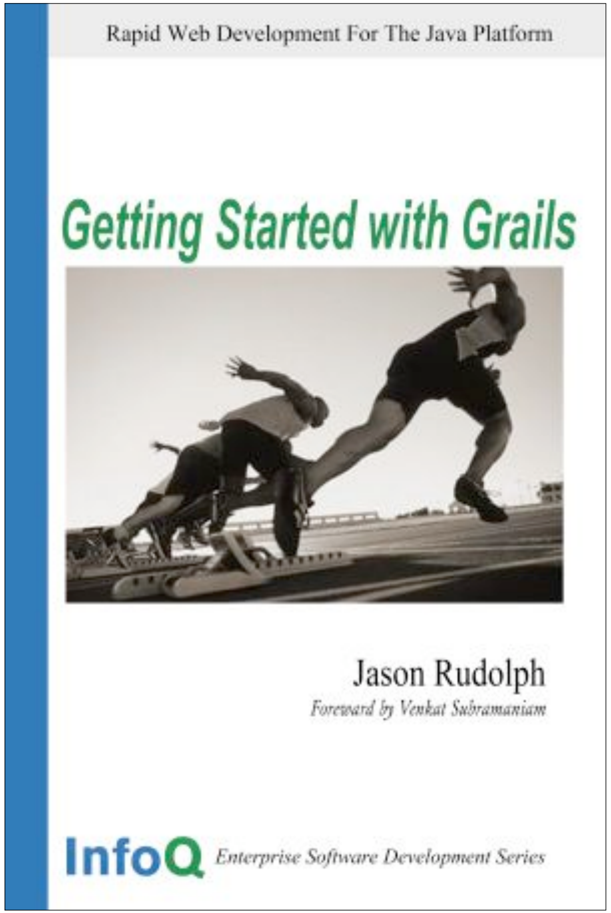
Refactotum #6

2G Experience

Jason Rudolph
jason@thinkrelevance.com

Introduction



Rapid Web Development For The Java Platform

Getting Started with Grails

Jason Rudolph

Foreward by Venkat Subramaniam

InfoQ *Enterprise Software Development Series*

- Jason Rudolph
- jasonrudolph.com
- thinkrelevance.com

refactor |rēˈfaktər|

verb

to improve a program’s readability or simplify its structure without changing its results; sometimes informally referred to as "cleaning it up"

http://en.wikipedia.org/wiki/refactor

relevance

factotum |fakˈtōtəm|

noun (pl. -tums)
a person having many diverse activities or responsibilities
: *he was employed as the general factotum.*

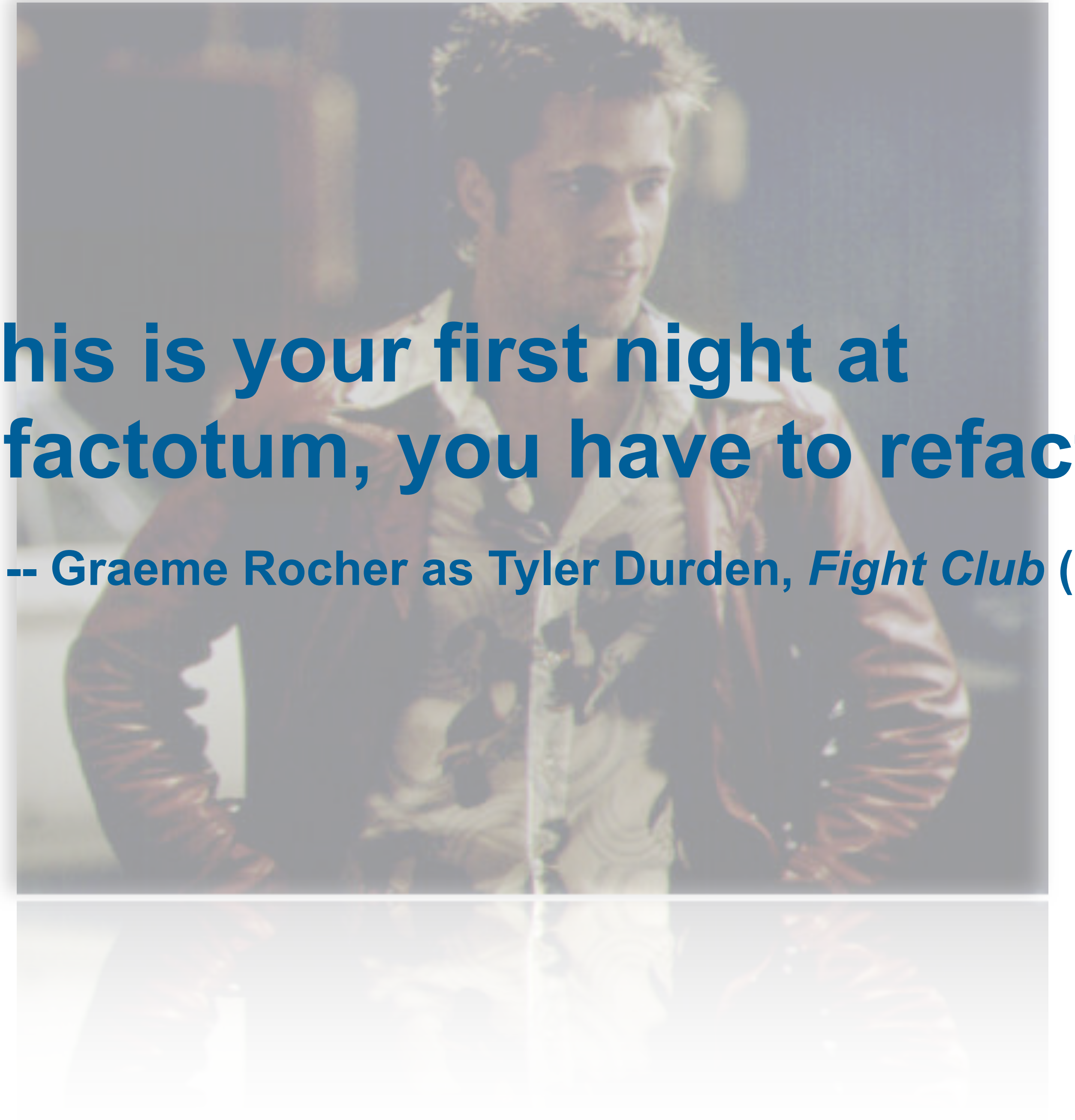
http://www.encyclopedia.com/doc/1O999-factotum.html

How to be a Refactotum

1. Set aside four hours
2. Choose an open source project
3. Get edge
4. Get metrics
5. Analyze
6. Contribute
7. Submit

Refactotum History

- Typo - eRubyCon 2007
- JRuby - No Fluff, Just Stuff
- ActiveRecord
- YUIDomCollapse - The Ajax Experience 2007
- Rake - RubyConf 2007
- Grails - 2G Experience



If this is your first night at Refactotum, you have to refactor.

-- Graeme Rocher as Tyler Durden, *Fight Club* (almost)

Up and Running with Grails Source

- **Go to grails.org/installation**
- **Get the SVN bits**
- **Verify tests pass**
  - ant test
- **Use Cobertura to hunt for suspicious code**
  - ant test-with-coverage

Build Noise

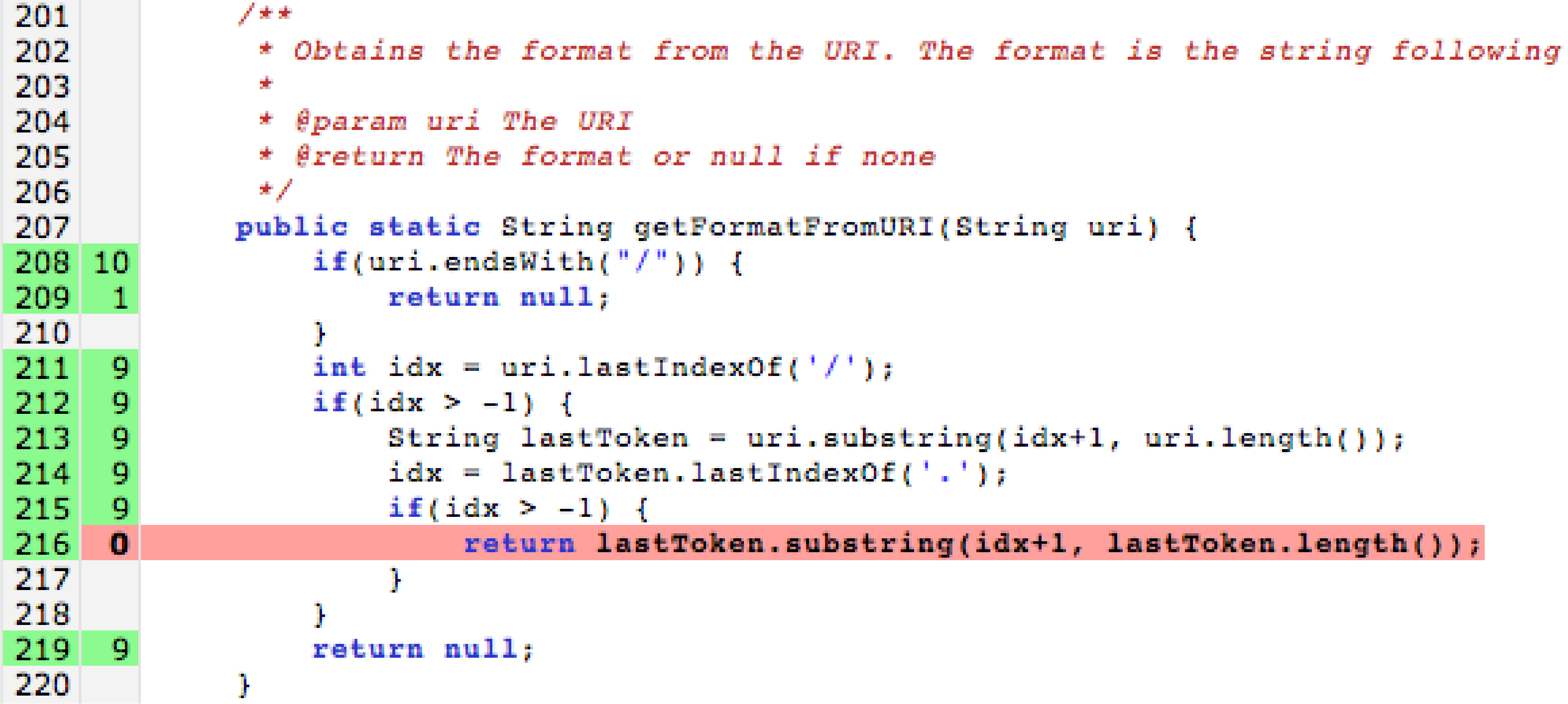
```
[groovyc] .../grails/src/web/org/codehaus/groovy/grails/web/
  servlet/mvc/GrailsHttpSession.java:21: warning:
  [deprecation] javax.servlet.http.HttpSessionContext in
  javax.servlet.http has been deprecated
[groovyc] import javax.servlet.http.HttpSessionContext;
[groovyc]                           ^
...
[groovyc] .../grails/src/commons/org/codehaus/groovy/grails/
  commons/DefaultGrailsDomainClassProperty.java:487: warning:
  [deprecation] getPersistantProperties() in
  org.codehaus.groovy.grails.commons.GrailsDomainClass has
  been deprecated
[groovyc] public GrailsDomainClassProperty[] getPersistantProperties()
[groovyc]                                    ^
...
[groovyc] 3 warnings
...
[groovyc] 6 warnings
```

