



.NET MVC

UNIT IV

- Jalpa Poriya





HTML HELPER CLASSES

HTML helpers are the way by which we can render html on the view page of MVC.

These Helpers are simple functions that let the developer specify the type of HTML needed on the view.

HTML Helper can be **loosely typed view** and **strongly typed view**.

HtmlHelper - TextBox and TextBoxFor

HtmlHelper class includes two extension methods which creates a textbox (`<input type="text">`) element in razor view: `TextBox()` and `TextBoxFor()`.

The `TextBox()` method is loosely typed method whereas `TextBoxFor()` is a strongly typed method.

SYNTAX



```
Html.TextBox(string name, string value, object htmlAttributes)
```

```
Html.TextBoxFor(lambda expression, object htmlAttributes)
```

OTHER CONTROLS OF HELPER

- `TextArea(string name, string value, object htmlAttributes)`
- `TextAreaFor(lambda expression, object htmlAttributes)`
- `RadioButton(string name, object value, bool isChecked, object htmlAttributes)`
- `RadioButtonFor(expression, object value, object htmlAttributes)`
- `CheckBox(string name, bool isChecked, object htmlAttributes)`
- `CheckBoxFor(lambda expression, object htmlAttributes)`
- `DropDownList(string name, IEnumerable<SelectListItem> selectList, string optionLabel, object htmlAttributes)`
- `DropDownListFor(expression, IEnumerable<SelectListItem> selectList, string optionLabel, object htmlAttributes)`

Other Controls of Helper

- `Hidden(string name, object value, object htmlAttributes)`
- `HiddenFor(Expression)`
- `Password(string name, object value, object htmlAttributes)`
- `PasswordFor(Expression, object htmlAttributes)`
- `Display(string expression)`
- `DisplayFor(expression)`
- `Label(string expression, string labelText, object htmlAttributes)`
- `LabelFor(expression)`



EDITOR

- We have seen different HtmlHelper methods used to generate different html elements in the previous sections.
- ASP.NET MVC also includes a method that generates html input elements based on the datatype.
- EditorFor() extension method generates html elements based on the data type of the model object's property.



Property Data Type

string

Int

decimal, float

Boolean

Enum

DateTime

Html Element

```
<input type="text" >
```

```
<input type="number" >
```

```
<input type="text" >
```

```
<input type="checkbox" >
```

```
<input type="text" >
```

```
<input type="datetime" >
```


AJAX with
MVC

What is AJAX?

AJAX stands for Asynchronous JavaScript And XML that enable web applications to **retrieve data** from the server **asynchronously**.

Web application using **AJAX enables partial page updates**, only the related section of the page is updated, without reloading the entire page.

Advantages of AJAX

1. AJAX applications are non blocking.

As AJAX requests are asynchronous, the user doesn't have to wait for the request processing to complete.

Even while the request is still being processed by the server, the application remains responsive and the user can interact with the application.

When the request processing is complete, the user interface is automatically updated.

This is not the case with, synchronous requests. The user interface is blocked and the user cannot do anything else until the request has completed processing.

2. Better performance and reduced network traffic.

AJAX enables an application to send and receive only the data that is required. As a result, there is reduced traffic between the client and the server and better performance.

3. No screen flicker.

An AJAX response consists of only the data that is relevant to the request. As a result, only a portion of the page is updated avoiding full page refresh and screen flickers.

Disadvantages of AJAX:

1. AJAX requests cannot be bookmarked easily
2. AJAX relies on JavaScript. If JavaScript is disabled, AJAX application won't work.
3. Harder to debug
4. Search Engine like Google, Bing, Yahoo etc cannot index AJAX pages.

Application using AJAX

1. Many web sites like Google, bing, youtube, yahoo use AJAX to implement AutoComplete feature.
2. Gmail use AJAX to implement AutoSave feature
3. Gmail use AJAX to implement RemoteValidation i.e to validate if a username is already in use, when creating a gmail account.
4. Facebook use AJAX, to load data as you keep scrolling down.

