AJAX with ASP.NET MVC

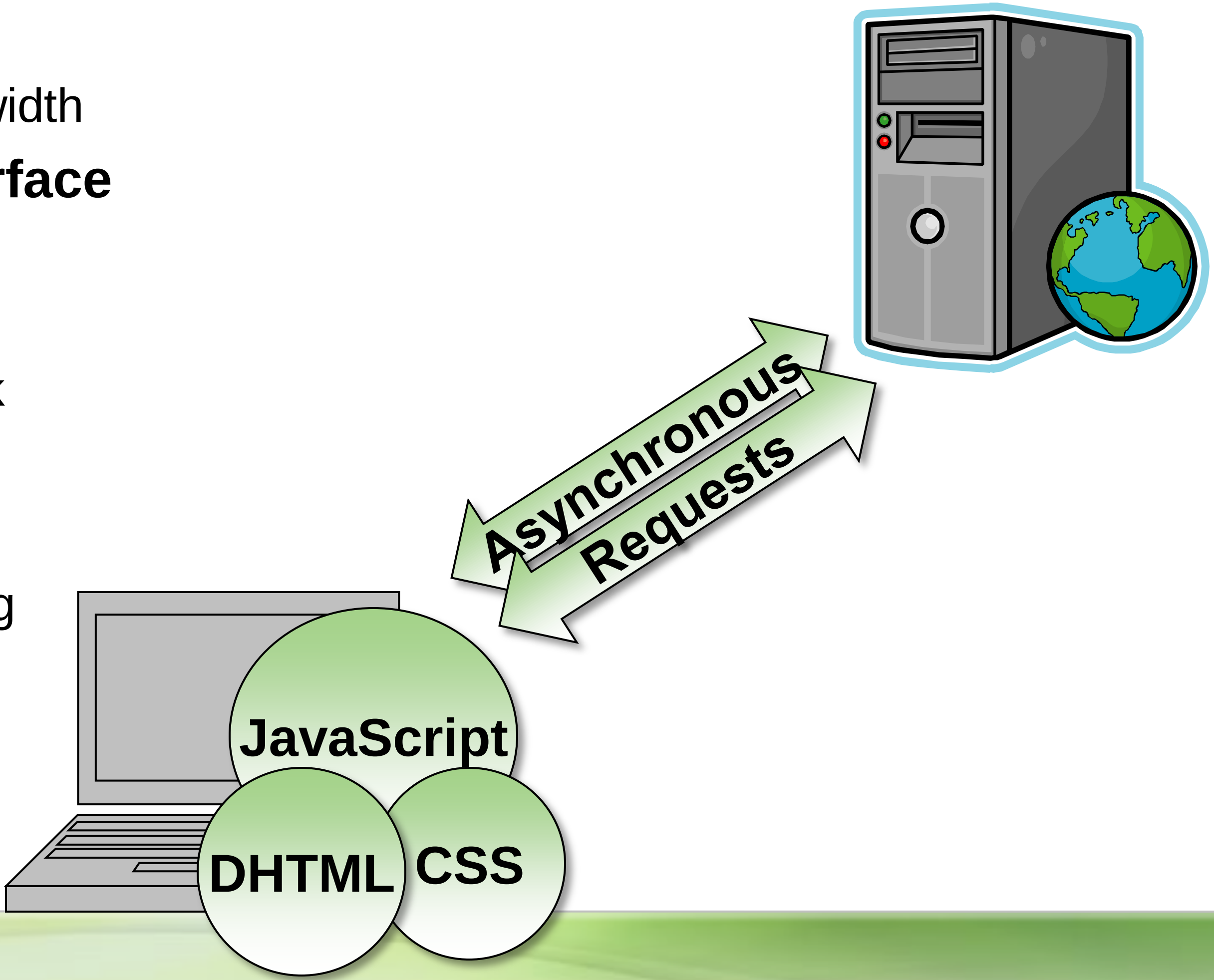
pluralsight
see what you can learn

Overview

- Ajax Helpers
- ASP.NET AJAX Client Libraries
- Ajax & Partial Views
- jQuery
- JSON Results

Motivations for AJAX

- **Build a more responsive application**
  - Less wait time
  - Use less bandwidth
- **Build a rich interface**
  - Less flicker
  - More animation
  - Better feedback
- **Downsides**
  - Harder to test
  - Harder to debug



