**A Sample Application: Part 2**

So far, we’ve written a lot of code that does approximately nothing, but that’s OK. By specifying all the behaviors that we want our core objects to have, we’ve created a solid framework for rapidly building out the interface.

Let’s start by hooking up our text input and the “Load Trending Terms” button. For the text input, we’ll capture the term that was entered in the input and pass it as we trigger the Twitter container’s getResults event. Clicking the “Load Trending Terms” will trigger the Twitter container’s getTrends event:

$('form').submit(function(e) {

e.preventDefault();

var term = $('#search\_term').val();

$('#twitter').trigger('getResults', [ term ]);

});

$('#get\_trends').click(function() {

$('#twitter').trigger('getTrends');

});

By adding a few buttons with the appropriate IDs, we can make it possible to remove, collapse, expand, and refresh all results containers at once, as shown below. For the remove button, note how we’re passing a value of true to the event handler as its second argument, telling the event handler that we don’t want to verify the removal of individual containers.

$.each(['refresh', 'expand', 'collapse'], function(i, ev) {

$('#' + ev).click(function(e) { $('#twitter div.results').trigger(ev); });

});

$('#remove').click(function(e) {

if (confirm('Remove all results?')) {

$('#twitter div.results').trigger('remove', [ true ]);

}

});

**Conclusion**

Custom events offer a new way of thinking about your code: they put the emphasis on the target of a behavior, not on the element that triggers it. If you take the time at the outset to spell out the pieces of your application, as well as the behaviors those pieces need to exhibit, custom events can provide a powerful way for you to “talk” to those pieces, either one at a time or en masse. Once the behaviors of a piece have been described, it becomes trivial to trigger those behaviors from anywhere, allowing for rapid creation of and experimentation with interface options. Finally, custom events can enhance code readability and maintainability, by making clear the relationship between an element and its behaviors.

You can see the full application at demos/custom-events.html and demos/js/custom-events.js in the sample code.