relevance

Shhhhh

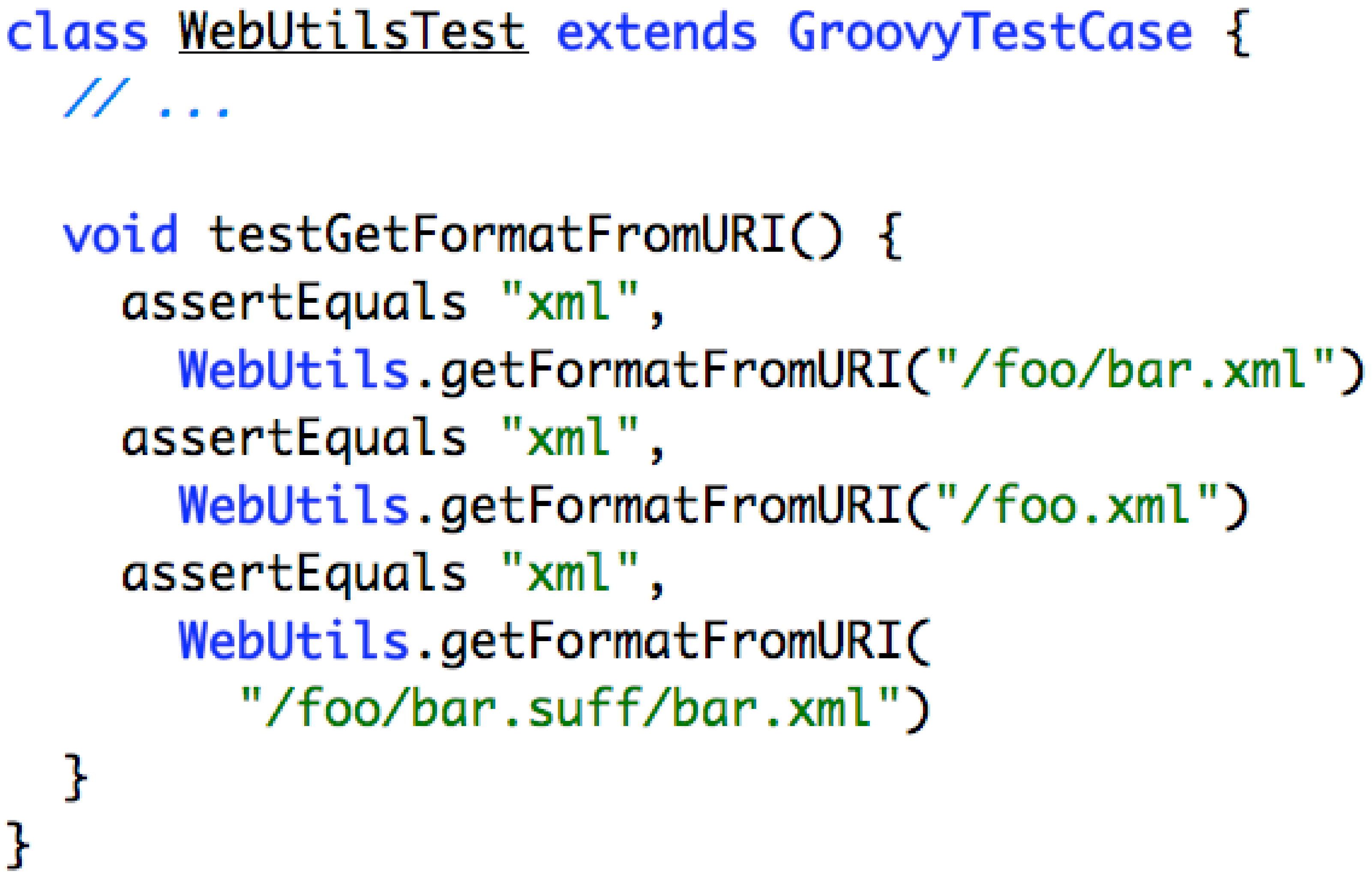
Demo

Are You Covered?



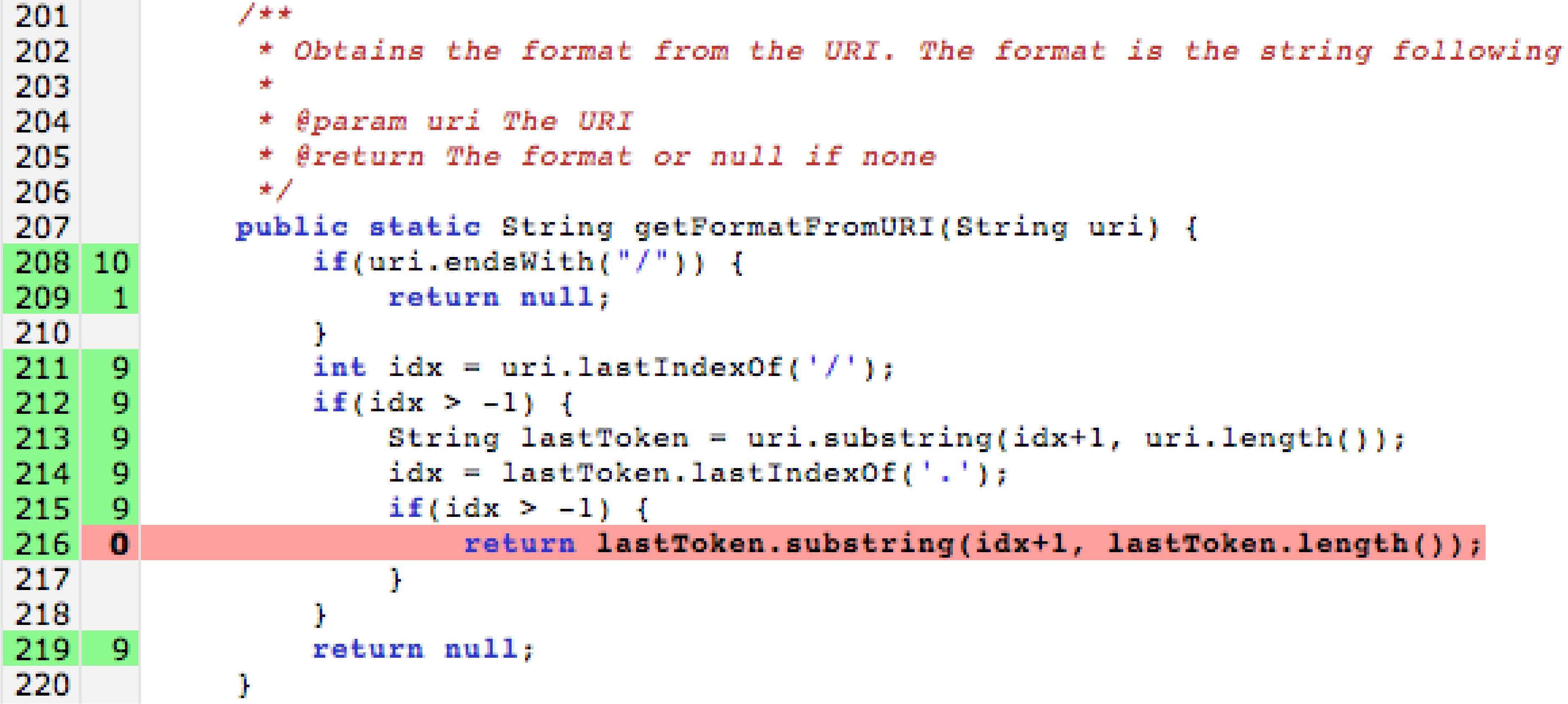
```
        /**
         * Obtains the format from the URI. The format is the string following
         *
         * @param uri The URI
         * @return The format or null if none
         */
        public static String getFormatFromURI(String uri) {
10          if(uri.endsWith("/")) {
 1              return null;
            }
 9          int idx = uri.lastIndexOf('/');
 9          if(idx > -1) {
 9              String lastToken = uri.substring(idx+1, uri.length());
 9              idx = lastToken.lastIndexOf('.');
 9              if(idx > -1) {
 0                  return lastToken.substring(idx+1, lastToken.length());
                }
            }
 9          return null;
        }
```

Shouldn't This Do the Trick?



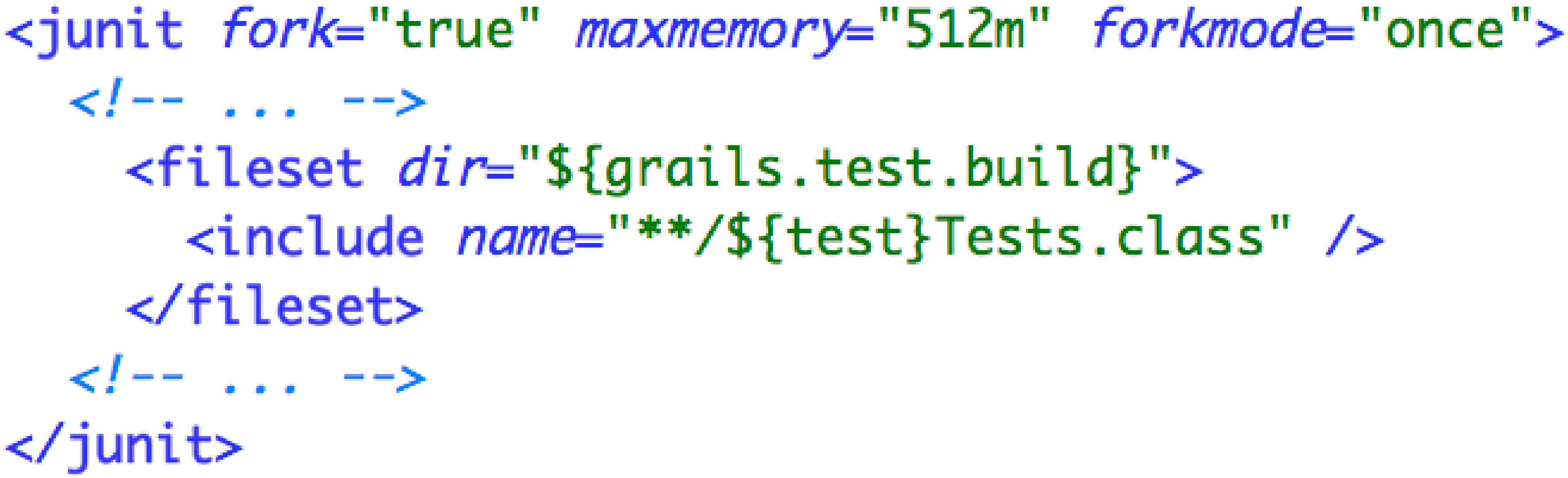
```
class WebUtilsTest extends GroovyTestCase {
  // ...

  void testGetFormatFromURI() {
    assertEquals "xml",
      WebUtils.getFormatFromURI("/foo/bar.xml")
    assertEquals "xml",
      WebUtils.getFormatFromURI("/foo.xml")
    assertEquals "xml",
      WebUtils.getFormatFromURI(
        "/foo/bar.suff/bar.xml")
  }
}
```

What Gives?



