

Testing JavaScript

Liam McLennan
www.pluralsight.com



Outline

- **Testing?**
- **JUnit**
- **Testing DOM Interactions**
- **BDD**
- **BDD Testing with Jasmine**
- **Integrating testing into the build process**

Testing?

- Coded tests that execute automatically
- Verify that actual behavior matches expected behavior
- Prevent runtime and logical errors

Why?

- **With great power, comes great responsibility**
- **To catch bugs that a compiler would normally pickup**
- **To facilitate confident refactoring**
- **To catch regression bugs**

QUnit

- <http://docs.jquery.com/QUnit>
- Part of the jQuery project – used to test jQuery
- Tests run in a web page
- Modules declared with:

```
module(moduleName);
```

- Tests declared with:

```
test(testName, callback);
```

- **Assertions**
 - `ok(boolean, message);`
 - `equal(actual, expected, message);`
 - `deepEqual(actual, expected, message);`

QUnit Template

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
    <script src="http://code.jquery.com/jquery-latest.js"></script>
    <link rel="stylesheet" href="qunit.css" type="text/css" media="screen" />
    <script type="text/javascript" src="qunit.js"></script>
</head>
<body>
    <h1 id="qunit-header">QUnit example</h1>
    <h2 id="qunit-banner"></h2>
    <div id="qunit-testrunner-toolbar"></div>
    <h2 id="qunit-userAgent"></h2>
    <ol id="qunit-tests"></ol>
    <div id="qunit-fixture">test markup, will be hidden</div>
</body>
</html>
```

Testing DOM Interaction

- QUnit is suited because all tests are in the context of a page
- Tests can manipulate the DOM and then assert that the expected changes happened
- jQuery greatly assists with DOM manipulation and inspection

Behaviour-Driven Development (BDD)

- **Outside-in style of development, starting with tests that specify the behaviour of the application**
- **Scenarios (tests) are often specified using the given / when / then syntax**

Jasmine

- <http://pivotal.github.com/jasmine>
- Behaviour-driven development (BDD) framework for JavaScript

```
describe("some scenario", function () {  
    beforeEach(function () {  
        // establish context  
    });  
  
    it("should behave in a certain way", function () {  
        // do something  
        // assert expectations  
    });  
});
```

Adding JavaScript Testing to Builds

- Jasmine includes a command line entry point that runs the specifications using Selenium
- The result can be used in an automated build
- First you need Ruby[↑]

Summary

- **JavaScript requires more thorough testing than statically typed compiled code**
- **QUnit**
 - Good for unit testing
 - Good for testing DOM interactions
- **BDD**
- **Jasmine**
 - Similar to QUnit but Jasmine adds a BDD DSL

For more in-depth **online** developer **training** visit



on-demand content from authors you **trust**

