



HtmlUnit:

An Efficient Approach to Testing Web Applications

Marc Guillemot

Independent Consultant
mguillemot@yahoo.fr

Daniel Gredler

Sr. Software Developer
DHL Global Mail
daniel.gredler@gmail.com



Your Speakers



> **Marc Guillemot**

- Independent consultant
- Lead developer of HtmlUnit and WebTest
- Committer to Groovy



> **Daniel Gredler**

- Sr. Software Developer @ DHL Global Mail
- Committer to HtmlUnit and Apache Tapestry



Agenda

- **What is HtmlUnit?**
- **Simulation: Possibilities & Limitations**
- **AJAX**
- **Inside HtmlUnit**
- **Tools which use HtmlUnit**
- **Future Plans**
- **Q&A**



Requirements for Successful Test Automation

- **Awareness for quality assurance**
 - Unit tests
 - Continuous Integration
- **Test early**
- **Write testable HTML code**
- **Don't write fragile tests!**



What is HtmlUnit?

- **A project with a very bad name!**
- **100% Java-based headless browser**
- **Open Source (Apache 2 license)**
 - Hosted at SourceForge
 - 7 committers (3 very active)
 - Numerous contributors
 - Very mature
- **Useful for integration testing, screen scraping, etc.**



Sample Use

```
@Test
public void htmlUnitWebsiteFromGoogle() throws Exception {
    WebClient webClient = new WebClient(BrowserVersion.FIREFOX_3);

    HtmlPage startPage = webClient.getPage("http://www.google.com");
    assertEquals("Google", startPage.getTitleText());

    HtmlInput queryField = (HtmlInput) startPage.getElementsByName("q").get(0);
    queryField.setValueAttribute("HtmlUnit");

    HtmlElement button =
        startPage.getFirstByXPath("//input[@value = \"I'm Feeling Lucky\"]");
    HtmlPage p2 = button.click();
    assertEquals("HtmlUnit - Welcome to HtmlUnit", p2.getTitleText());
}
```



HtmlUnit Simulates “Real” Browsers

- **Focus on 4 browsers**
 - Firefox 2 & 3
 - Internet Explorer 6 & 7
- **Tries to mimic browser behavior**
 - HTTP requests
 - HTML parsing
 - CSS parsing
 - JavaScript execution



Simulation?

- **“Real end users don’t use HtmlUnit.”**
- **“Tests should run in a real browser to ensure that the application works.”**

(demo)



Simulation!?!?

- **Tests don't ensure that an application works correctly**
- **Tests find errors -- that's all**
- **HtmlUnit can help to find a lot of errors**



What kind of errors can be found?

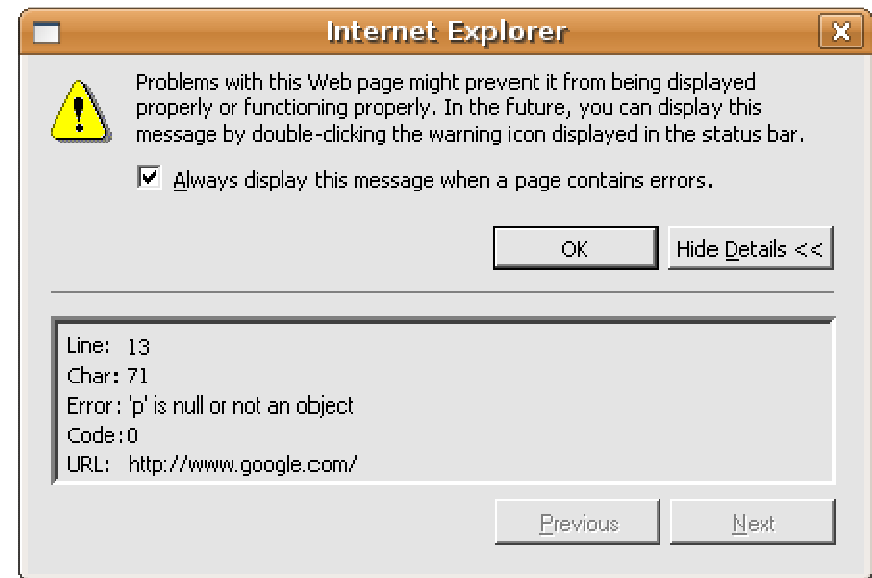
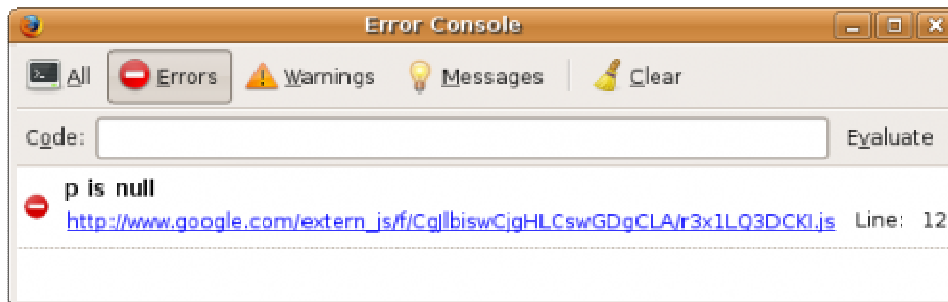
- **Like any test tool: functional errors**
- **JavaScript errors**
- **HTTP errors: 404, 500, ...**
- **“Incorrectness”**
 - incorrect HTML
 - malformed URLs
 - incorrect headers
 - ...