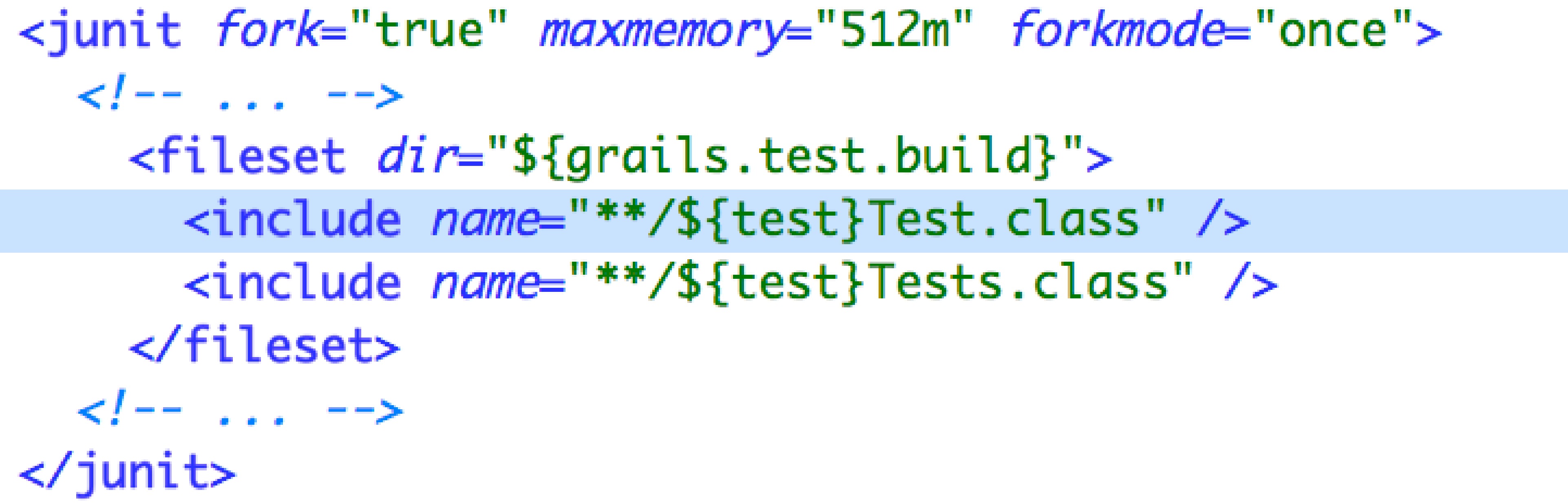
```
        /**
         * Obtains the format from the URI. The format is the string following
         *
         * @param uri The URI
         * @return The format or null if none
         */
        public static String getFormatFromURI(String uri) {
10          if(uri.endsWith("/")) {
 1              return null;
            }
 9          int idx = uri.lastIndexOf('/');
 9          if(idx > -1) {
 9              String lastToken = uri.substring(idx+1, uri.length());
 9              idx = lastToken.lastIndexOf('.');
 9              if(idx > -1) {
 0                  return lastToken.substring(idx+1, lastToken.length());
                }
            }
 9          return null;
        }
```

Testing. Testing. Is This Thing On?



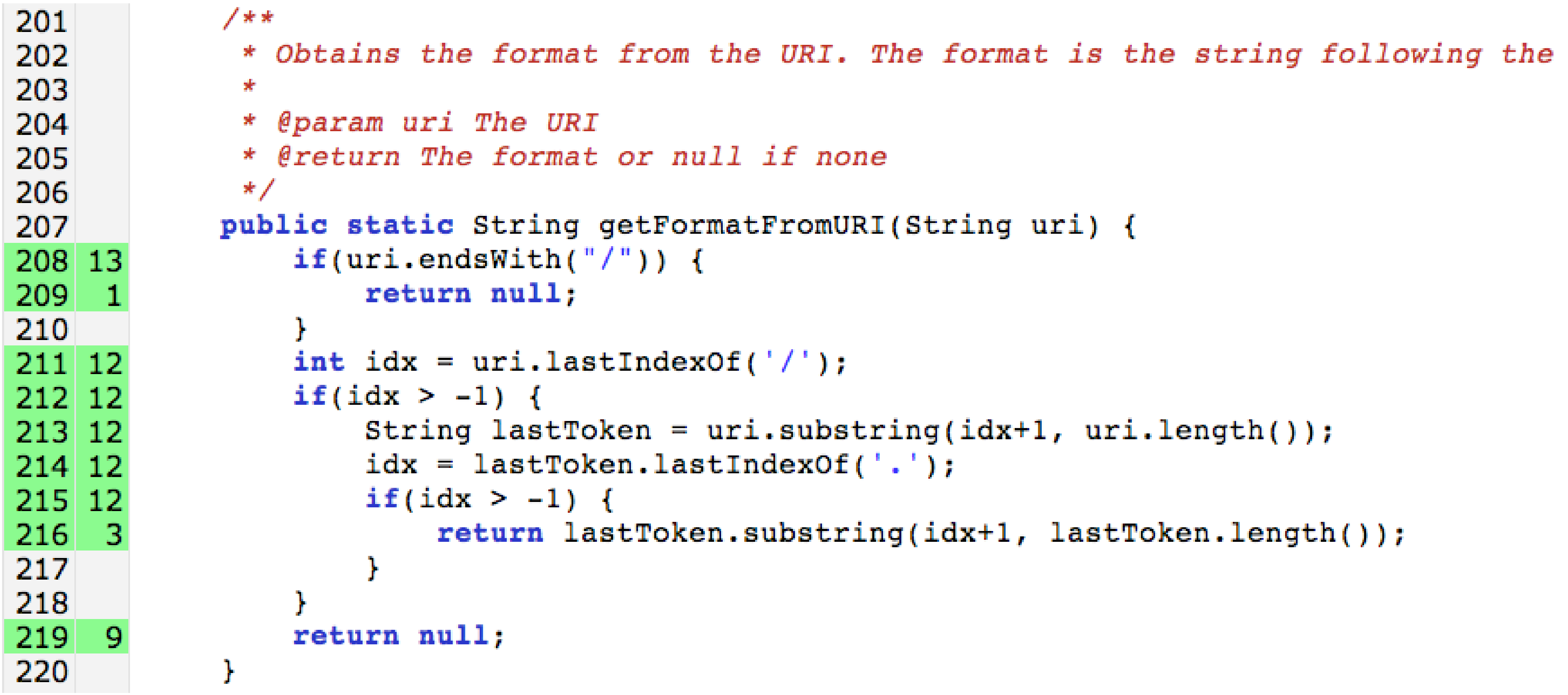
```
<junit fork="true" maxmemory="512m" forkmode="once">
  <!-- ... -->
    <fileset dir="${grails.test.build}">
      <include name="**/${test}Tests.class" />
    </fileset>
  <!-- ... -->
</junit>
```

Being Inclusive



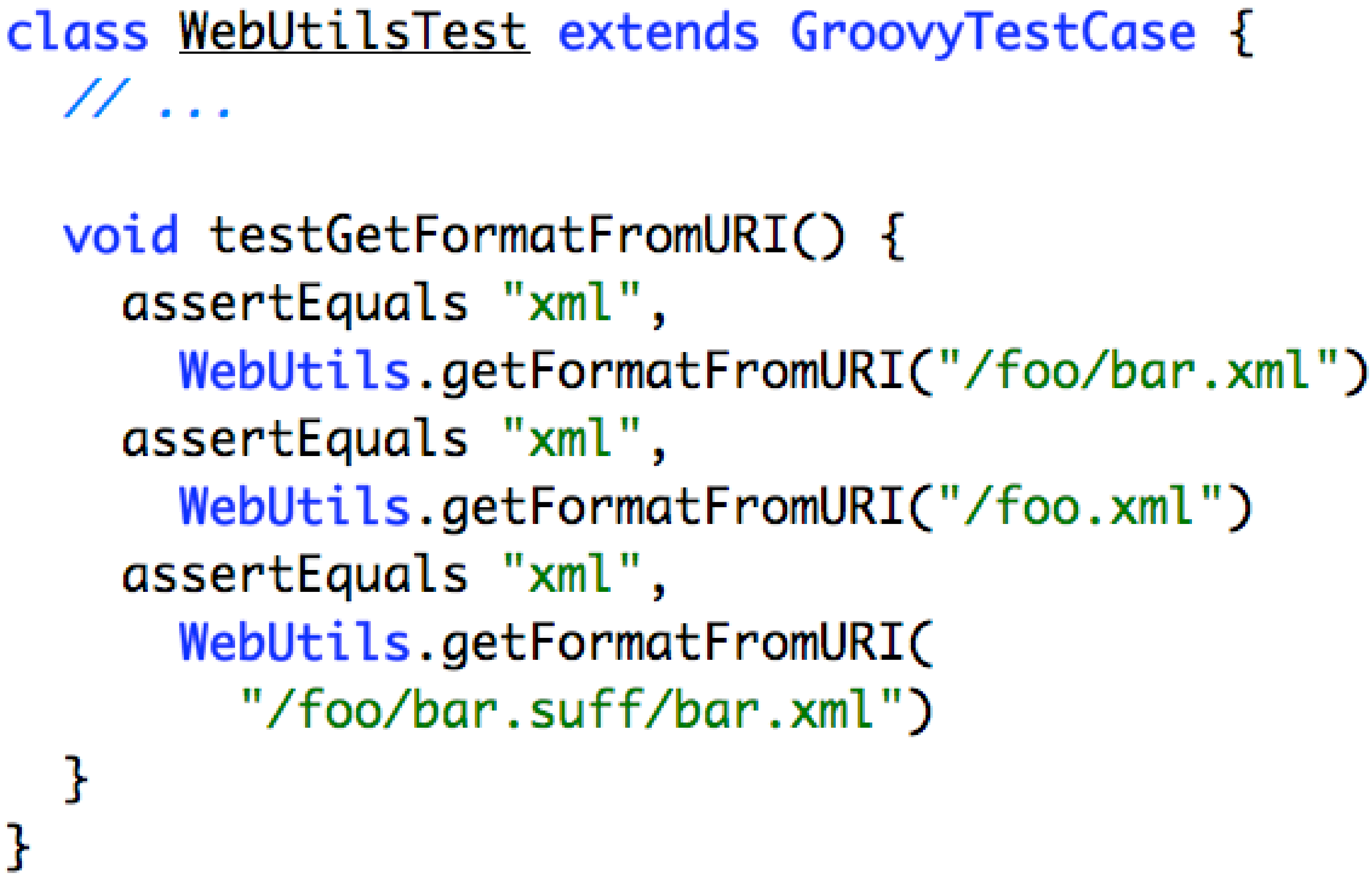
```
<junit fork="true" maxmemory="512m" forkmode="once">
  <!-- ... -->
    <fileset dir="${grails.test.build}">
      <include name="**/${test}Test.class" />
      <include name="**/${test}Tests.class" />
    </fileset>
  <!-- ... -->
</junit>
```