Ajax File with JQuery

```
<script src="~/Scripts/jquery-1.7.1.min.js"  
type="text/javascript"></script>
```

```
<script src="~/Scripts/jquery.unobtrusive-ajax.min.js"  
type="text/javascript"></script>
```

AJAX in Employee Project

[ALL](#) | [Male](#) | [Female](#)

Nmae	Gender	Age	Select Photo
John	Male	23	~/Photos/Emp1.jpg
Mary	Female	21	~/Photos/Emp2.png
Josh	Male	31	~/Photos/Emp3.jpg

AjaxOption Method with ActionLink

Syntax:

```
@Ajax.ActionLink(LinkTitle, MethodName, new AjaxOptions { HttpMethod,
InsertionMode, UpdateTargetId, OnComplete });
```

Example:

```
@Ajax.ActionLink("All", "All",
    new AjaxOptions
    {
        HttpMethod = "GET", // HttpMethod to use, GET or POST
        UpdateTargetId = "divID", // ID of the HTML element to update
        InsertionMode = InsertionMode.Replace // Replace the existing
contents
```

Example of AJAX with Shared View

Create Application with sql database.

Add "Shared" folder (if it doesn't already exists) in "Views" folder. Right click on "Shared" folder and add a partial view, with name = `_Employee.cshtml`.

Template - List | Model = Employee | Context = EmployeeContext

Right click on the "Controllers" folder and add a controller with the following settings.

Controller Name = AjaxDemoController | Template = Empty MVC controller



JSON with JQuery

JALPA PORIYA

Adding jQuery to Your Web Pages



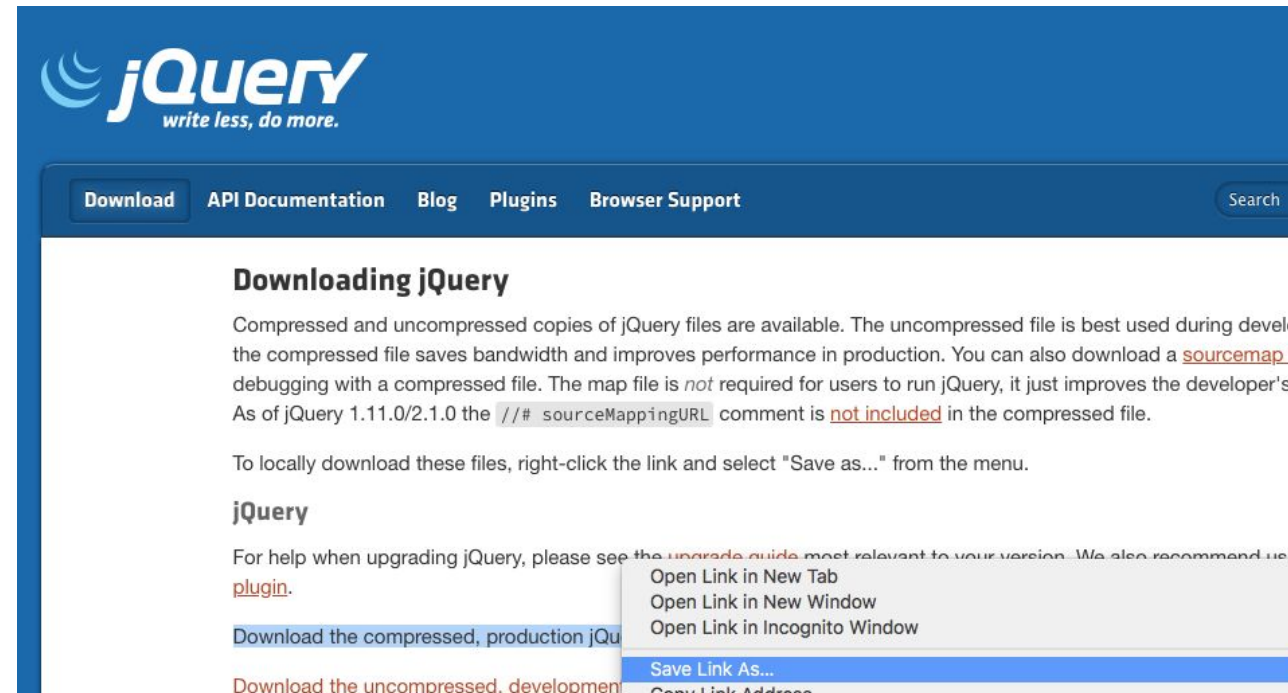
There are several ways to start using jQuery on your web site. You can:

