Efficient JavaScript Unit Testing
Hazem Saleh
<table>
<thead>
<tr>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developers Life without Unit testing.</td>
</tr>
<tr>
<td>What is unit testing? and why?</td>
</tr>
<tr>
<td>Current Complexities in testing JavaScript code.</td>
</tr>
<tr>
<td>Requirements of a good JavaScript unit testing tool.</td>
</tr>
<tr>
<td>What is JsTestDriver.</td>
</tr>
<tr>
<td>JsTestDriver Architecture &amp; Configuration</td>
</tr>
<tr>
<td>JsTestDriver Eclipse plugin.</td>
</tr>
<tr>
<td>Writing a JavaScript TestCase.</td>
</tr>
<tr>
<td>JsTestDriver common constructs.</td>
</tr>
<tr>
<td>Writing asynchronous JavaScript TestCase.</td>
</tr>
<tr>
<td>JsTestDriver Compatibility</td>
</tr>
<tr>
<td>Generating reports from test cases.</td>
</tr>
</tbody>
</table>
About Me

• Staff Software Engineer / Consulting IT Specialist in IBM Egypt, Cairo Lab, SWG Services.
• Web 2.0 and WebSphere Portal Subject Matter Expert.
• Apache MyFaces committer.
• Founder of GMaps4JSF.
• Author of the definitive guide to Apache MyFaces book, and reviewer of many other books.
• DeveloperWorks Contributing Author.
• International Technical Speaker in both local and international conferences (such as JavaOne).

• Blog: http://www.technicaladvices.com
• Twitter: http://www.twitter.com/hazems
Developers Life without Unit testing

Complex integration between the system components.
Developers Life without Unit testing

Unmanaged number of new/regression defects especially when the system complexity increases.
Developers Life without Unit testing

Low application quality.

>CONFESS_2012
Developers Life without Unit testing

Long testing cycle.
What is unit testing and why?

Unit testing means testing every component in the system in an Independent way to ensure that it is working properly.

Unit testing helps in detecting BUGGY components in the early stages of the project.

A test case is a set of steps and conditions to test the features and functionalities of the application.
What is unit testing and why?

Integration is much simplified.

Defects are managed. Regression defects should not happen if the defect is resolved by creating a new test case.

Test cases can be a good reference for system documentation.

Test cases can improve the system design and be the basis of code refactoring.

Application quality increases.

Testing cycle is reduced.
Current Complexities in testing JavaScript code

Slow

Requires a lot of time to test on all the browsers.

Inflexible

JavaScript code that runs on a specific browser does not necessarily mean that it will work on other browsers.

Supporting a new browser means allocating a new budget for testing the system again on this new browser and for the new/regression defects fixes.
Requirements of a good JavaScript unit testing tool

- Can execute across all the browsers over all the platforms.
- Easy setup.
- Easy configuration.
- Fast Test case execution.
- Integration with IDEs.
- Integration with build management tools.
Developers Life without Unit testing.

What is unit testing? and why?

Current Complexities in testing JavaScript code.

Requirements of a good JavaScript unit testing tool.

What is JsTestDriver.

JsTestDriver Architecture & Configuration

JsTestDriver Eclipse plugin.

Writing a JavaScript Test Case.

JsTestDriver common constructs.

Writing asynchronous JavaScript Test Case.

JsTestDriver Compatibility

Generating reports from test cases.
What is JsTestDriver

One of the best Open source JavaScript testing tools.

Meets all of the previous requirements and more:

- Supports all the browsers / all platforms. ✓
- Easy setup and configuration. ✓
- Fast Test case execution. ✓
- Integration with IDEs and build management tools. ✓
Developers Life without Unit testing.
What is unit testing? and why?
Current Complexities in testing JavaScript code.
Requirements of a good JavaScript unit testing tool.
What is JsTestDriver.
JsTestDriver Architecture & Configuration.
JsTestDriver Eclipse plugin.
Writing a JavaScript TestCase.
JsTestDriver common constructs.
Writing asynchronous JavaScript TestCase.
JsTestDriver Compatibility
Generating reports from test cases.
JsTestDriver Architecture
JsTestDriver configuration

1. Run the test cases using the following command line:

```
>CONFESS_2012
```
JsTestDriver configuration

1. Download the jsTestDriver latest jars from

   http://code.google.com/p/js-test-driver/downloads/list
JsTestDriver configuration

1. Create the jsTestDriver.conf file (under the JS folder) with the following initial content:

   server: http://localhost:9876
   load:
     - js-src/*.js
     - js-test/*.js
JsTestDriver configuration