Going Green



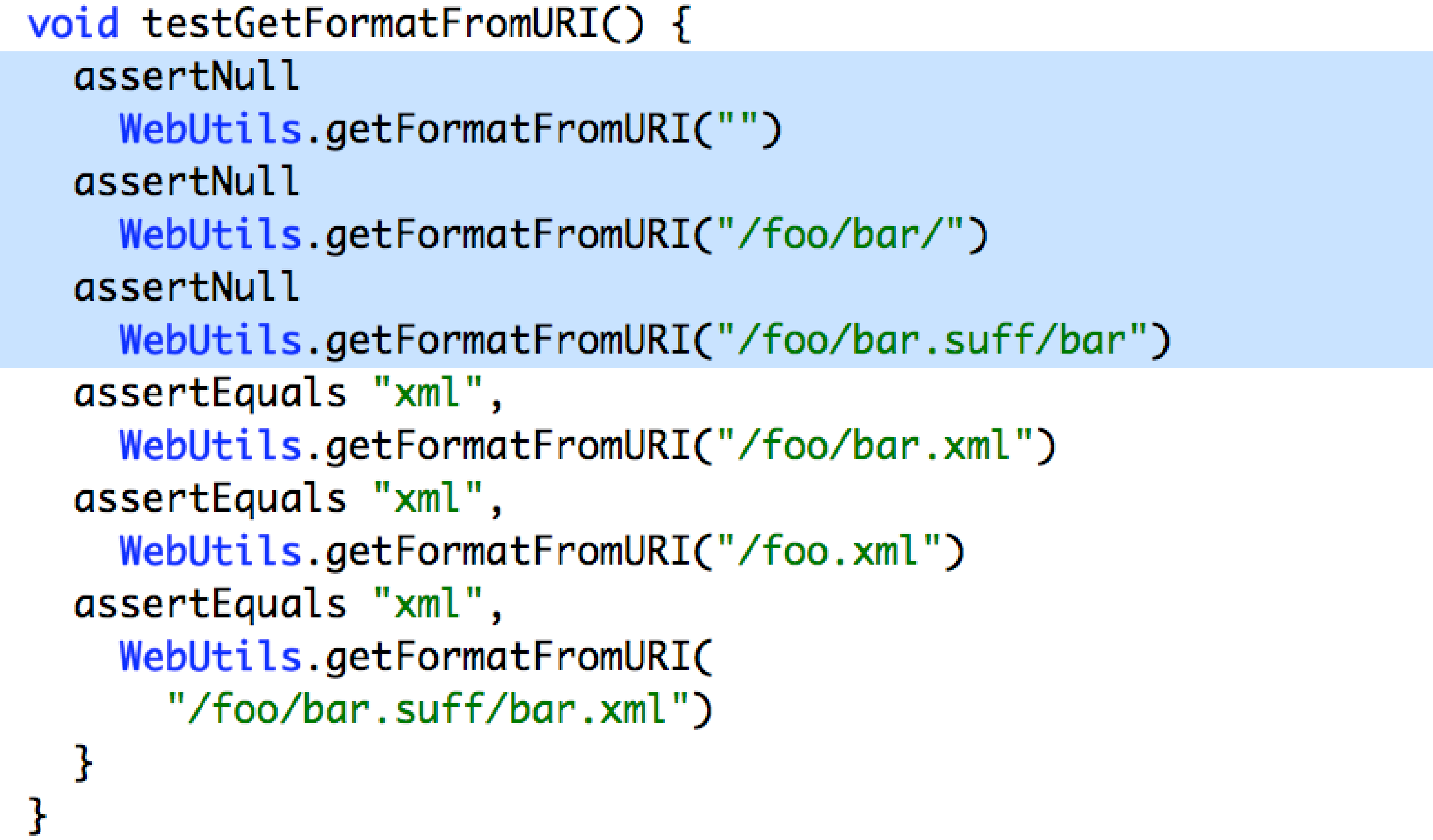
```
      /**
       * Obtains the format from the URI. The format is the string following the
       *
       * @param uri The URI
       * @return The format or null if none
       */
      public static String getFormatFromURI(String uri) {
13        if(uri.endsWith("/")) {
 1            return null;
          }
12        int idx = uri.lastIndexOf('/');
12        if(idx > -1) {
12            String lastToken = uri.substring(idx+1, uri.length());
12            idx = lastToken.lastIndexOf('.');
12            if(idx > -1) {
 3                return lastToken.substring(idx+1, lastToken.length());
              }
          }
 9        return null;
      }
```

Do We *Deserve* Full Credit?



```
class WebUtilsTest extends GroovyTestCase {
  // ...

  void testGetFormatFromURI() {
    assertEquals "xml",
      WebUtils.getFormatFromURI("/foo/bar.xml")
    assertEquals "xml",
      WebUtils.getFormatFromURI("/foo.xml")
    assertEquals "xml",
      WebUtils.getFormatFromURI(
        "/foo/bar.suff/bar.xml")
  }
}
```

Edge Cases Matter

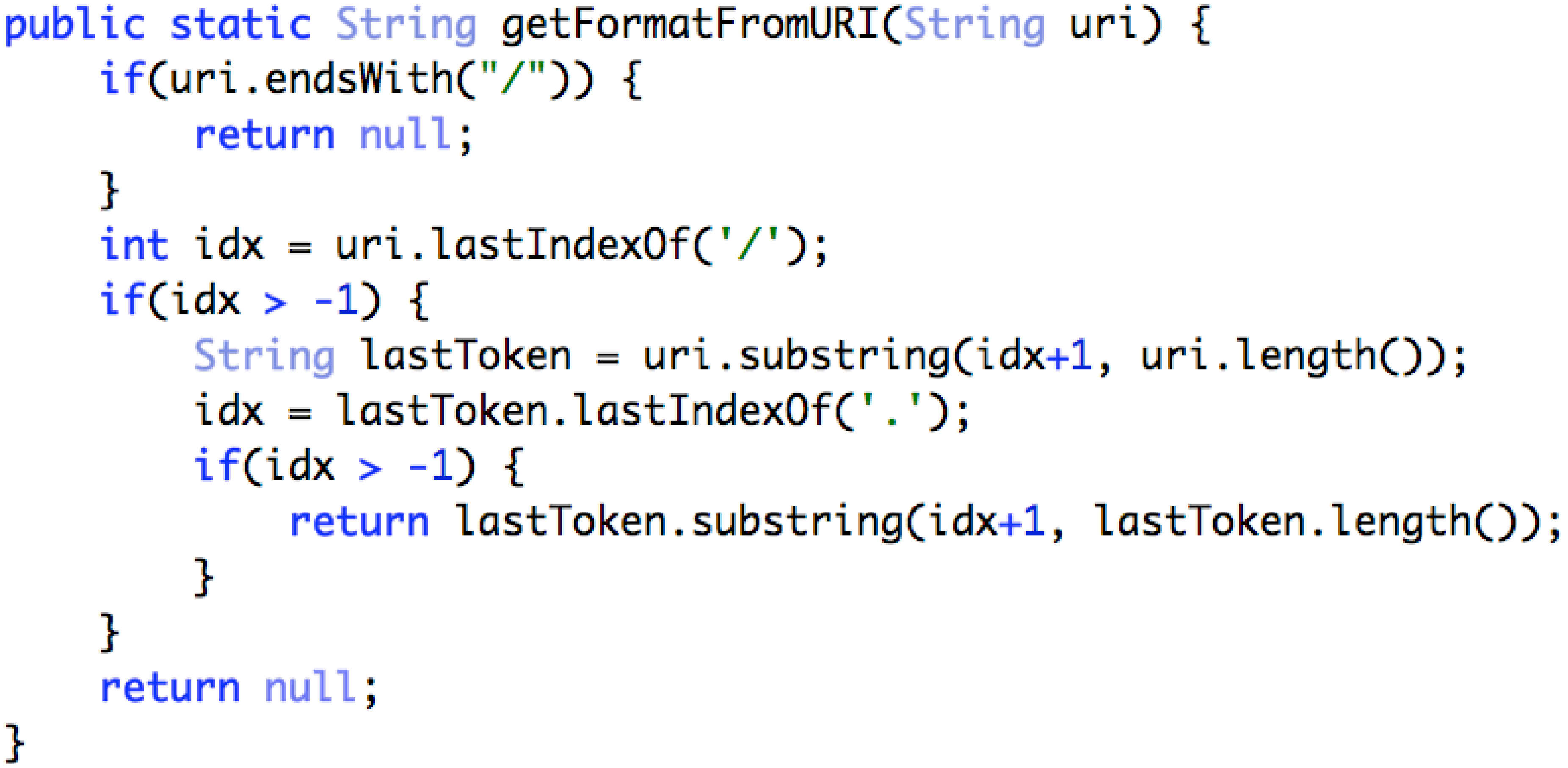


```
void testGetFormatFromURI() {
  assertNull
    WebUtils.getFormatFromURI("")
  assertNull
    WebUtils.getFormatFromURI("/foo/bar/")
  assertNull
    WebUtils.getFormatFromURI("/foo/bar.suff/bar")
  assertEquals "xml",
    WebUtils.getFormatFromURI("/foo/bar.xml")
  assertEquals "xml",
    WebUtils.getFormatFromURI("/foo.xml")
  assertEquals "xml",
    WebUtils.getFormatFromURI(
      "/foo/bar.suff/bar.xml")
  }
}
```

Is 100% Enough?

