Dojo Utilities & Tools



Dojo Utilities & Tools

- Testing via DOH
 - > Functional testing
 - > Performance testing
- Build system
- ShrinkSafe
- Debugging via Firebug

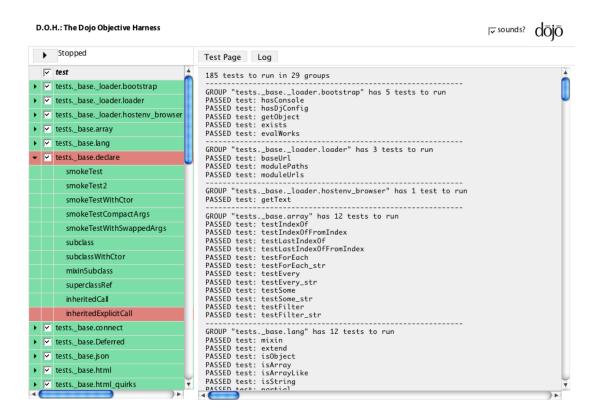
Where do you get these tools?

Comes with Dojo toolkit source package

Testing via Dojo Objective Harness (DOH)

The Dojo Objective Harness (DOH)

- Dojo Objective Harness ("d.o.h")
 - New unit test harness
 - Command-line or within the browser
- Unit test JavaScript and aggregate tests into a single interface



Unit Tests Supported

- Basic DOH infrastructure is for code-only tests
 - > DOH Robot adds UI testing in DojoX.
- Support for setup and tear-down.
- Build a test around a page.
- Include all of the different tests styles in a test suite.

DOH Test Assertions

- doh.assertTrue(x), doh.t(x)
- doh.assertFalse(x), doh.f(x)
- doh.assertEqual(x, y), doh.is(x, y)

Build System

What is and Why Use Dojo Build System?

- To achieve the improved performance and better user experience by
 - Reducing the number of files sent over the wire
 - > Reducing the JavaScript file sizes

Dojo Build System 4-Step Process

- First, it groups together modules into "layers".
 - A layer, which is one big .js file, loads faster than the individual .js modules that comprise it
- Second, it "interns" external non-JavaScript files.
 - > This is most important for Dijit templates, which are kept in a separate HTML file. Interning pulls the entire file in and assigns it to a string.
- Third, it smooshes the layer down with ShrinkSafe.
 - ShrinkSafe removes unneeded whitespace and comments, and compacts variable names down to smaller ones. This file downloads and parses faster than the original.

Dojo Build System 4-Step Process

- Finally, it copies all non-layered scripts to the appropriate places.
 - While this doesn't speed anything up, it ensures that all Dojo modules can be loaded, even if not present in a layer.
 - If you use a particular module only once or twice, keeping it out of the layers makes those layers load faster.

ShrinkSafe

What is ShrinkSafe?

- ShrinkSafe is a standalone Java-based JavaScript compressor which utilizes Rhino to parse code and safely shorten the results.
- ShrinkSafe renames local references to short names prefixed with an underscore.
 - This saves bytes on the wire and also provides some obfuscation of the code.
- It also eliminates whitespace and comments when generating the new code.
- Global references and property names remain unchanged such that external references to the compressed code should be safe.

Example: ShrinkSafe

Compresses bigcode.js to smallercode.js

```
java -jar shrinksafe.jar bigcode.js > smallercode.js
```

Compresses three files into one

```
java -jar shrinksafe.jar file1.js file2.js file3.js >
combined.js
```

Debugging via Firebug or Firebug Lite

Firebug or Firebug Lite

- Firebug, an open source debugging extension for Firefox, is "essential" for JavaScript, HTML and CSS debugging
- If you use Internet Explorer or Safari, you can use the Firebug Lite library, bundled with Dojo.
 - This gives you some of the logging and command line features of Firebug.
 - It's not a full emulation, but it's a fairly good alternative and is fully API-compatible
 - To use Firebug Lite, you must include the isDebug config parameter no effect to Firefox/Firebug

```
<script type="text/javascript" src="http://path/dojo.js"
djConfig="parseOnLoad: true, isDebug: true"></script>
```



Thank you!

Check JavaPassion.com Codecamps!
http://www.javapassion.com/codecamps
"Learn with Passion!"