1. Start the server using the following command line

   `java -jar JsTestDriver-1.3.2.jar`

2. Optional parameters

   - `[--port 9876]`
   - `[--browser
      "{PATH}\firefox.exe","{PATH}\iexplore.exe","{PATH}\Safari.exe"]`
JsTestDriver configuration

Run the test cases using the following command line

```
java -jar JsTestDriver-1.3.2.jar --tests all
```

........
Total 9 tests (Passed: 9; Fails: 0; Errors: 0) (16.00 ms)
  Firefox 9.0.1 Windows: Run 3 tests (Passed: 3; Fails: 0; Errors 0) (3.00 ms)
  Safari 534.52.7 Windows: Run 3 tests (Passed: 3; Fails: 0; Errors 0) (4.00 ms)
  Microsoft Internet Explorer 7.0 Windows: Run 3 tests (Passed: 3; Fails: 0; Errors 0) (16.00 ms)

>CONFESS_2012
Developers Life without Unit testing.
What is unit testing? and why?
Current Complexities in testing JavaScript code.
Requirements of a good JavaScript unit testing tool.
What is JsTestDriver.
JsTestDriver Architecture & Configuration
JsTestDriver Eclipse plugin.
Writing a JavaScript Test Case.
JsTestDriver common constructs.
Writing asynchronous JavaScript Test Case.
JsTestDriver Compatibility
Generating reports from test cases.
Instead of using command lines for starting the server and running the test cases, you can directly use the JsTestDriver Eclipse plugin.

To install the JsTestDriver Eclipse plugin install the plugin from the following URL: http://js-test-driver.googlecode.com/svn/update/
JsTestDriver Eclipse plugin
JsTestDriver Eclipse plugin
OUTLINE

- Developers Life without Unit testing.
- What is unit testing? and why?
- Current Complexities in testing JavaScript code.
- Requirements of a good JavaScript unit testing tool.
- What is JsTestDriver.
- JsTestDriver Architecture & Configuration
- JsTestDriver Eclipse plugin.
- Writing a JavaScript Test Case.
- JsTestDriver common constructs.
- Writing asynchronous JavaScript Test Case.
- JsTestDriver Compatibility
- Generating reports from test cases.
Writing a JavaScript Test Case

```javascript
ApplicationUtilTest = TestCase("ApplicationUtilTest");

ApplicationUtilTest.prototype.setUp = function () {
    /*:DOC += ...HTML fragment code goes here (single root) ...*/
};

ApplicationUtilTest.prototype.testMethod1 = function () {
    … validate using the jsTestDriver constructs …
}

ApplicationUtilTest.prototype.testMethod2 = function () {
    … validate using the jsTestDriver constructs …
}

...
JsTestDriver common constructs

fail("msg")

assertTrue("msg", actual)

assertFalse("msg", actual)

assertSame("msg", expected, actual)

assertNotSame("msg", expected, actual)

assertNull("msg", actual)

assertNotNull("msg", actual)
DEMO
Let’s write synchronous
JS Test cases …
Developers Life without Unit testing.

What is unit testing? and why?

Current Complexities in testing JavaScript code.

Requirements of a good JavaScript unit testing tool.

What is JsTestDriver.

JsTestDriver Architecture & Configuration

JsTestDriver Eclipse plugin.

Writing a JavaScript Test Case.

JsTestDriver common constructs.

Writing asynchronous JavaScript Test Case.

JsTestDriver Compatibility

Generating reports from test cases.
Writing asynchronous JavaScript TestCase

JsTestDriver provides AsyncTestCase object for performing asynchronous JavaScript unit testing.

In order to test the asynchronous operations, JSTD provides queues.

Each queue contains set of callbacks for testing the Asynchronous system.

>CONFESS_2012
There are two types of callbacks:

- **Normal callback**: MUST be called to verify the Ajax operation success
- **Error callback**: Represents the error path. If it is called, then the test should fail.

The test runner does not move to the next queue until the current queue executes all of its normal callbacks. If a specific normal callback is not called for a specific amount of time (30 seconds), the test fails.
Writing asynchronous JavaScript Test Case

```javascript
WeatherClientTest = AsyncTestCase("WeatherClientTest");