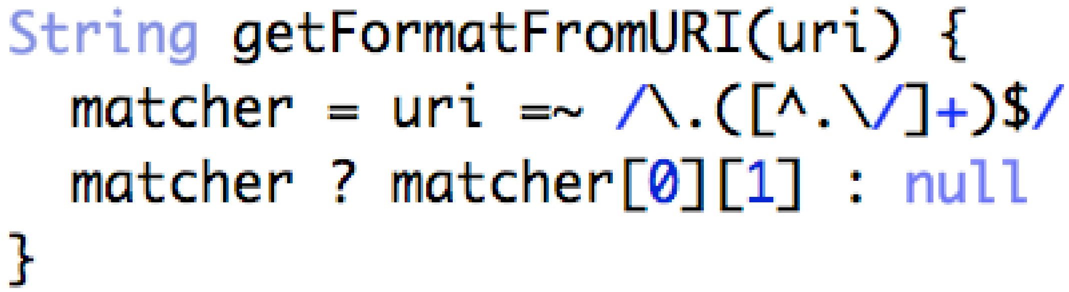
relevance

Still Not Exactly Elegant



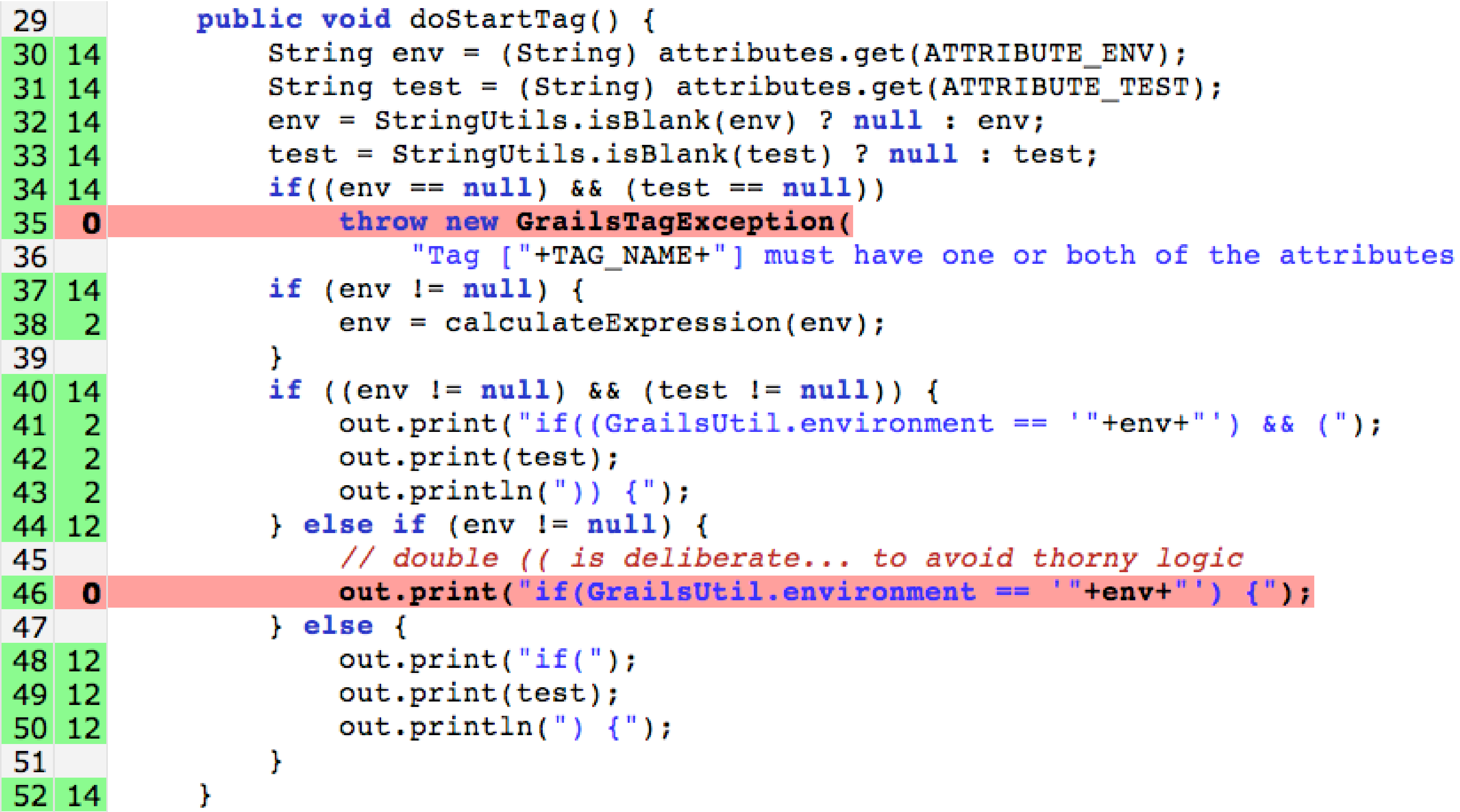
```java
public static String getFormatFromURI(String uri) {
    if(uri.endsWith("/")) {
        return null;
    }
    int idx = uri.lastIndexOf('/');
    if(idx > -1) {
        String lastToken = uri.substring(idx+1, uri.length());
        idx = lastToken.lastIndexOf('.');
        if(idx > -1) {
            return lastToken.substring(idx+1, lastToken.length());
        }
    }
    return null;
}
```

If Only It Were Groovy



```
String getFormatFromURI(uri) {
  matcher = uri =~ /^\.([^.\/]+)$/
  matcher ? matcher[0][1] : null
}
```

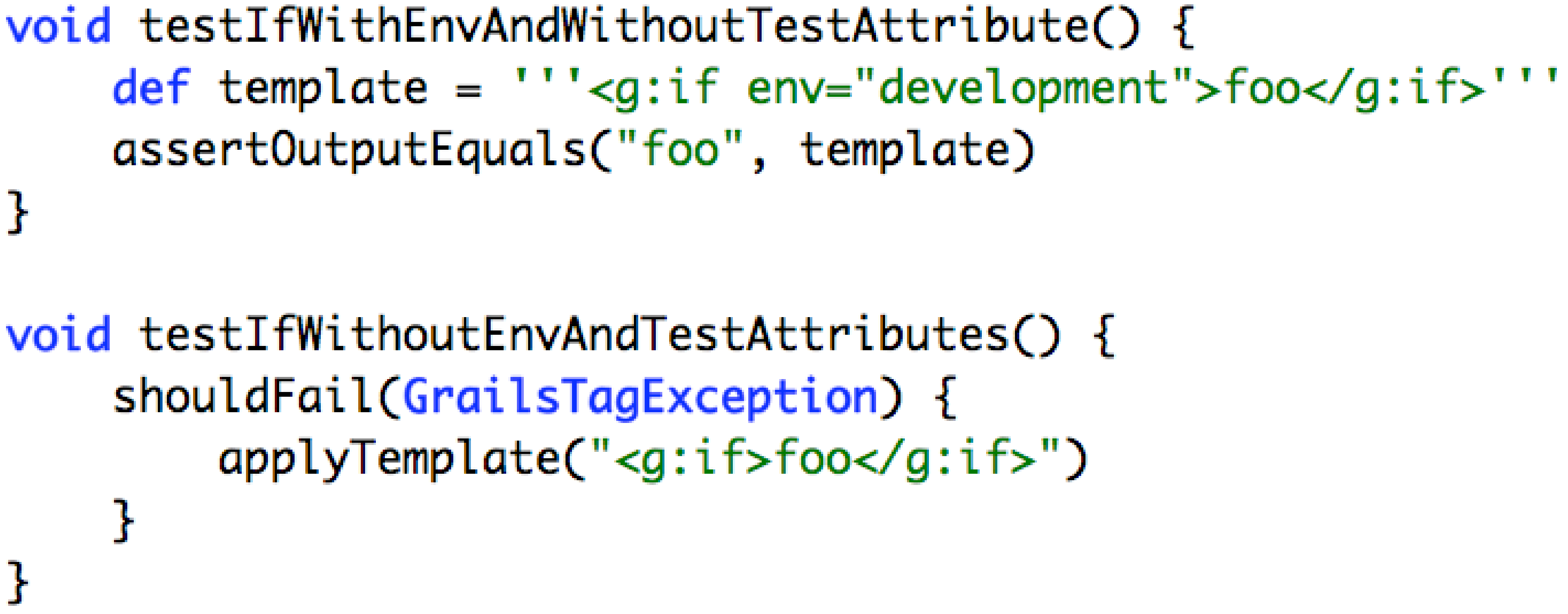
Trolling for “Thorny Logic”



```
        public void doStartTag() {
14          String env = (String) attributes.get(ATTRIBUTE_ENV);
14          String test = (String) attributes.get(ATTRIBUTE_TEST);
14          env = StringUtils.isBlank(env) ? null : env;
14          test = StringUtils.isBlank(test) ? null : test;
14          if((env == null) && (test == null))
 0              throw new GrailsTagException(
                    "Tag ["+TAG_NAME+"] must have one or both of the attributes
14          if (env != null) {
 2              env = calculateExpression(env);
            }
14          if ((env != null) && (test != null)) {
 2              out.print("if((GrailsUtil.environment == '"+env+"') && (");
 2              out.print(test);
 2              out.println(")) {");
12          } else if (env != null) {
                // double (( is deliberate... to avoid thorny logic
 0              out.print("if(GrailsUtil.environment == '"+env+"') {");
            } else {
12              out.print("if(");
12              out.print(test);
12              out.println(") {");
            }
14      }
```

