

jQuery: making HTML5 custom data attributes work in XML

jQuery is not designed specifically to work with XML documents. So creating an XML counterpart of the HTML5 custom data attributes in XML becomes pretty complicated if we decide to use jQuery. Despite these problems, this is something that can be achieved if we know XML and the DOM.

We start with the following XML document:

```
<?xml version="1.0" encoding="utf-8" ?>
<document xmlns="http://www.w3.org/1999/xhtml"
xmlns:data="http://site.com/ns/data">
    <ul>
        <li>Test</li>
        <li>Test</li>
        <li><data:action type="click"
message="Test">Test</data:action></li>
    </ul>
    <!-- Scripts here -->
</document>
```

The XHTML namespace has been inserted just for convenience. By doing so, web browsers will recognize our elements. The only exception is our custom data element which belongs to the data namespace. This element has two attributes, namely type and message which will be used by jQuery to add a custom event to the element:

```
(function($) {  
    $.xmldata = function(element, options) {  
        options = $.extend({  
            callback: function() {}  
  
        },  
        options);  
  
        var message =  
element.getAttribute('message');  
        var type =  
element.getAttribute('type');  
        element.addEventListener(type,  
function() {  
            options.callback(message);  
  
        }, false);  
  
    };  
  
})(jQuery);
```

Bear in mind that we need to select an element which belongs to a given namespace. Remember that from a DOM perspective `data:action` is made up by two components: the namespace prefix (`data`) and the local name of our element (`action`):

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"></script>
<script type="text/javascript" src="xmldata.js">
</script>
<script type="text/javascript">
    <![CDATA[
        document.addEventListener('DOMContentLoaded',
function() {

            var list =
document.getElementsByTagName('ul')[0];
            var action =
list.getElementsByTagNameNS('http://site.com/ns/data'
, 'action')[0];

            $.xmldata(action, {
                callback: function(msg) {
                    alert(msg);

                }

            });

        });

    ]]>
</script>
```

In short: by using namespaces we can bring the idea behind HTML5 custom data attributes even in XML.