ASP.NET & jQuery

The Write Less, Do More Library

Overview

- **jQuery History & Motivation**
- **Integration with Visual Studio**
- **jQuery Features**
  - $ - the jQuery function
  - Selectors
  - Events
  - Chaining
  - Effects
  - DOM Manipulation
  - AJAX
- **Plugins**

A Brief History of jQuery

- **Open source project by John Resig (http://ejohn.org/)**
  - Started in 2005
- **Goals**
  - Leverage CSS selectors
  - No syntactic fluff
  - Lightweight
  - Be compatible
  - Be extensible
- **jQuery Today**
  - The most popular JavaScript framework
  - Still open source software
  - 1000+ plug-ins available

```
$('#header > b').click(function() {
    alert(this.innerHTML);
});
$('.content').mouseover(function() {
    this.innerHTML = 'Content replaced';
});
```

jQuery and Microsoft

- **jQuery ships with Visual Studio (post 2008)**
  - For use with ASP.NET and ASP.NET MVC
  - Fully supported product
- **Works with ASP.NET AJAX libraries**
  - Side by side
  - Future integration
- **Using jQuery**
  - Include jQuery-<version>.js file
  - Use jQuery-<version>-vsdoc.js for intellisense support only

```
/// <reference path="jquery-1.2.6-vsdoc.js" />

function pageLoad() {
   // ...

}
```

```
<asp:ScriptManager ID="ScriptManager1" runat="server">
    <Scripts>
        <asp:ScriptReference Path="~/scripts/jquery-1.2.6.js" />
        <asp:ScriptReference Path="…" />
    </Scripts>
</asp:ScriptManager>
```

$ function

- **$ is the jQuery function**
  - Returns a jQuery object
  - $ is “overloaded” to perform different tasks
- **Analyzes incoming arguments**
  - String (selector expression to select elements)
  - String (containing HTML for manipulation)
  - DOM elements (to bind events or set properties)
  - Function object (invoked on document ready)

```
/// <reference path="jquery-1.2.6-vsdoc.js" />

$(function() {

    $("<div>Hello, World</div>").appendTo(document.body);
    $("div").wrapInner("<h1></h1>");
});
```

Selectors

- **Selectors find DOM elements to manipulate**
  - Based on CSS Selector syntax
  - Find by id, class, element name and hierarchical positioning
- **Selectors return a sequence of matching DOM elements**
  - Later methods operate on every matched element
  - Like LINQ, can then create processing pipeline

```
<div id="menu">
    <p>ASP.NET Instructors</p>
    <div class="section">
        <a href=“#">Fritz</a>
        <a href="#">Scott</a>
    </div>
    <p>ASP.NET AJAX Instructors</p>
    <div class="section">
...
```

```
$(function() {
    $("#menu").width(150);
    $("#menu > p").css("background-color", "#CCCC99");
    $("a").css("display", "block");
});
```

Selector Reference

Pattern	Meaning
*	Match all elements
E	Any element of type E
E F	Any F that is a descendant of E
E > F	Any F that is a child of E
E + F	Any F that is immediately preceded by an E
E[foo]	Any E element with a foo attribute
E[foo="bar"]	Any E element with a foo attribute of bar
#id (*#id)	An element with an ID of id
.classname (*.classname)	Any element with a class of classname

Full list available at:
http://www.w3.org/TR/css3-selectors/

Selector Pseudo-classes

- **Further refinement of a selection**
  - Always starts with a colon (:)
  - Examples include :link, :visited, :odd, :even, :not, :first

```
$("a:even").css("background-color", "#CC99CC");
$("a:odd").css("background-color", "#AA99AA");
```

```
$("div:not('#menu')").css("padding-left", "10px");
```

```
$("a:contains('Scott')").wrap("<em><b></b></em>");
```

Working with Attributes

- **The attr() function**
  - Reads a property value from first matched element
  - Set one or more properties on all matched elements
- **Class functions**
  - addClass()
  - removeClass()
  - toggleClass()
- **Others**
  - html()
  - val()
  - text()

```
$("a:even").addClass("evenrow");
$("a:odd").addClass("oddrow");

$("a:contains('Scott')").attr("href",
   "http://www.pluralsight.com/main/instructor.aspx?name=scott-allen");
```

Events

- **Generic bind() function**
  - Binds a function to any event by name
- **Event helpers**
  - Dedicated methods for common events (click, blur, submit, etc).
- **Pass in function to execute**
  - Invoked with this reference set to each matching element

```
$("a").mouseover(function() {
    $(this).addClass("highlight");
});

$("a").mouseout(function() {
    $(this).removeClass("highlight");
});
```

Chaining

- **jQuery uses a builder design pattern**
  - Each method call returns the jQuery object
  - Continue to work with the sequence of matched elements by chaining method calls
  - The end() function reverts last “destructive” filter

```
$(function() {
    $("a").mouseover(function() { $(this).addClass("highlight"); })
          .mouseout(function() { $(this).removeClass("highlight"); })
          .filter(":even").addClass("evenrow").end()
          .filter(":odd").addClass("oddrow").end()
          .filter(":contains('Scott')").attr("href",
            "http://www.pluralsight.com/main/instructor.aspx?name=scott-allen");
});
```