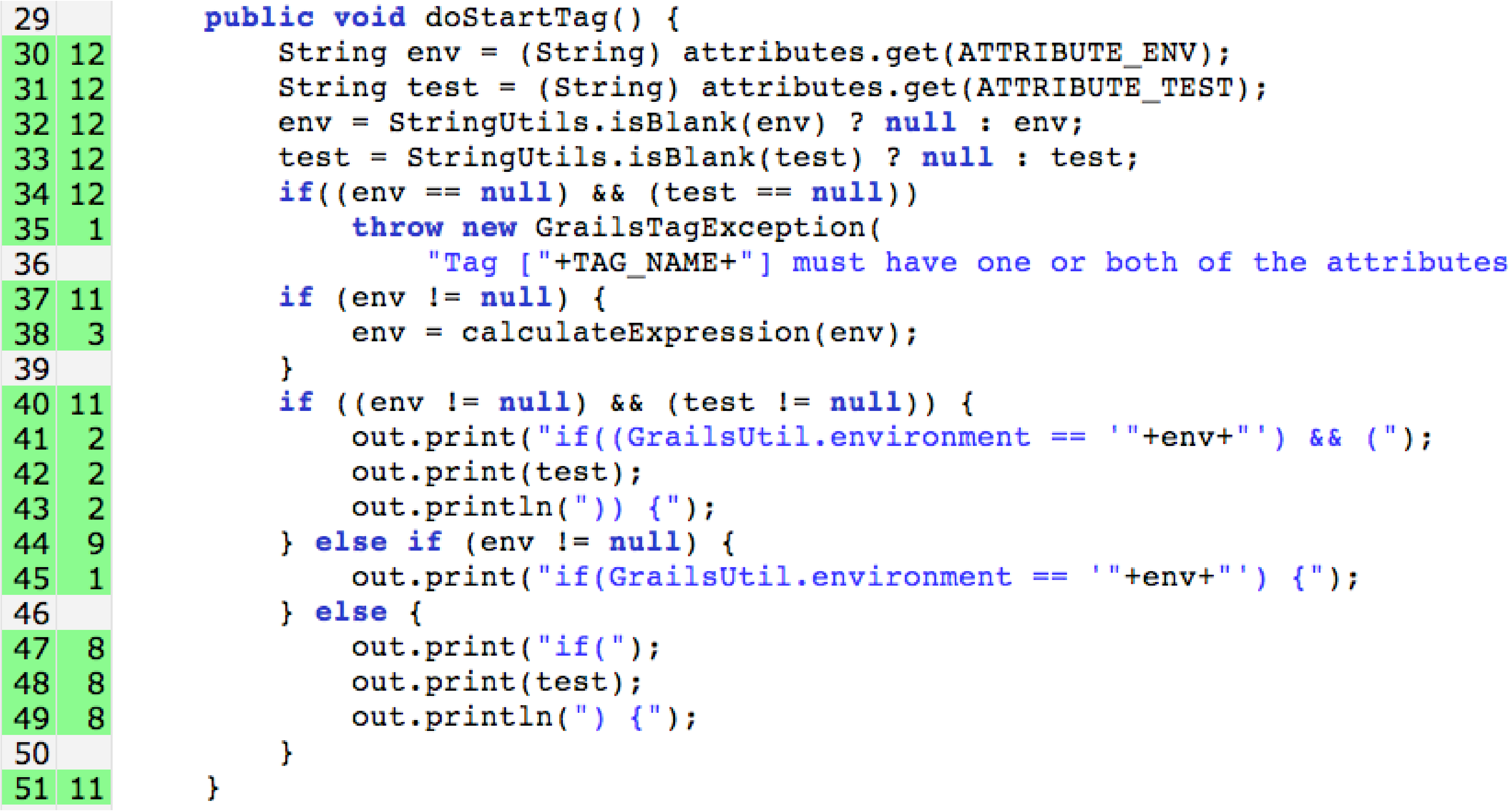
Red, Green, Refactor

First, the Tests



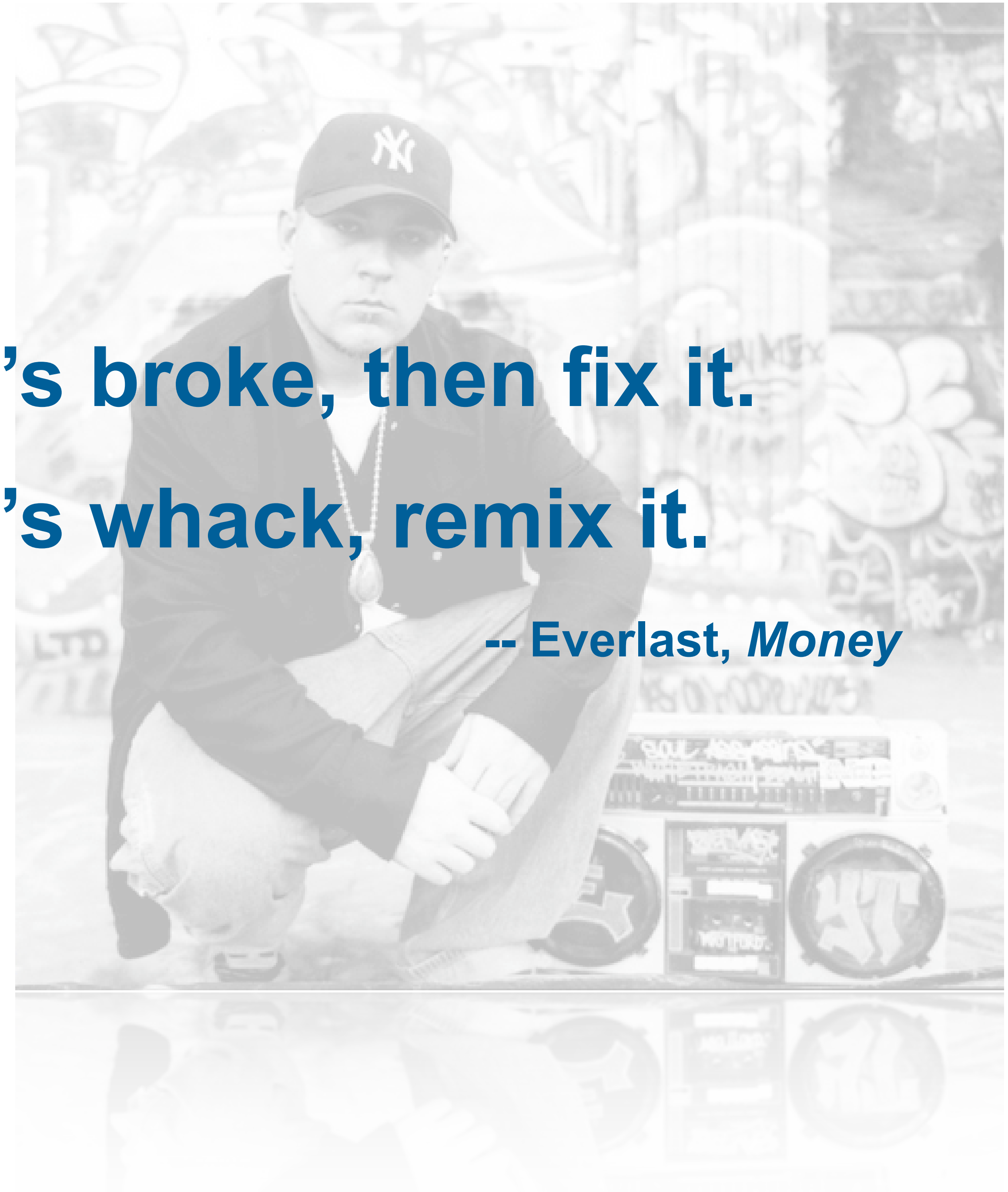
```
void testIfWithEnvAndWithoutTestAttribute() {
    def template = '''<g:if env="development">foo</g:if>'''
    assertOutputEquals("foo", template)
}

void testIfWithoutEnvAndTestAttributes() {
    shouldFail(GrailsTagException) {
        applyTemplate("<g:if>foo</g:if>")
    }
}
```

Greenlit for Refactoring



```
29          public void doStartTag() {
30  12          String env = (String) attributes.get(ATTRIBUTE_ENV);
31  12          String test = (String) attributes.get(ATTRIBUTE_TEST);
32  12          env = StringUtils.isBlank(env) ? null : env;
33  12          test = StringUtils.isBlank(test) ? null : test;
34  12          if((env == null) && (test == null))
35   1              throw new GrailsTagException(
36                      "Tag ["+TAG_NAME+"] must have one or both of the attributes
37  11          if (env != null) {
38   3              env = calculateExpression(env);
39              }
40  11          if ((env != null) && (test != null)) {
41   2              out.print("if((GrailsUtil.environment == '"+env+"') && (");
42   2              out.print(test);
43   2              out.println(")) {");
44   9          } else if (env != null) {
45   1              out.print("if(GrailsUtil.environment == '"+env+"') {");
46              } else {
47   8              out.print("if(");
48   8              out.print(test);
49   8              out.println(") {");
50              }
51  11      }
```

relevance

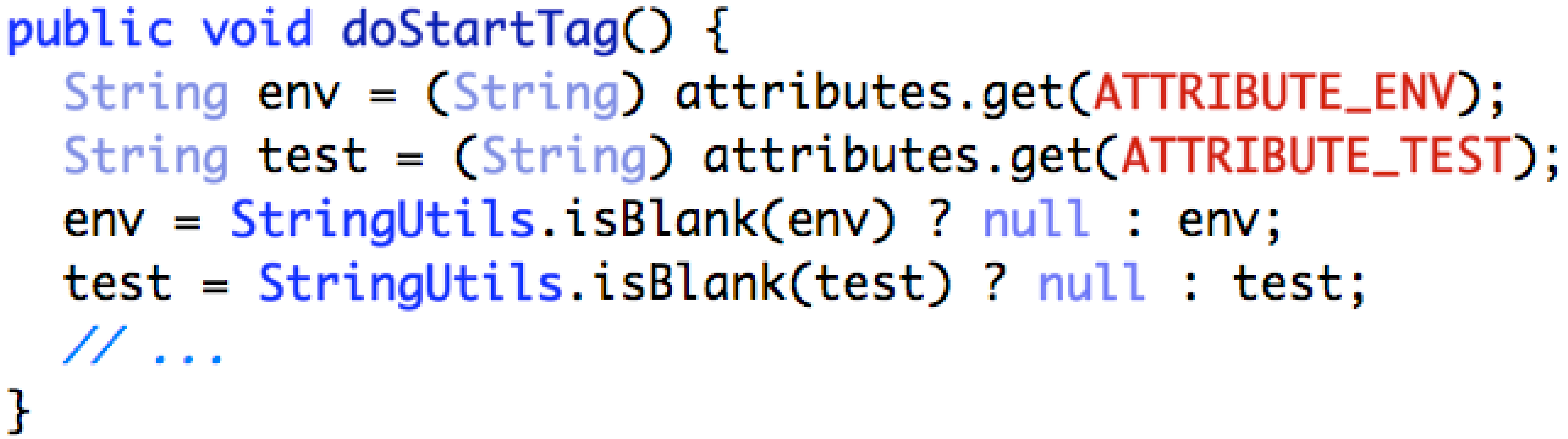


If it's broke, then fix it.

If it's whack, remix it.

