

jQuery Auto-Filtering Table

Posted At : October 2, 2008 6:47 AM | Posted By : Bob Silverberg

Related Categories: jQuery

I've been using [jQuery](#) for about a year now and I must admit that pretty much any JavaScript that I wrote prior to that looks ugly to me now. I know I'm nowhere near the first person to say this, but if you write JavaScript and you aren't using jQuery, you should check it out. Now.

Anyway, I wanted to show you how easy it is to do something cool with jQuery. The next thing I'll admit is that much of what I'm doing with jQuery involves using existing plugins written by other, more seasoned JavaScripters than me. But enough talk, here's the table:

Country	Province/State	City
Canada	Alberta	Calgary
Canada	Alberta	Edmonton
Canada	British Columbia	Vancouver
Canada	British Columbia	Victoria
Canada	Ontario	Ajax
Canada	Ontario	Kingston
Canada	Ontario	London
Canada	Ontario	Ottawa
Canada	Ontario	Toronto
United States	California	Los Angeles
United States	California	San Francisco
United States	California	San Jose
United States	Texas	Dallas
United States	Texas	Houston
United States	Texas	Lubbock

You'll notice that you can sort the table by any of the columns, in both ascending and descending order, and even by multiple columns if you hold down the Shift key while clicking. That sortable behaviour, and the nice zebra striping is all part of the awesome [tablesorter](#) plugin, written by [Christian Bach](#).

Now see what happens when you click on a Country or a Province/State in the table. That's right, the table filters itself based on your selection. Clicking on the same Country or Province/State removes the filter. Once the table is filtered you can still sort on any or all of the columns. This filtering functionality is provided by the only slightly less awesome [uiTableFilter](#) plugin, written by [Greg Weber](#).

All I had to do was write a few lines of code to implement the plugins and set up the events to be fired when clicking the data in the table:

```
<script type="text/javascript">
$(document).ready(function() {
  $table = $("#myTable")
  .tablesorter({widthFixed: true, widgets: ['zebra']});
  FilterText = "";
  ColumnArray = ["Country", "Province/State"];
  for (i=0; i<ColumnArray.length; i++) {
    $("#myTable tbody tr").find("td:eq(" + i + ")").click( function() {
      clickedText = $(this).text();
      FilterText = ((FilterText == clickedText) ? "" : clickedText );
      $.uiTableFilter( $table, FilterText, ColumnArray[i]);
    });
  }
});
</script>
```

First I make the table, which has an id of "myTable", sortable using the tablesorter plugin. Next I create a couple of local variables:

- FilterText will hold the current value that is being used to filter the table.

- ColumnArray is an array of column headings that I want to be clickable.

I then loop through the ColumnArray assigning a click event to the proper column of each table row. The click event captures the text that was clicked on and saves it to clickedText. clickedText is then compared to FilterText to determine whether to apply a new filter (if a new value was clicked) or remove the filter (if the same value was clicked). Finally the new filter is applied to the proper column of the table.

And there you have it, a pretty neat trick accomplished with very few lines of JavaScript Code.

Regarding the [tablesorter](#) plugin, it truly is awesome and can do a lot more than I have described in this post. In addition to being incredibly configurable, it also has a paging add-on that gives you a full featured paging control for your table.