QUnit JavaScript Testing in the Enterprise

David Lee

Oct. 11, 2011
Getting to Know You

• How many of you work on Websites that use JavaScript?
• How many currently have a Unit Testing strategy?

• About me.
Outline for Today’s Talk

- What’s All This About Unit Testing?
- What Do You Mean By JavaScript Unit Testing?
- The QUnit Framework
- The End Result
What is Unit Testing?

• What is all this hullaballoo about?
What is Unit Testing?

When the rubber meets the road testing is...

• Developers code tests that
  • describe desired functionality,
  • access this functionality,
  • then verify that this functionality meets expectations

• Reporting
  • Should include reporting
  • And scheduled test runs
Unit Testing Takes Time

• Unit testing requires up front time
• Later this saves time for development and QA
• Time spent up front includes:
  – Developer time:
    • Test creation --- coding of tests
  – Infrastructure Setup:
    • Automation of tests
    • Configuration of testing reports
Unit Testing Has Many Benefits

• Let’s see what some of these are...
Unit Testing Benefits

• Fewer bugs
Unit Testing Gets Your Attention

• Immediate feedback when code goes wrong (in a good way)
A Time Saver

• Overall the process of development, testing and dealing with bugs takes less time
And What else...

• Intercepts errors before they happen
• Keeps developers focused on the requirements
• Reduces workload on QA / Dev
JavaScript Unit Testing?

- Why test JavaScript?
- Typical unit testing strategies
- JavaScript unit testing strategies
- Problems being near the UI
Why JavaScript Testing?

• More and more technologies bring the logic client-side:
  – JSON
  – JQuery
  – AJAX
  – HTML5
  – HTML5 for Mobile devices

• Richer client-side experiences lead to more JavaScript
Why JavaScript Unit Testing?

• With more JavaScript in the client there is more opportunity for code errors in JavaScript
• Relevant behaviors remain untested when ignoring JavaScript
Typically unit testing is done on application code residing in the middle layer.
But JavaScript unit testing is done on script that resides in or near the user interface.
Problems being near the UI

- JavaScript & HTML are hard to separate
- Problems relying on HTML
JavaScript & HTML are hard to separate

• Historically JavaScript & HTML are inseparable
• JQuery make it harder to separate JavaScript from HTML

\$/".className > div"\).empty();
Problems Relying on HTML

• Testing strategies say to test logic and not the User Interface

• If we rely on HTML it can cause problems:
  – HTML is fluid (changeable)
    • Controls, add-ins, database records, etc. all change HTML
  – Designers & Programmers (two hands in the mix)
How to test, but include HTML?

• Option 1: Copy HTML for our tests?
• Option 2: Run our tests directly against the web page?
• Option 3: Not test JavaScript that has a relationship to HTML
• Option 4: Some sort of hybrid approach
Option 1: Copy HTML for our tests?

- HTML may be required, so should we copy it for our tests?

  Static HTML may become old and irrelevant
Option 1: Copy HTML for our tests?

Copy HTML
Option 2: Run tests against the page

- Tests could modify or significantly change the behavior / look / feel of a web page
- Example: running a test that inserts a block of text... what happens at the end of the test.
Option 3: Avoid HTML Altogether

• We ruled this out because it limited the number of tests we were able to create.

• Many pages with significant JavaScript would remain untested.
The QUnit Framework

• A look at QUnit examples
The End Result

- Our HTML Hybrid Testing Strategy
- Our JavaScript Testing Strategy
Our HTML Hybrid Testing Strategy

• Runs tests in an IFRAME
• Copies HTML from the actual page immediately
Our JavaScript Testing Strategy

- Make it easy to wire in tests
- Use site-wide hooks to keep the developer’s job easy
- Put test failures and successes in plain view
And Some Final Thoughts...

• Unit Testing is a valuable resource for YOU!
• The Use of JavaScript Unit Testing is timely and relevant given today’s technologies.
• To either deny or use HTML in your JavaScript tests will take forethought. In many cases it is not a simple decision.
• The provided libraries and sample tests will hopefully give you a jump start into the world of JavaScript Unit Testing.
Questions

• Questions?
QUnit JavaScript Testing in the Enterprise

Thank you!