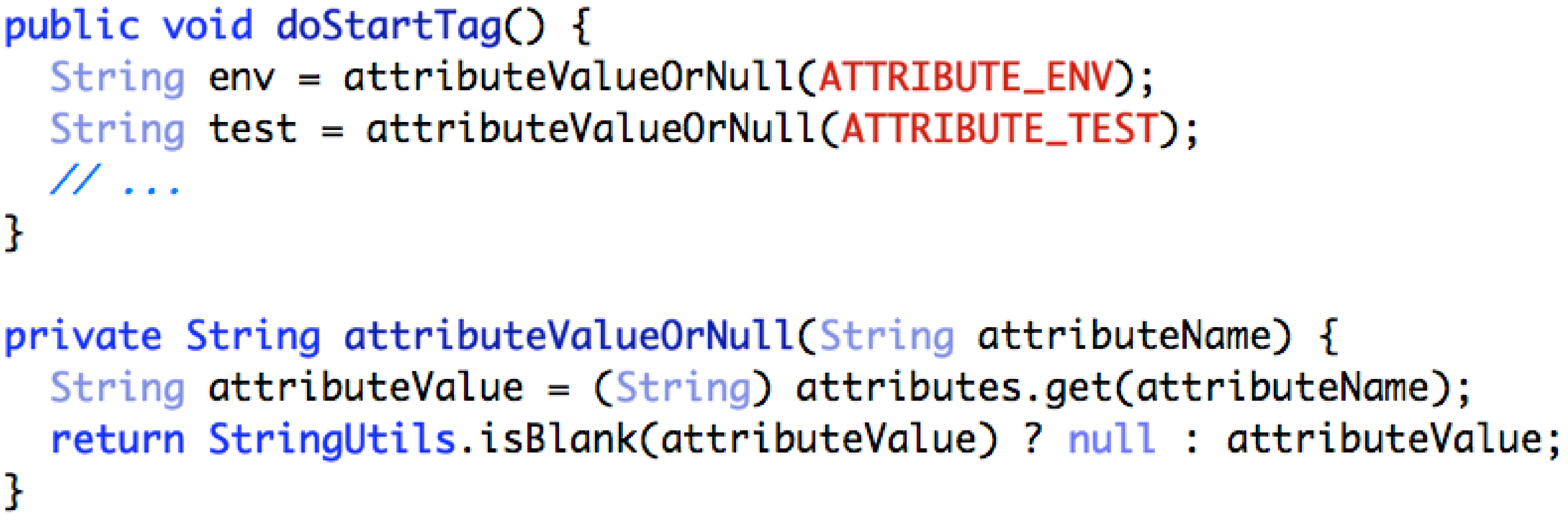
-- Everlast, *Money*

Problem: Too Much Repetition



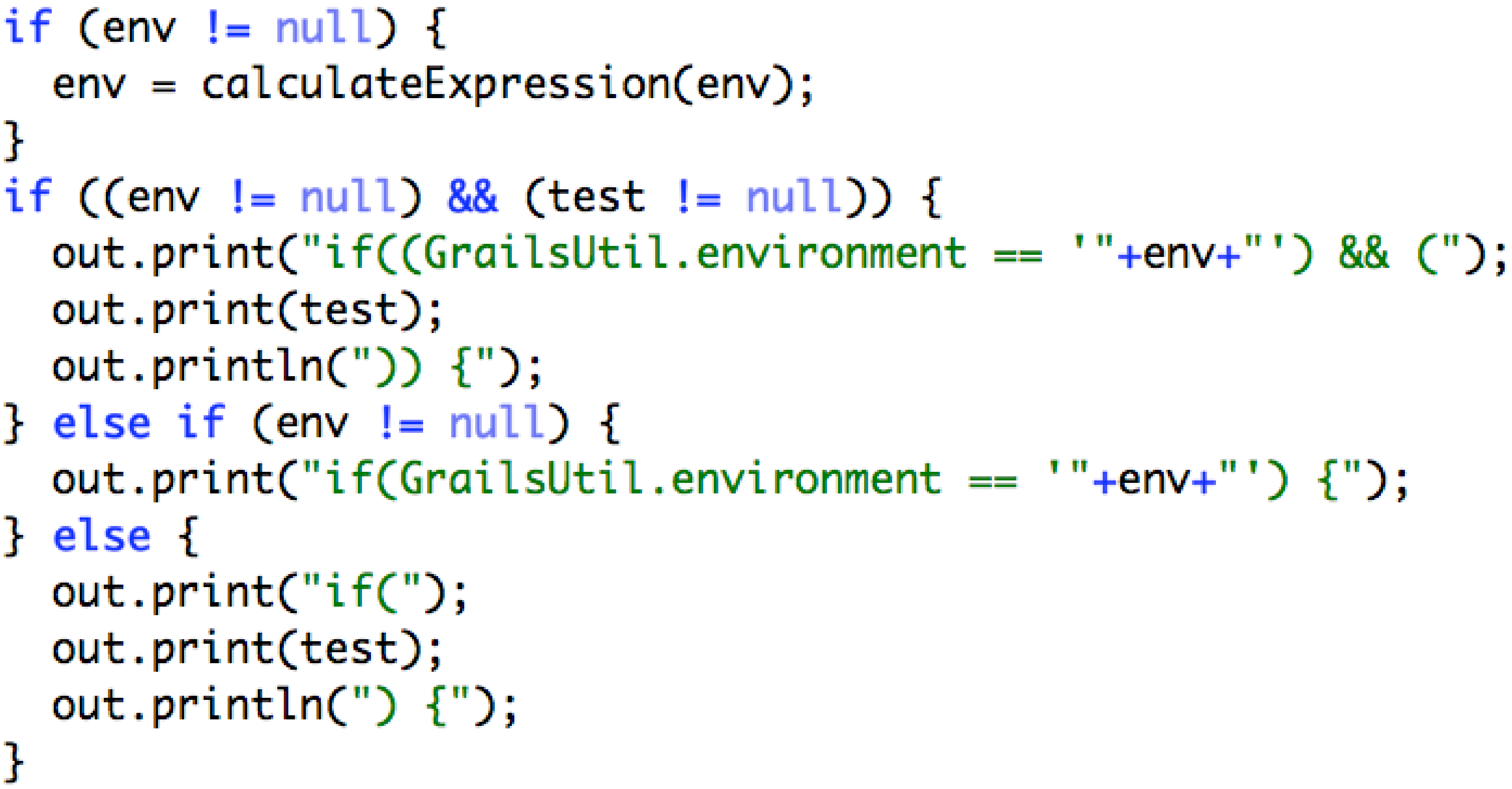
```java
public void doStartTag() {
  String env = (String) attributes.get(ATTRIBUTE_ENV);
  String test = (String) attributes.get(ATTRIBUTE_TEST);
  env = StringUtils.isBlank(env) ? null : env;
  test = StringUtils.isBlank(test) ? null : test;
  // ...
}
```

Solution: DRY It Up



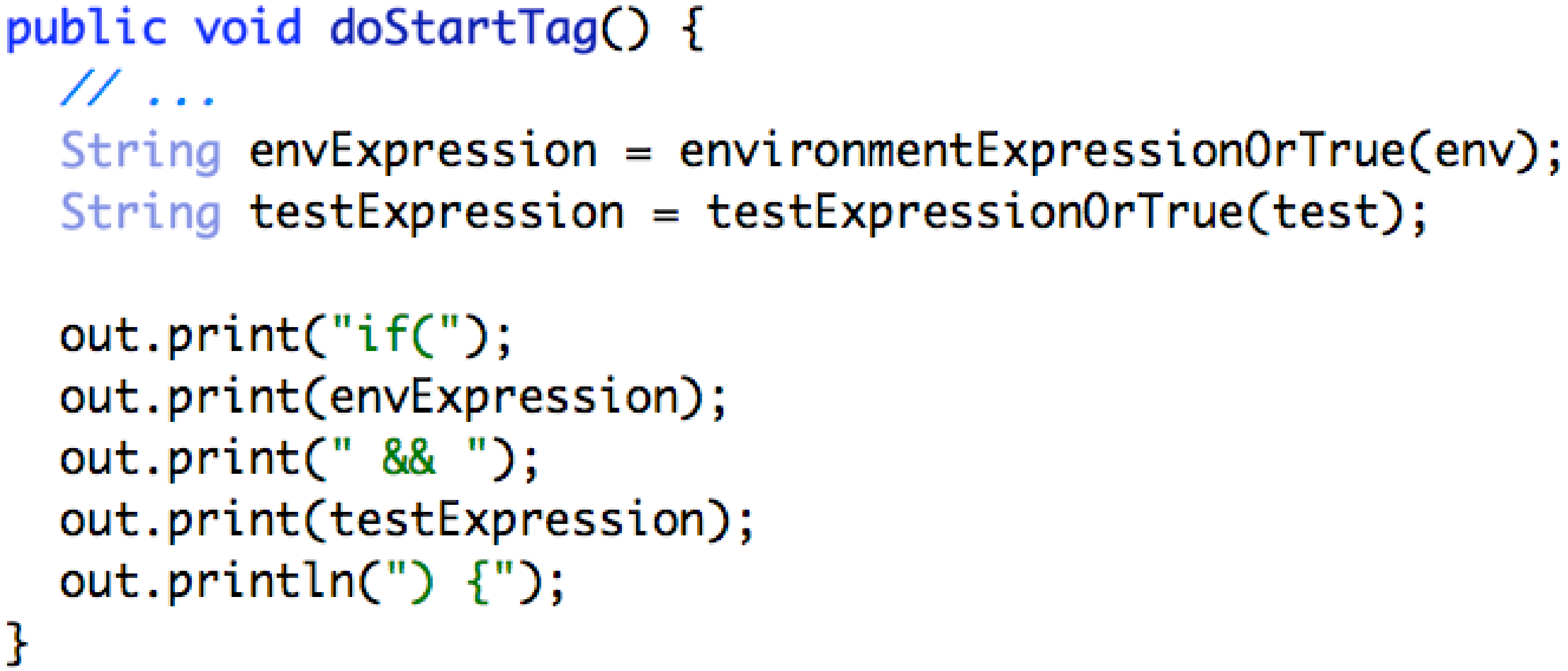
```
public void doStartTag() {
  String env = attributeValueOrNull(ATTRIBUTE_ENV);
  String test = attributeValueOrNull(ATTRIBUTE_TEST);
  // ...
}

private String attributeValueOrNull(String attributeName) {
  String attributeValue = (String) attributes.get(attributeName);
  return StringUtils.isBlank(attributeValue) ? null : attributeValue;
}
```

Problem: Too Many Conditions



```
if (env != null) {
  env = calculateExpression(env);
}
if ((env != null) && (test != null)) {
  out.print("if((GrailsUtil.environment == '"+env+"') && (");
  out.print(test);
  out.println(")) {");
} else if (env != null) {
  out.print("if(GrailsUtil.environment == '"+env+"') {");
} else {
  out.print("if(");
  out.print(test);
  out.println(") {");
}
```

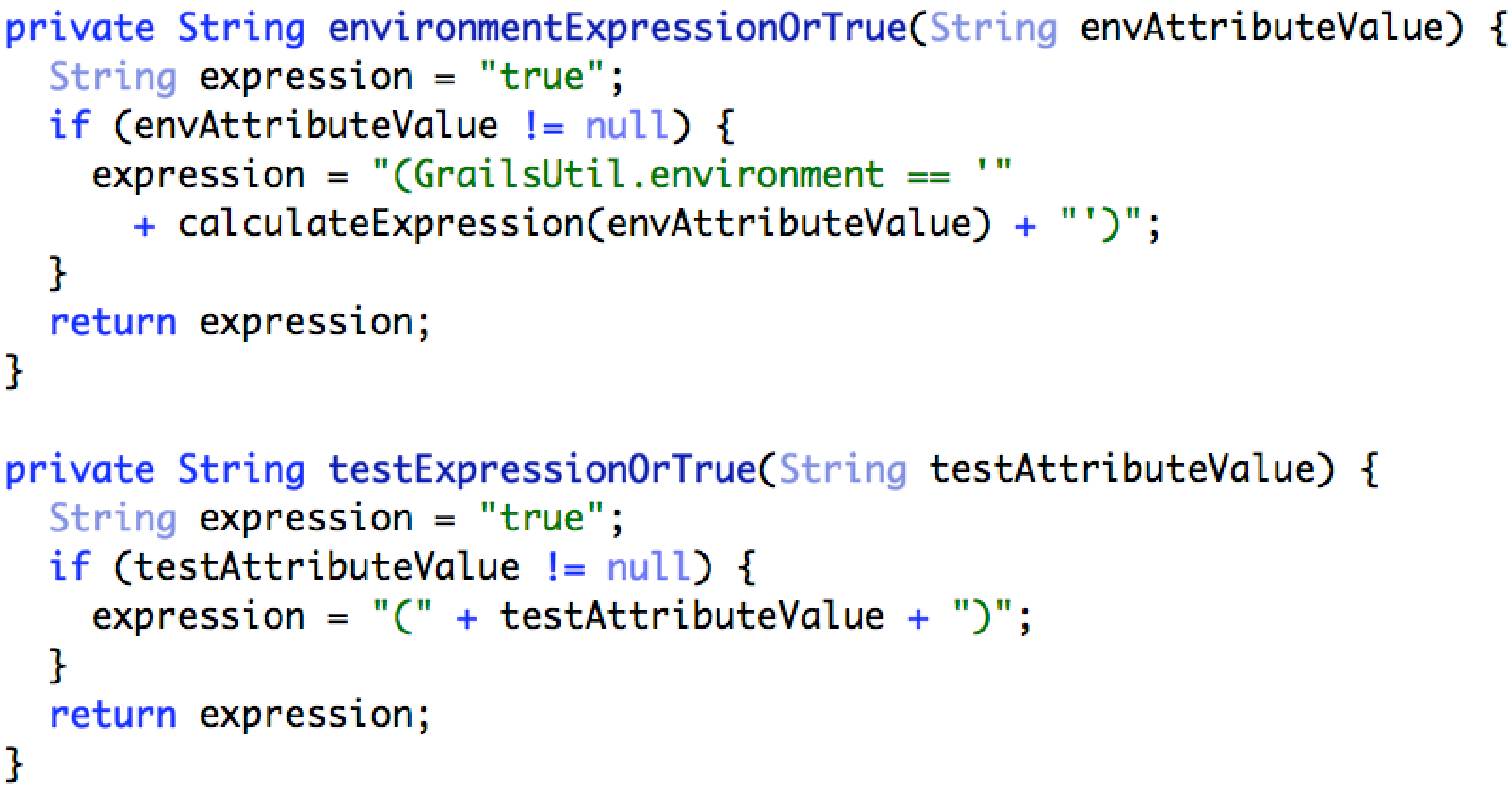
Solution: Defaults + Extraction



```
public void doStartTag() {
  // ...
  String envExpression = environmentExpressionOrTrue(env);
  String testExpression = testExpressionOrTrue(test);

  out.print("if(");
  out.print(envExpression);
  out.print(" && ");
  out.print(testExpression);
  out.println(") {");
}
```

Solution: Defaults + Extraction

(Continued)



