AJAX/JS Framework Showdown - jQuery vs Spry

Posted At : October 29, 2008 1:57 PM | Posted By : Charlie Griefer
Related Categories: Tools of the Trade, jQuery, Spry, ColdFusion

In an ongoing effort to put together our development standards at Amcom, we're currently researching AJAX/JS frameworks. The choices have been essentially narrowed down to Spry or jQuery.

I haven't done a considerable amount of work with either, but my personal preference leans towards jQuery. I'm more familiar with it than Spry (while I haven't really gotten my hands too dirty with it, I've used a number of jQuery plugins), and being someone who likes JavaScript I'm comfortable with the syntax.

The boss, however, being a big Flex guy, really gravitates towards the Spry syntax, as it's more familiar to him. The boss, being a fair and noble kind of boss (yes, he reads the blog), told me to go out and do an objective analysis of the two and see if one comes out significantly ahead of the other.

I built a few demo apps using each (http://amcomtech.net/labs/ajaxframeworks/) and from a technical perspective I have to admit, it's still kind of a tie.

In putting together comparable demos of each, they both allowed me to put together some pretty slick functionality in a relatively short period of time. Neither required significantly more (or less) code to implement than the other.

In the table-striping example, here's the relevant jQuery code:

```
$(function() {
 $('#striped tbody tr:even').addClass('odd');
 $('#striped tbody tr:odd').addClass('even');
});
```

...and the relevant Spry snippet (which goes right into the HTML, not JavaScript):

```
<tr spry:repeat="dsEmployees" spry:even="even" spry:odd="odd">
 <td>{employeeID}</td>
 <td>{lastName}</td>
 <td>{firstName}</td>
</tr>
```

Both are pretty straightforward. The advantage to the Spry code is that, if you're not familiar with or comfortable with JavaScript, you can easily implement the effect in HTML. The downside that I can see is that you lose the ColdFusion context of the output data. They're now Spry variables. Being comfortable with ColdFusion and knowing how to manipulate those values (should the need arise) seems like a pretty big loss to me.

In the filter function comparison, both are pretty even as well. Both the jQuery and Spry solutions require some JavaScript to be written. Although with the jQuery solution I'm essentially just showing/hiding table rows as opposed to applying a true filter to the data. The Spry solution appears to actually manipulate the result set. In fact, it provides a built variables ({ds_RowCount} and {ds_UnfilteredRowCount}) that make it very easy to say, "Displaying 'x' of 'y' Records". The same thing could be done in jQuery, but it'd be a manual process to calculate the value.

Loading external content into a div was also equally as simple. 1 liner for jQuery, 2 liner for Spry. Not a significant difference there.

So it seems that from an implementation perspective, it's still a pretty tight race. There's no compelling reason to choose one over the other. Time to broaden the scope of the evaluation.

As far as I can tell, jQuery is a bit more... robust, shall we say. There's huge support for it in the community. New plugins are developed and released on a pretty regular basis. As I alluded to before, I've not really written much jQuery myself, but I've been able to incorporate some pretty cool effects by grabbing some of the available jQuery plugins.

There are also (to my knowledge) 2 books published on jQuery. It's got a high-traffic mailing list (to be fair, Spry may have one as well... have not looked). I would think that support for jQuery would be more readily available.

It's also my belief (or feeling) that beyond the samples that are available for Spry (auto complete, various widgets, etc), that jQuery is more robust in allowing you to manipulate the elements on the page. In the sample app that I put together, I used a frameset with a list of links to each demo on the left. I wanted to have some sort of indicator in that navigation to show which page the user is currently on. I've done this plenty in ColdFusion, using a series of conditionals. But using jQuery, it was as simple as:

```
$(function() {
 $('a').click(function() {
  $('a').removeClass('activeLink');
  $(this).addClass('activeLink');
 });
});
```

That's it, in its entirety. No convoluted conditionals in the markup making it difficult to read. I'm not sure that Spry allows for that kind of manipulation of elements on the page.

So, the (semi) conclusion that I've drawn is that both frameworks make it painfully easy to incorporate some pretty nice client side features. I can honestly say that I was pleasantly surprised by how easy Spry made it. Not that it was any easier (or any more difficult) than jQuery... I just had a different set of expectations going in :)

What it will likely come down to is the community. The community will be that compelling argument that I can present to show that there's justification in choosing jQuery over Spry. The support is there. The documentation is there (in the way of both web site and books, as well as hundreds of blog entries by so many different jQuery users). I think that during development, when we hit a wall (and we hit 'em hard when we do), there will be more avenues available to us to get us around (or through) the wall.

What say you? Agree? Disagree? Is there anything that I've overlooked or anything that I've looked at incorrectly? Thoughts would be greatly appreciated, as I don't think the vote has been quite decided yet (although we are waiting on the ballots from FL and expect that will clear everything right up).

As an aside, I've zipped up the demo files and attached them to this entry. They should work out of the box, as long as you put them under your web root in a directory called "labs". You can also view them online at http://amcomtech.net/labs/ajaxframeworks/.

> **UPDATE:**
> A new page has been added to the sample code online (as well as included in the zip) to demonstrate a more "pure" JavaScript approach to table striping via Spry. Thanks to Ray Camden.