Example JavaScript Error

- **www.google.com (Sept. 2008)**

JavaScript error loading page <http://www.google.com/>: TypeError: Cannot call method "item" of null (http://www.google.com/extern_js/f/CgJlbiswCjgHLCswGDgCLA/FQ1b7bFTJak.js#12)



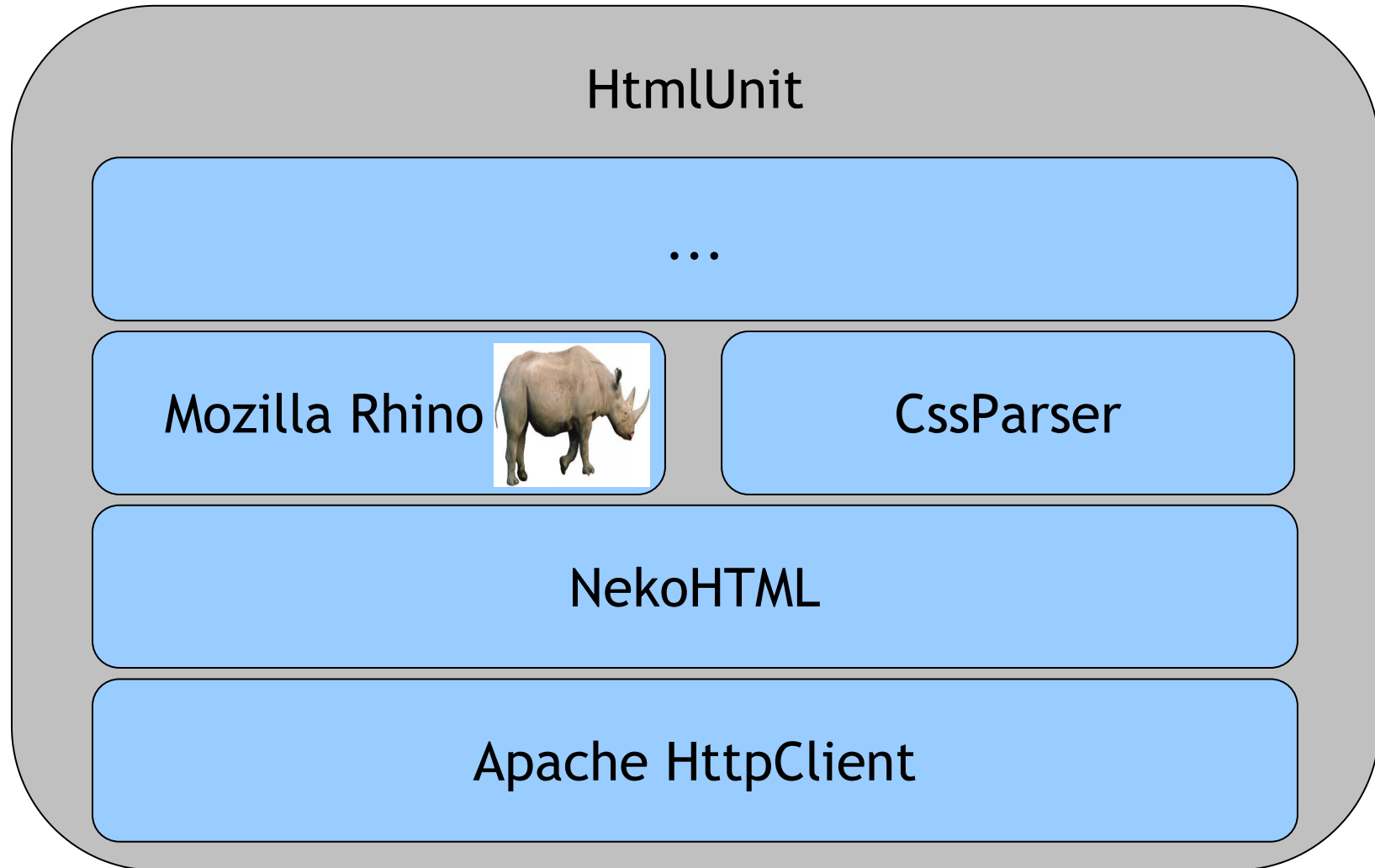


AJAX Timing

- **AJAX testing = need for synchronization**
- **HtmlUnit solutions:**
 - Polling is not the right way!
 - `NicelyResynchronizingAjaxController`
 - `WebClient.waitForBackgroundJavaScript(delay)` (coming in 2.5)



Architecture





Customization

- **Enable / Disable**
 - JavaScript
 - CSS
 - Popup Blocker

- **Throw / No Throw**
 - on Script Error
 - on HTTP Failure Status Codes

- **Use Insecure SSL**

- ...



Extension Points

- **Alert / Confirm / Prompt / Status Handlers**
- **JavaScript Preprocessors**
- **JavaScript Debugger Callbacks**
- **Custom Web Connections**
- **Incorrectness Listeners: HTML, CSS, etc.**
- **...**



Extension Point Example: Falsify Responses

```
final WebClient webClient = new WebClient();

webClient.setWebConnection(new FalsifyingWebConnection(webClient) {
    @Override
    public WebResponse getResponse(final WebRequestSettings wrs)
        throws IOException {

        if ("some.other.server".equals(wrs.getUrl().getHost())) {
            final String body = "An error occurred";
            final List<NameValuePair> headers = Collections.emptyList();
            final WebResponseData wrd = new WebResponseData(body.getBytes(),
                500, "Application error", headers);
            return new WebResponseImpl(wrd, wrs.getUrl(), wrs.getHttpMethod(), 1);
        }

        return super.getResponse(wrs);
    }
});
```

