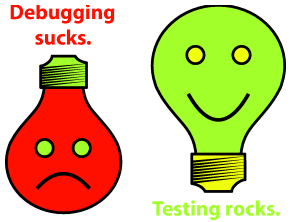


Testing on the Toilet

Naming Unit Tests Responsibly

January 30, 2007

For a class, try having a corresponding set of test methods, where each one describes a responsibility of the object, with the first word implicitly the name of the class under test. For example, in Java:

```
class HtmlLinkRewriterTest ... {
  void testAppendsAdditionalParameterToUrlsInHrefAttributes(){…}
  void testDoesNotRewriteImageOrJavascriptLinks(){…}
  void testThrowsExceptionIfHrefContainsSessionId(){…}
  void testEncodesParameterValue(){…}
}
```

This can be read as:

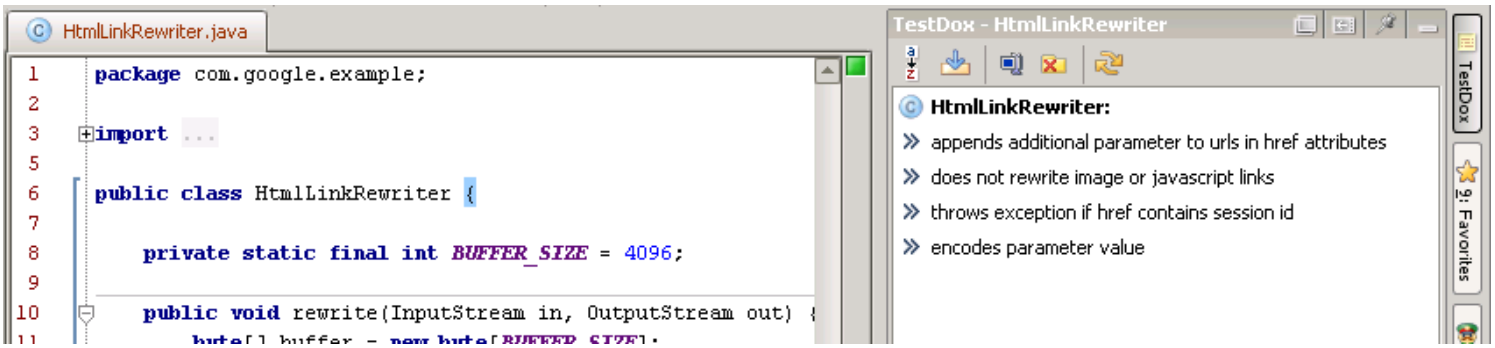
> HtmlLinkRewriter appends additional parameter to URLs in href attributes.
> HtmlLinkRewriter does not rewrite image or JavaScript links.
> HtmlLinkRewriter throws exception if href contains session ID.
> HtmlLinkRewriter encodes parameter value.

Benefits

The tests emphasizes the object's responsibilities (or features) rather than public methods and inputs/output. This makes it easier for future engineers who want to know what it does without having to delve into the code.

These naming conventions can help point out smells. For example, when it's hard to construct a sentence where the first word is the class under test, it suggests the test may be in the wrong place. And classes that are hard to describe in general often need to be broken down into smaller classes with clearer responsibilities.

Additionally, tools can be used to help understand code quicker:



(This example shows a class in IntelliJ with the TestDox plugin giving an overview of the test.)

More information, feedback, and discussion:

http://googletesting.blogspot.com