Effects

- **Basics**
  - hide, show, toggle
- **Sliding**
  - slideUp, slideDown, slideToggle
- **Fading**
  - fadeIn, fadeOut, fadeTo
- **Custom animate function**

```
$(".section").hide();

$("#menu > p").click(function() {
    $(this).next().slideToggle("slow");
});
```

Server Communication

- **High level options**
  - load()
  - get()
  - post()
  - getJSON()
- **Low level options**
  - ajax()

```
$(function() {
    $("a").click(getInstructorInfo);
}
```

```
function getInstructorInfo(event) {
    var instructorName = $(this).text();
    $("#detail").load("getInstructorInfo.ashx",
                      { "instructor": instructorName });
    return true;
}
```

Using get() and post()

- **Allows access to raw XmlHTTPRequest object**
- **Packages data into query string (GET) or form values (POST)**
- **Callback function invoked to process result**
  - Only invoked on successful completion

```
function getInstructorInfo(event) {

    var instructorName = $(this).text();
    $.post("getInstructorInfo.ashx",                          // url
          { "instructor" : instructorName },                  // data
          function(result) { $("#detail").html(result) },     // callback
          "html");                                            // type of data
}
```

Error Handling

- **Need to use lower level ajax() method or ajaxError() method.**
- **With ajax() method, single parameter specifies all options**
  - Includes success and error callback functions

```
var instructorName = "Dr. Evil";
$.ajax(
    {
        type: "POST",
        url: "getInstructorInfo.ashx",
        data: { "instructor": instructorName },
        timeout: 5000, // ms
        success: function(result) { $("#detail").html(result) },
        error: function(xhr, status, exception) {
           $("#detail").html(
                    "There was an error.<br/> " +
                    "Status: " + status + "<br/>" +
                    "XHR Status: " + xhr.statusText + "<br/>");
            }
     }
);
```

Global AJAX Events

- **Local events are callbacks for specific AJAX calls**
- **Global events raised for all jQuery AJAX calls**
  - Can disable on a per-call basis

```
$("#log").ajaxComplete(function(event, xhr, options) { $(this).append("complete <br/>"); })
         .ajaxError(function(event, xhr, options) { $(this).append("error <br/>"); })
         .ajaxSend(function(event, xhr, options) { $(this).append("send <br/>"); })
         .ajaxStart(function(event, xhr, options) { $(this).append("start <br/>"); })
         .ajaxStop(function(event, xhr, options) { $(this).append("stop <br/>"); })
         .ajaxSuccess(function(event, xhr, options) { $(this).append("success <br/>"); });
```

Calling WCF Services

- **Requires webHttpBinding**
  - AJAX-enabled WCF service
- **jQuery will attempt to automatically parse JSON results**
  - Doesn't work with DateTime fields
- **Consider using a JSON library**

```
public class Instructor
{
    public string Name { get; set; }
    public string[] Traits { get; set; }
    public DateTime BirthDate { get; set; }
}
```

```
function getInstructorInfo(event) {
    var instructorName = $(this).text();
    $.ajax(
        {
            type: "POST",                                          // POST by default
            url: "InstructorInfoService.svc/GetInstructorInfo",    // svc endpoint
            data: '{ "name": "' + instructorName + '" }',          // json format
            timeout: 5000,                                         // ms
            contentType: "application/json",                       // accept header
            dataFilter: parseJson,
            success: onSuccess
        });
}
```

```
[OperationContract]
public Instructor GetInstructorInfo(string name)
{
   // ...
}
```

jQuery UI

- **Implemented in ui.jQuery.js**
  - Abstractions for low-level interaction (sortables, draggables)
  - More effects (explode, pulsate, bounce)
  - Includes widgets (accordian, date picker)
  - Download from ui.jquery.com

```
$("#menu").resizable();
$("#menu > p").click(function() {
    var section = $(this).next();
    if (section.css("display") == "none") {
        section.slideDown("slow");
    }
    else {
        section.hide("explode", { pieces: 20 }, 1000);
    }
});
```

jQuery Plugins

- **jQuery UI is one plugin**
- **Over 1000 more available at plugins.jquery.com**
  - Animation
  - Layout
  - Media
  - Data
- **Many extensibility points**
  - Add new methods, functions
  - Add or extend selectors
  - Extend existing methods

```
jQuery.log = {
  error: function() { /**/ },
  warning : function() { /**/ },
};

jQuery.fn.debug = function() {
    return this.each(function() {
        alert(this);
    });
};

// ...
$.log.error(/**/);

$("a").debug();
```

jQuery & MicrosoftAjax

	jQuery	Microsoft Ajax
Server-side integration		★★★
CLR feel & features		★★★
JSON and WCF Support	★	★★★
Animations	★★★	★
Lightweight	★★★	
DOM Selection	★★★	★
CSS / Attribute Manipulation	★★★	

Summary

- **jQuery is complementary to ASP.NET AJAX**
  - DOM element selection
  - DOM element manipulation
- **Leverages features of JavaScript language**
  - Functional programming with closures
  - Dynamic language features for extensibility
- **Myriad of communication options**
  - But no real JSON support
  - Still need client side data binding
- **Plug-ins available for almost every scenario**