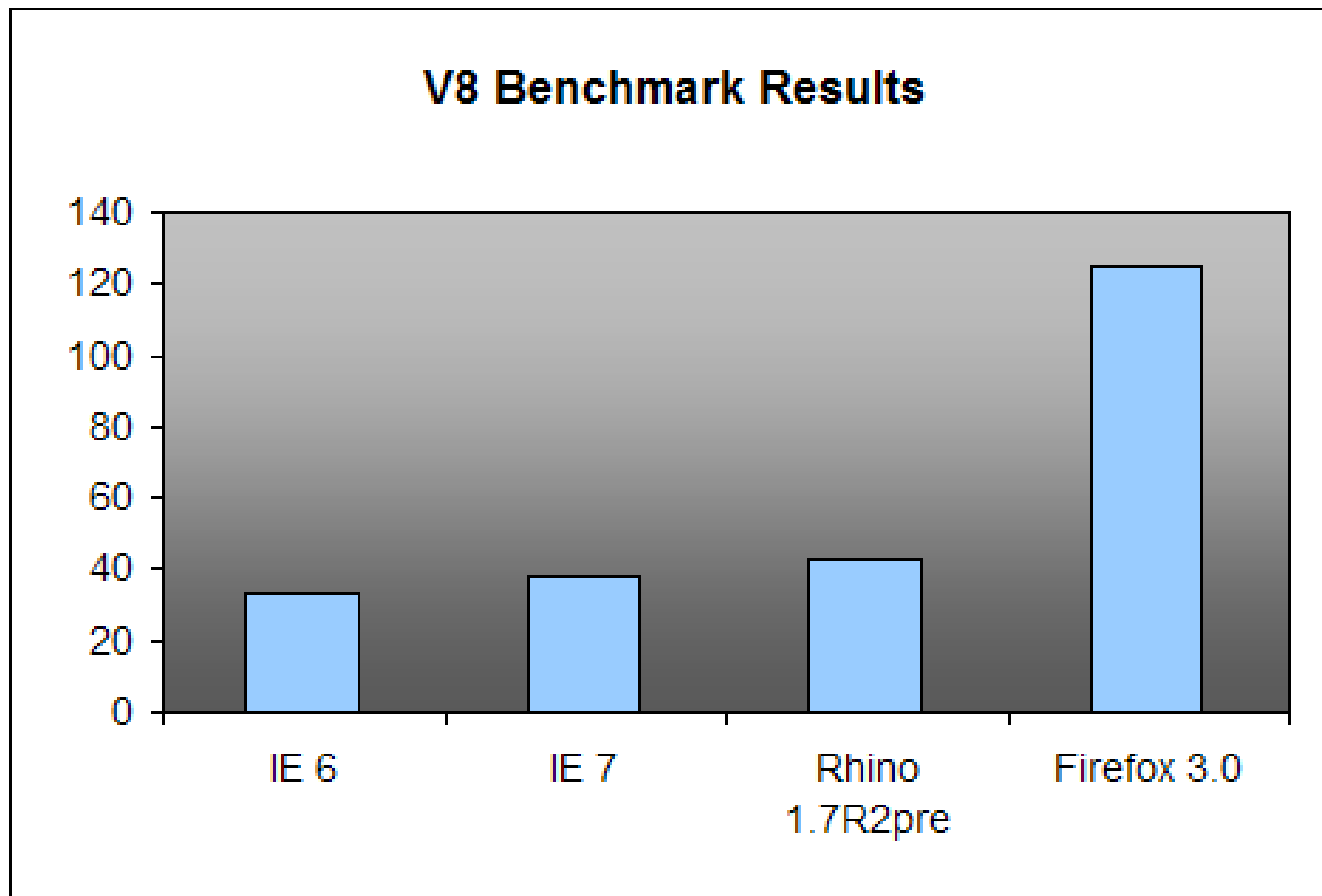



Performance

- **Reduce network traffic**
- **No rendering**
- **No browser startup pause**
- **Data point: Celerity vs. Watir**
 - **Simple local file: test time reduced by 99%**
 - **Google image search: test time reduced by 69%**
 - **Digg front page scraping: test time reduced by 74%**
 - **Local file with DOM access: test time reduced by 97%**



Performance: Rhino vs. IE





Other Advantages

● Platform Independence

- “I’m a PC” developer...
- vs. “I’m a Mac” developer...
- vs. build server...
- vs. continuous integration server

● Scalability

- Standard JVM setup...
- vs. grid component...
- vs. cloud infrastructure



Limitations

- **Simulation not 100% correct:**
 - for incorrect HTML code
 - for JavaScript execution
 - on the HTTP layer
- **RIAs**
 - No support for Flash or Silverlight
 - Applets supported only as proof of concept



Ensuring Accuracy

- **“Simple” unit tests**
- **Unit tests using WebDriver**
- **AJAX library integration tests**
- **JavaScript execution flow comparisons**