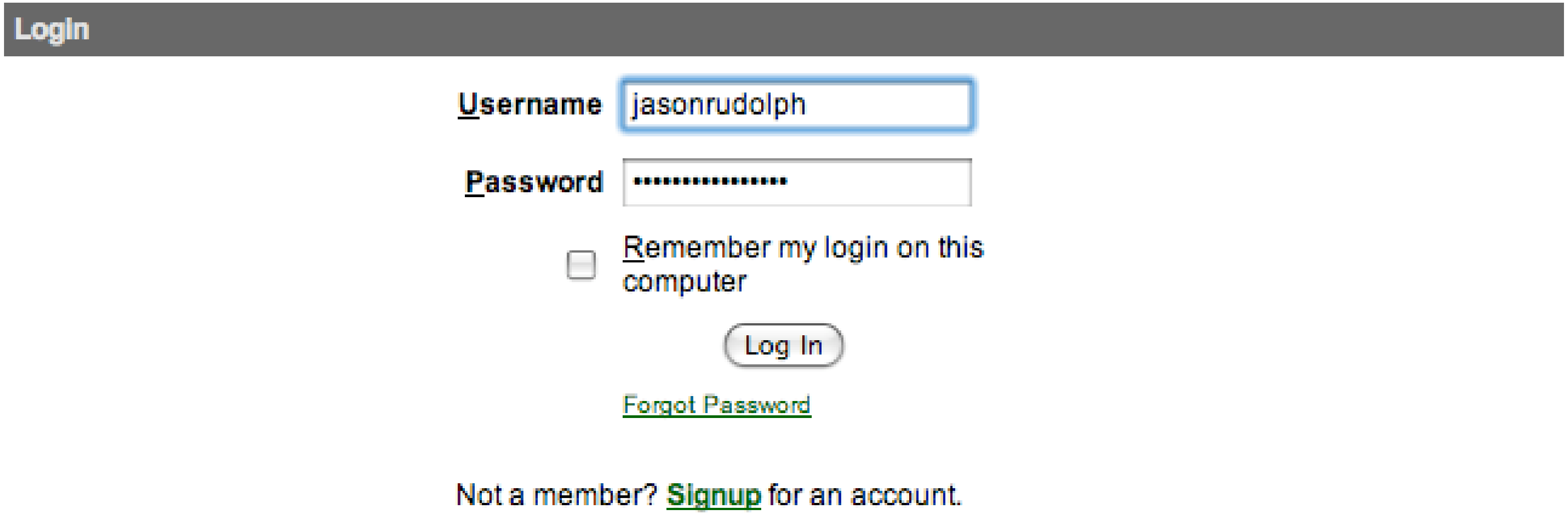
```
private String environmentExpressionOrTrue(String envAttributeValue) {
  String expression = "true";
  if (envAttributeValue != null) {
    expression = "(GrailsUtil.environment == '"
      + calculateExpression(envAttributeValue) + "')";
  }
  return expression;
}

private String testExpressionOrTrue(String testAttributeValue) {
  String expression = "true";
  if (testAttributeValue != null) {
    expression = "(" + testAttributeValue + ")";
  }
  return expression;
}
```

Test, Patch, Repeat

```
# Verify all tests pass
grails> ant test-with-coverage
...
BUILD SUCCESSFUL

# Verify test coverage still at 100%
grails> open cruise/coverage/report/index.html

# Create the patch
grails> svn diff > refactotum_2gx.diff
```

Login to JIRA

http://jira.codehaus.org



Login

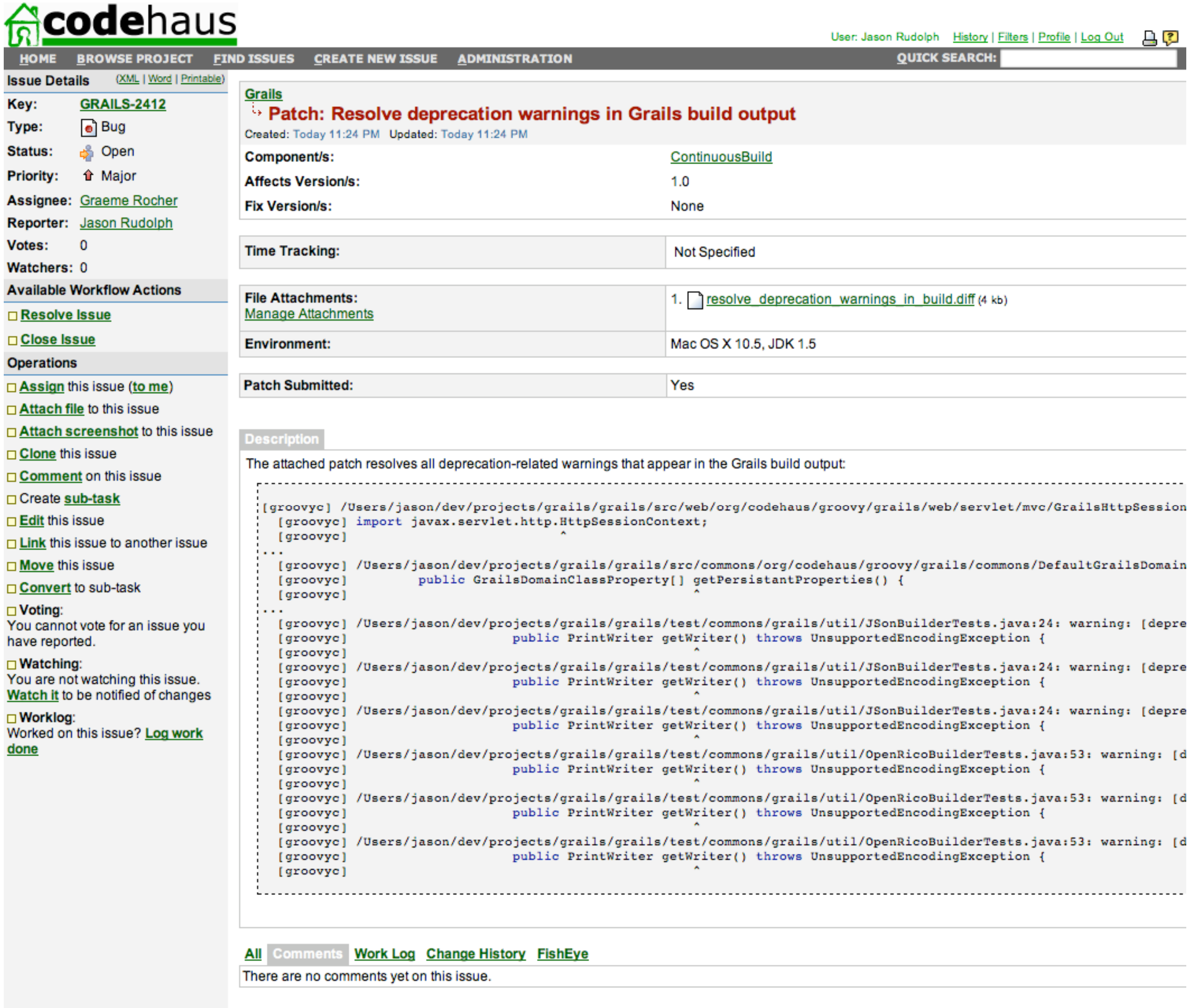
Username jasonrudolph

Password ••••••••••••••••

□ Remember my login on this computer

Not a member? Signup for an account.

Submit Patch



User: Jason Rudolph   History | Filters | Profile | Log Out

Issue Details (XML | Word | Printable)

Key:	GRAILS-2412
Type:	Bug
Status:	Open
Priority:	Major
Assignee:	Graeme Rocher
Reporter:	Jason Rudolph
Votes:	0
Watchers:	0

Available Workflow Actions

- □ **Resolve Issue**
- □ **Close Issue**

Operations

- □ **Assign** this issue (**to me**)
- □ **Attach file** to this issue
- □ **Attach screenshot** to this issue
- □ **Clone** this issue
- □ **Comment** on this issue
- □ Create **sub-task**
- □ **Edit** this issue
- □ **Link** this issue to another issue
- □ **Move** this issue
- □ **Convert** to sub-task
- □ **Voting**:
  You cannot vote for an issue you have reported.
- □ **Watching**:
  You are not watching this issue.
  Watch it to be notified of changes
- □ **Worklog**:
  Worked on this issue? **Log work done**

Grails

Patch: Resolve deprecation warnings in Grails build output

Created: Today 11:24 PM   Updated: Today 11:24 PM

Component/s:	ContinuousBuild
Affects Version/s:	1.0
Fix Version/s:	None
Time Tracking:	Not Specified
File Attachments: Manage Attachments	1. resolve_deprecation_warnings_in_build.diff (4 kb)
Environment:	Mac OS X 10.5, JDK 1.5
Patch Submitted:	Yes

Description

The attached patch resolves all deprecation-related warnings that appear in the Grails build output:

```
[groovyc] /Users/jason/dev/projects/grails/grails/src/web/org/codehaus/groovy/grails/web/servlet/mvc/GrailsHttpSession
  [groovyc] import javax.servlet.http.HttpSessionContext;
  [groovyc]                            ^
...
  [groovyc] /Users/jason/dev/projects/grails/grails/src/commons/org/codehaus/groovy/grails/commons/DefaultGrailsDomain
  [groovyc]         public GrailsDomainClassProperty[] getPersistantProperties() {
  [groovyc]                                             ^
...
  [groovyc] /Users/jason/dev/projects/grails/grails/test/commons/grails/util/JSonBuilderTests.java:24: warning: [depre
  [groovyc]                     public PrintWriter getWriter() throws UnsupportedEncodingException {
  [groovyc]                                             ^
  [groovyc] /Users/jason/dev/projects/grails/grails/test/commons/grails/util/JSonBuilderTests.java:24: warning: [depre
  [groovyc]                     public PrintWriter getWriter() throws UnsupportedEncodingException {
  [groovyc]                                             ^
  [groovyc] /Users/jason/dev/projects/grails/grails/test/commons/grails/util/JSonBuilderTests.java:24: warning: [depre
  [groovyc]                     public PrintWriter getWriter() throws UnsupportedEncodingException {
  [groovyc]                                             ^
  [groovyc] /Users/jason/dev/projects/grails/grails/test/commons/grails/util/OpenRicoBuilderTests.java:53: warning: [d
  [groovyc]                     public PrintWriter getWriter() throws UnsupportedEncodingException {
  [groovyc]                                             ^
  [groovyc] /Users/jason/dev/projects/grails/grails/test/commons/grails/util/OpenRicoBuilderTests.java:53: warning: [d
  [groovyc]                     public PrintWriter getWriter() throws UnsupportedEncodingException {
  [groovyc]                                             ^
  [groovyc] /Users/jason/dev/projects/grails/grails/test/commons/grails/util/OpenRicoBuilderTests.java:53: warning: [d
  [groovyc]                     public PrintWriter getWriter() throws UnsupportedEncodingException {
  [groovyc]                                             ^
```

There are no comments yet on this issue.

Wrapping Up

Contributions Come in Many Flavors

- **Removed build noise**
- **Added missing tests to build**
- **Provided better *unit* tests for WebUtils.java**
- **Increased test coverage for GroovyIfTag.java**
- **Refactored GroovyIfTag.java**
  - Increased DRYness
  - Reduced complexity

Resources

- **This Week in Refactoring**
  - blog.thinkrelevance.com/twir
- **Jay Fields on Test Quality and Coverage**
  - blog.jayfields.com/2008/02/tests-reflect-code-quality.html

relevance

Resources

(Continued)

- **Are You Covered?**
  - en.wikipedia.org/wiki/code_coverage
  - cobertura.sourceforge.net
  - build.canoo.com/grails/artifacts/coverage/index.html

Resources
(Continued)

- **Refactotum JIRA Tickets**
  - jira.codehaus.org/browse/GRAILS-2412
  - jira.codehaus.org/browse/GRAILS-2469
  - jira.codehaus.org/browse/GRAILS-2470
  - jira.codehaus.org/browse/GRAILS-2471

Let’s Contribute!

Workshop