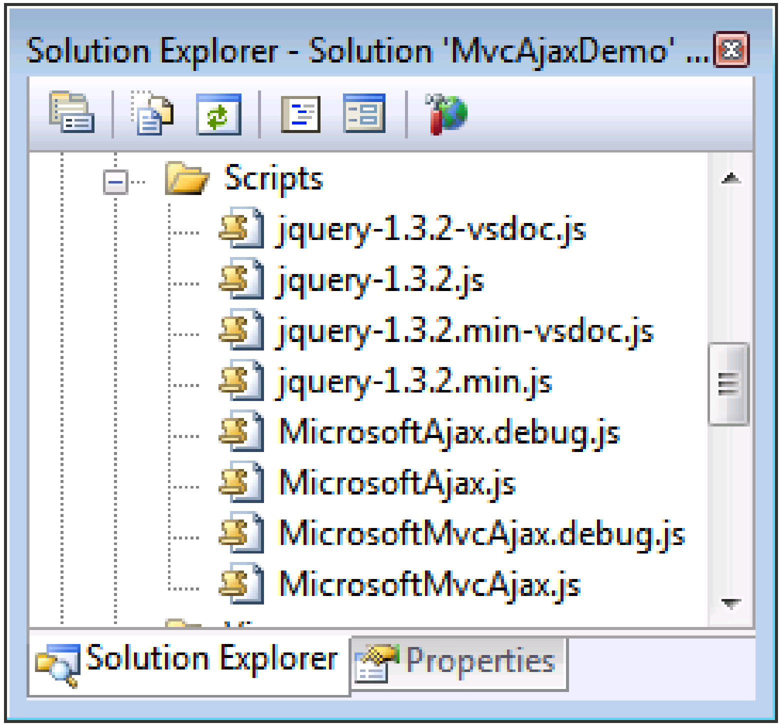
Raw Ajax

- **Tedious and error prone**

```javascript
function getServerTime() {
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "/Home/ServerTime", true);
    xhr.onreadystatechange = function() {
        if (xhr.readyState == 4) {
            if (xhr.status == "200") {
                var timeDiv = document.getElementById("timeDisplay");
                timeDiv.innerHTML = xhr.responseText;
            }
        }
    }
    xhr.send();
}
```

AJAX Infrastructure

- **Microsoft AJAX**
  - Component orientation
  - OOP Style
  - Gives CLR flavor to JavaScript
  - Support for WCF and JSON
- **jQuery**
  - Functional orientation
  - Plug-in oriented
  - CSS Selectors
  - DOM Manipulation


Solution Explorer - Solution 'MvcAjaxDemo' ...

Scripts
- jquery-1.3.2-vsdoc.js
- jquery-1.3.2.js
- jquery-1.3.2.min-vsdoc.js
- jquery-1.3.2.min.js
- MicrosoftAjax.debug.js
- MicrosoftAjax.js
- MicrosoftMvcAjax.debug.js
- MicrosoftMvcAjax.js

Managing Scripts

- **Script modes**
  - "Debug" versus "Release"
  - No built-in mode switching
  - jQuery Intellisense files for development only
- **Injecting scripts**
  - <script> tags
  - Url helper

```
<script src="../../Scripts/MicrosoftAjax.js" type="text/javascript" />
<link href="../../Content/Site.css" rel="stylesheet" type="text/css" />
```

```
<script src="<%= Url.Content("~/Scripts/MicrosoftAjax.js") %>"
        type="text/javascript" /> </script>
<link href="<%= Url.Content("~/Content/Site.css") %>" rel="stylesheet"
      type="text/css" /></link>
```

Ajax Helpers

- **Create links and forms that send async requests**

```
<div id="timeDisplay">
    <%= Ajax.ActionLink("Click here to set the server time",
                        "ServerTime",
                        new AjaxOptions
                        {
                            UpdateTargetId="timeDisplay",
                            HttpMethod="GET", // default
                            InsertionMode= InsertionMode.Replace, // default
                        })
    %>
</div>
```

