/git-commit)

# WHY?

- EASY TO READ IN THE TERMINAL.
- EASY TO READ IN THE IDE
- LOOKS GOOD ON [GITHUB.COM](#)
- IF YOU CAN'T DESCRIBE A COMMIT IN 50 CHARS, PERHAPS IT SHOULD BE TWO COMMITS!

commit 12724bf33211d68899d710fda376bd5636731e4c

Author: Dave Syer <dsyer@pivotal.io>

Date: Thu Feb 12 10:09:19 2015 +0000

Fix ordering of keys in PropertySourcesPropertyValues

Since @ConfigurationProperties binding uses a single instance of PropertySourcesPropertyValues per bean, there doesn't seem to be any issue with using a normal LinkedHashMap. Then the order passed in as PropertySources will be preserved.

Fixes gh-2487

commit 0ef3de4d82acc32a3f44d872e852d94feb8cd5da

Author: Andy Wilkinson <awilkinson@pivotal.io>

Date: Wed Feb 11 13:15:52 2015 +0000

Document how to disable auto registration of a Servlet or Filter bean

Closes gh-2173

commit b18330972884e8cb6832b0bdbd6c9f545dc1f501

Author: Andy Wilkinson <awilkinson@pivotal.io>

Date: Tue Feb 10 20:14:38 2015 +0000

Stop a BeanPostProcessor from preventing config of packages to scan

Previously, if a BeanPostProcessor bean was declared in a configuration class that depended on the persistence context, packages to scan would not be configured on the LocalContainerEntityManagerFactoryBean. This was due to the BeanPostProcessor bean triggering the early instantiation of the LCEMFB before EntityScanBeanPostProcessor, the BeanPostProcessor, that applies the @EntityScan configuration, had been given a chance to configure it.

This commit updates EntityScanBeanPostProcessor to implement Ordered with an order of zero. This ensures that its ordering is predictable and that it will be driven before any unordered BeanPostProcessor. This means that, unless the user has specifically ordered their BeanPostProcessor to run before EntityScanBeanPostProcessor, LCEMFB's packages to scan will be configured as expected.

Fixes gh-2478

commit 1035e5b02967d68f4ae607e11a7bf7c67e95af72

Author: Stephane Nicoll <snicoll@pivotal.io>

Date: Tue Feb 10 16:00:11 2015 +0100

Expose RepositoryRestMvcBootConfiguration

If an application defines a custom RepositoryRestMvcConfiguration, all Spring Boot defaults are lost. While this sounds sensible, it can be confusing as Spring Boot exposes properties (`spring.data.rest.\*`) that are no longer honored.

# GIT COMMIT TEMPLATE

~/.gitconfig

```
[core]
 editor = subl -w

[commit]
 template = ~/.gitmessage.txt
```

~/.gitmessage.txt

```
#=====
#=====

Fixes gh-
Use the template above to compose your commit message. Keep the 1st
line summary 50 chars or less and the body a maximum 72 chars per line
```

# GIT REBASE

```
git checkout -b gh-1234
git commit -m 'something'
git commit -m 'something else'
git commit -m 'unrelated polish'
git rebase -i master
```

# **FINAL COMPLETE EXAMPLE**

UndertowEmbeddedServletContainer

# UNDERTOW SUPPORT

- CONTRIBUTION TO BOOT 1.2 BY IVAN SOPOV
  - INITIAL POLISH TO FOLLOW BOOT CONVENTIONS 1864d79
  - WHITESPACE, EXCEPTIONS, EXTRACT METHODS a641f0c
- FURTHER COMMITS ADD FEATURES OR FIX ISSUES

**THANKS!**  
**QUESTIONS?**

@philip\_webb  
[speakerdeck.com/philwebb](http://speakerdeck.com/philwebb)  
[github.com/philwebb](https://github.com/philwebb)