“Simple” Unit Tests

```
@RunWith(BrowserRunner.class)
public class WindowTest extends WebTestCase {
    ...
    @Test
    @Alerts(IE = { "undefined", "undefined" },
            FF2 = { "[Node]", "[Element]" },
            FF3 = { "[object Node]", "[object Element]" })
    public void windowProperties() throws Exception {
        final String html = "<html><head><title>foo</title><script>"
            + "  function test() {"
            + "    alert(window.Node);"
            + "    alert(window.Element);"
            + "  }\n"
            + "</script></head><body onload='test()'"
            + "</body></html>";

        loadPageWithAlerts(html);
    }
    ...
}
```



Library Integration Tests

- **CurvyCorners**
- **Dojo**
- **ExtJS**
- **GWT**
- **JQuery**
- **MochiKit**
- **Prototype**
- **Sarissa**
- **YUI**
- **...**



Tools which use HtmlUnit

WebDriver

Canoo WebTest

Wepawet

JWebUnit

JSFUnit

Grails Functional
Testing Plugin

PushToTest
TestMaker

Celerity

TestPlan

Schnell

Perl HtmlUnit



Example: WebTest

```
import com.canoo.webtest.WebtestCase

class SimpleTest extends WebtestCase {
    void testWebtestOnGoogle() {
        webtest("check that WebTest is Google's top 'WebTest' result") {
            invoke "http://www.google.com/ncr", description: "Go to Google (in English)"
            verifyTitle "Google"
            setInputField name: "q", value: "WebTest"
            clickButton "I'm Feeling Lucky"
            verifyTitle "Canoo WebTest"
        }
    }
}
```

Result Summary

WebTests	#	%	Graph
✓	4	67	
✗	2	33	
Sum	6	100	
Steps	#	%	Graph
✓	16	62	
✗	2	8	
◻	8	31	
Sum	26	100	

Server Roundtrip Timing Profile

Secs	#	%	Histogram
0 - 1	23	96	
1 - 3	1	4	
3 - 5	0	0	
5 - 10	0	0	
10 - 30	0	0	
> 30	0	0	
Sum	24	100	

Test Scenario Overview (00:00:12)

#	Result	Name
1	✓	check that WebTest is Google's top 'WebTest' result
2	✓	Search some WebTest steps using Google restricted to webtest.ca
3	✗	[should fail!] Example of WebTest usage in TDD for a not yet existin
4	✓	check that WebTest is Google's top 'WebTest' result [test written as
5	✓	Search some WebTest steps using Google restricted to webtest.ca
6	✗	[should fail!] Example of WebTest usage in TDD for a not yet existin

Created using [Canoo Webtest \(R_1743\)](#). Report created at 18.02.2009 09:50

✓ check that WebTest is Google's top 'WebTest' result [test written as Groovy code]

Test started at Wed Feb 18 09:50:47 CET 2009, lasting 657 ms.

Source: `GoogleWebtestTest.groovy:-1:`

Base URL (used by **invoke** steps with a relative URL): <http://localhost/>

#	Result	Name	Parameter	Duration
1	✓	invoke Go to Google (in English) Resulting page	url <code>http://www.google.com/ncr</code>	311
2	✓	verifyTitle	nested text Google	3
3	✓	setInputField	name <code>q</code> value <code>WebTest</code>	79
4	✓	clickButton Resulting page	nested text I'm Feeling Lucky	264
5	✓	verifyTitle	nested text Canoo WebTest	0

[Back to Test Report Overview](#)



Example: WebDriver

```
@Test
public void htmlUnitSite() throws Exception {
    WebDriver driver = new HtmlUnitDriver();
    // WebDriver driver = new FirefoxDriver();
    // WebDriver driver = new InternetExplorerDriver();
    driver.get("http://www.google.com/");
    WebElement element = driver.findElement(By.name("q"));
    element.sendKeys("HtmlUnit");
    WebElement button = driver.findElement(
        By.xpath("//input[@value = \"I'm Feeling Lucky\"]"));
    button.click();
    assertEquals("HtmlUnit - Welcome to HtmlUnit", driver.getTitle());
}
```



Future Plans

- **Expand AJAX library integration testing**
- **Improved control of background JavaScript**
- **Support for other browsers**
 - Internet Explorer 8?
 - Chrome?
 - Safari?
- **Migrate to Apache?**
- **Frequent releases!**



Questions