WeatherClientTest.prototype.setUp = function () {
  /*:DOC += <form><div id="weatherInformation"></div></form>*/
};
WeatherClientTest.prototype.testGetWeatherConditionForCairo = function(queue) {
  queue.call('Testing getting weather Information ...', function(callbacks) {
    var weatherClient = new appnamespace.WeatherClient();

    var successCallBack = callbacks.add(function(weatherClient) {
      weatherClient.displayWeatherInformation(weatherClient);
    });

    var failureCallBack = callbacks.addErrback('Unable to retrieve weather information');
    // call asynchronous API
    weatherClient.getWeatherCondition("1521894", "weatherInformation",
      successCallBack,
      failureCallBack);
  });
};
WeatherClientTest.prototype.testGetWeatherConditionForParis = function(queue) {
  //Step 2 ...
};
```
Let’s write Asynchronous JS Test cases ...
JsTestDriver Compatibility

JsTestDriver is not only a JavaScript unit testing framework BUT it is a test runner for many other JavaScript unit testing frameworks.

JsTestDriver is compatibility with the following JavaScript unit testing frameworks through adapters:

- Jasmine
- Yahoo UI Test
- QUnit
JsTestDriver Compatibility

In order to run the previous unit testing frameworks on the top of the JSTD test runner. You need to configure the framework adapter and source before the test files as follows:

server: http://localhost:9876

load:
- jasmine/lib/jasmine-1.1.0/jasmine.js
- jasmine/lib/adapter/JasmineAdapter.js
- js-src/Basics.js
- js-test/BasicsSpec.js
DEMO
Running Jasmine Test cases on the top of JSTD
Developers Life without Unit testing.

What is unit testing? and why?

Current Complexities in testing JavaScript code.

Requirements of a good JavaScript unit testing tool.

What is JsTestDriver.

JsTestDriver Architecture & Configuration

JsTestDriver Eclipse plugin.

Writing a JavaScript Test Case.

JsTestDriver common constructs.

Writing asynchronous JavaScript Test Case.

JsTestDriver Compatibility

Generating reports from test cases.
Generating reports from test cases

JSTD can generate code coverage files.

Code coverage describes how much the source code is tested.

Coverage Criteria:

- Function coverage
- Statement coverage
- Branch coverage
JsTestDriver can generate code coverage for your JavaScript code using the code coverage plugin.
Generating reports from test cases

Configuring the plugin:

1. Download the "coverage.jar".
2. Add the coverage plugin declaration to the configuration file:
   ```
   plugin:
   - name: "coverage"
   jar: "plugins/coverage.jar"
   module: "com.google.jstestdriver.coverage.CoverageModule"
   ```
3. Specify the `--testOutput <<output_folder>>` flag in the test running command.
Unfortunately, JsTestDriver does not generate HTML reports directly, but it generates the test coverage files in LCOV format. You can generate the test reports using the LCOV visualizer tool:

http://ltp.sourceforge.net/coverage/lcov.php

Here is an example of a code coverage report:

<table>
<thead>
<tr>
<th>Directory</th>
<th>Line Coverage</th>
<th>Functions</th>
<th>Branches</th>
</tr>
</thead>
<tbody>
<tr>
<td>example</td>
<td>90.0%</td>
<td>100.0%</td>
<td>75.0%</td>
</tr>
<tr>
<td>example/method</td>
<td>91.7%</td>
<td>100.0%</td>
<td>83.3%</td>
</tr>
</tbody>
</table>

Generated by LCOV version 1.9
Generating reports from test cases

The JsTestDriver LCOV file name is usually:

```
<config filename>-coverage.dat
(jsTestDriver.conf-coverage.dat)
```

To generate the report from the LCOV file using the LCOV visualizer tool:

```
genhtml jsTestDriver.conf-coverage.dat
```

### LCOV - code coverage report

<table>
<thead>
<tr>
<th>Directory</th>
<th>Line Coverage</th>
<th>Functions</th>
<th>Branches</th>
</tr>
</thead>
<tbody>
<tr>
<td>example</td>
<td>90.6%</td>
<td>9/10</td>
<td>1/1</td>
</tr>
<tr>
<td>example/methoda</td>
<td>91.7%</td>
<td>11/12</td>
<td>100.0%</td>
</tr>
</tbody>
</table>

Generated by: [LCOV version 1.9](https://lava.linuxfoundation.org/lcov/1.9)
Conclusion
Conclusion

Testing JavaScript code is important for increasing the application quality and for speeding up fixing defects and minimizing the number of regression defects.

Good JavaScript tool should be configurable, easy to use, and working with all the browsers.

JsTestDriver is one of the most powerful JavaScript unit testing tools that can be used for testing both synchronous and asynchronous JavaScript code on all the browsers.