AjaxOptions

- **Includes events for**
  - OnBegin
  - OnComplete
  - OnFailure
  - OnSuccess
- **Confirmation prompt**

```
<%= Ajax.ActionLink("Click here to set the server time",
                    "ServerTime",
                    new AjaxOptions
                    {
                        LoadingElementId="loadingDisplay",
                        Confirm="Are you sure?",
                        UpdateTargetId="timeDisplay",
                    })

%>
```

```
<div id="loadingDisplay" style="display:none">
    <img src="../../Content/spinner.gif" />
</div>
```

Partial Page Rendering

- **Use PartialViewResults in controller actions**

```
if (Request.IsAjaxRequest())
{
    return PartialView("MovieTable", movies);
}

return View("Index", movies);
```

```
<% using (Ajax.BeginForm("Index", "Movie",
             new AjaxOptions { HttpMethod="GET",
                               UpdateTargetId="movieTable" }))
   { %>
        <input type="text" name="q" />
        <input type="submit" value="Search" />
<% } %>
```

Ajax Helpers and Errors

- **Default behavior is to fail silently**
- **Override default by specifying OnFailure option**

```
function searchFailed(ajaxContext) {
    var response = ajaxContext.get_response();
    var element = ajaxContext.get_updateTarget();
    element.innerHTML = "Error: server returned a " +
                         response.get_statusCode();
}
```

Beyond the Built-in Helpers

- **Ajax Helpers cover simple scenarios**
  - Replacing HTML content
  - Partial page rendering
- **Other scenarios require some JavaScript coding**
  - Auto-complete textboxes
  - Client-side validation
  - Invoking JSON services and actions
  - Animations

Auto-Complete

- **jQuery auto-complete plug-in**
  - http://bassistance.de/jquery-plugins/

```
$(function() {
   $("#searchBox").autocomplete("/Movie/SearchCandidates",
                                 { minChars: 3 });
});
```

```
public string SearchCandidates(string q, int limit)
{
    var ctx = new MoviesContext();
    var movies = ctx.MovieSet
                    .Where(m => m.Title.StartsWith(q))
                    .OrderByDescending(m => m.ReleaseDate)
                    .Take(limit)
                    .Select(m => m.Title);

    return String.Join(Environment.NewLine, movies.ToArray());

}
```

Client Validation

- **jQuery validate plug-in**
  - Don’t forget you need server-side validation, too!

```
$(function() {
    $(".create > form").validate({
        rules: {
            Title: {
                required: true,
                minlength: 3,
                maxlength: 50
            },
            ReleaseDate: {
                required: true,
                date: true
            }
        }
    });
});
```

Date Pickers and Other Widgets

- **Many lightweight jQuery widgets available from jQuery UI**
  - http://jqueryui.com/
  - Calendar
  - Accordian
  - Slider
  - Dialog
  - Tab
  - Progress Bar

```
$(".create #ReleaseDate").datepicker();
```

JSON & MVC

- JsonResult will serialize objects to JSON
- jQuery and ASP.NET AJAX can work with JSON

```
$.getJSON("/Instructor/InstructorNames", "", function(data) {
  $(data).each(function() {
    …
  });
});
```

```
public JsonResult InstructorNames()
{
    var repository = new InstructorRepository();
    var names =
            from i in repository.FindAll()
            select new
            {
                ID = i.ID,
                Name = i.Name
            };
    return Json(names);
}
```

WCF Services

- **Available through ASP.NET AJAX proxies**

```
<script src="../../Services/InstructorService.svc/js" type="text/javascript"></script>
```

```
var service = new InstructorService();
service.GetInstructorNames(function(result) {
    $(result).each(function() {
        $("<option>").val(this.ID)
                     .text(this.Name)
                     .appendTo("#instructorSelect");
    });
});
```

```
[OperationContract]
public Instructor GetInstructorDetails(int id)
{
    var repository = new InstructorRepository();
    var instructor = repository.FindByID(id);
    return instructor;
}
```

Summary

- **ASP.NET MVC has basic AJAX building blocks**
  - AJAX Helpers
  - Partial Views
  - Client consumable action results
- **ASP.NET MVC AJAX Client Libraries**
  - Invoke web services
  - Support AJAX Helpers
- **jQuery**
  - DOM Manipulation
  - Plugins for any feature
  - jQuery UI for widgets