Download the jQuery library from jQuery.com

Include jQuery from a CDN, like Google

Link your web page with js file

```
<head>  
<script src="jquery-3.4.1.min.js"></script>  
</head>
```



jQuery Syntax



Basic syntax is: `$(selector).action()`

A \$ sign to define/access **jQuery**

A (*selector*) to "query (or find)" HTML elements

A jQuery *action*() to be performed on the element(s)

Example: `$("#p").hide()`

The Document Ready Event



```
$(document).ready(function()  
{  
  // code  
});
```

```
$(function()  
{  
  // code  
});
```

jQuery Selector



The element Selector

`$("p").hide()` - hides all `<p>` elements.

The #id Selector

`$(".test").hide()` - hides all elements with `class="test"`.

The .class Selector

`$("#test").hide()` - hides the element with `id="test"`.

More Examples of jQuery Selectors



<code>\$("*")</code>	Selects all elements
<code>\$(this)</code>	Selects the current HTML element
<code>\$("p.intro")</code>	Selects all <code><p></code> elements with <code>class="intro"</code>
<code>\$("p:first")</code>	Selects the first <code><p></code> element
<code>\$("ul li:first")</code>	Selects the first <code></code> element of the first <code></code>
<code>\$(":button")</code>	Selects all <code><button></code> elements and <code><input></code> elements of <code>type="button"</code>
<code>\$("tr:even")</code>	Selects all even <code><tr></code> elements
<code>\$("tr:odd")</code>	Selects all odd <code><tr></code> elements

Events of jQuery



click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

Example:

```
$(“button”).click(function(){  
  // action goes here!!  
});
```

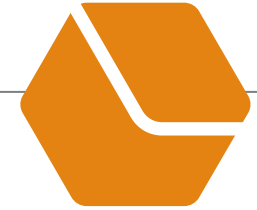
jQuery hide() and show()



```
$("#hide").click(function(){  
  $("#p").hide();  
});
```

```
$("#show").click(function(){  
  $("#p").show();  
});
```

jQuery Fade



```
$("#button").click(function(){  
    $("#div1").fadeIn();  
    $("#div2").fadeOut("slow"); //value  
    $("#div3").fadeToggle(3000); //duration  
    $("#div3").fadeTo("slow", 0.7); //with opacity  
});
```

jQuery Slide



```
$("#flip").click(function(){  
    $("#panel").slideDown();  
    $("#panel").slideUp();  
    $("#panel").slideToggle();  
});
```

jQuery animate



```
$("#button").click(function(){  
    $("#div").animate({  
        left: '250px', opacity: '0.5', height: '150px', width: '150px'  
    });  
});
```

jQuery - css() Method

Get value of css



To return the value of a specified CSS property, use the following syntax:

```
css("propertyname");
```

The following example will return the background-color value of the FIRST matched element: Example

```
$("#p").css("background-color");
```

Set a CSS Property



To set a specified CSS property, use the following syntax:

```
css("propertyname","value");
```

The following example will set the background-color value for ALL matched elements: Example

```
$("#p").css("background-color", "yellow");
```

jQuery - Add Elements



jQuery methods that are used to add new content:

1. `append()` - Inserts content at the end of the selected elements
2. `prepend()` - Inserts content at the beginning of the selected elements
3. `after()` - Inserts content after the selected elements
4. `before()` - Inserts content before the selected elements



JSON WITH MVC

Results in Action Method

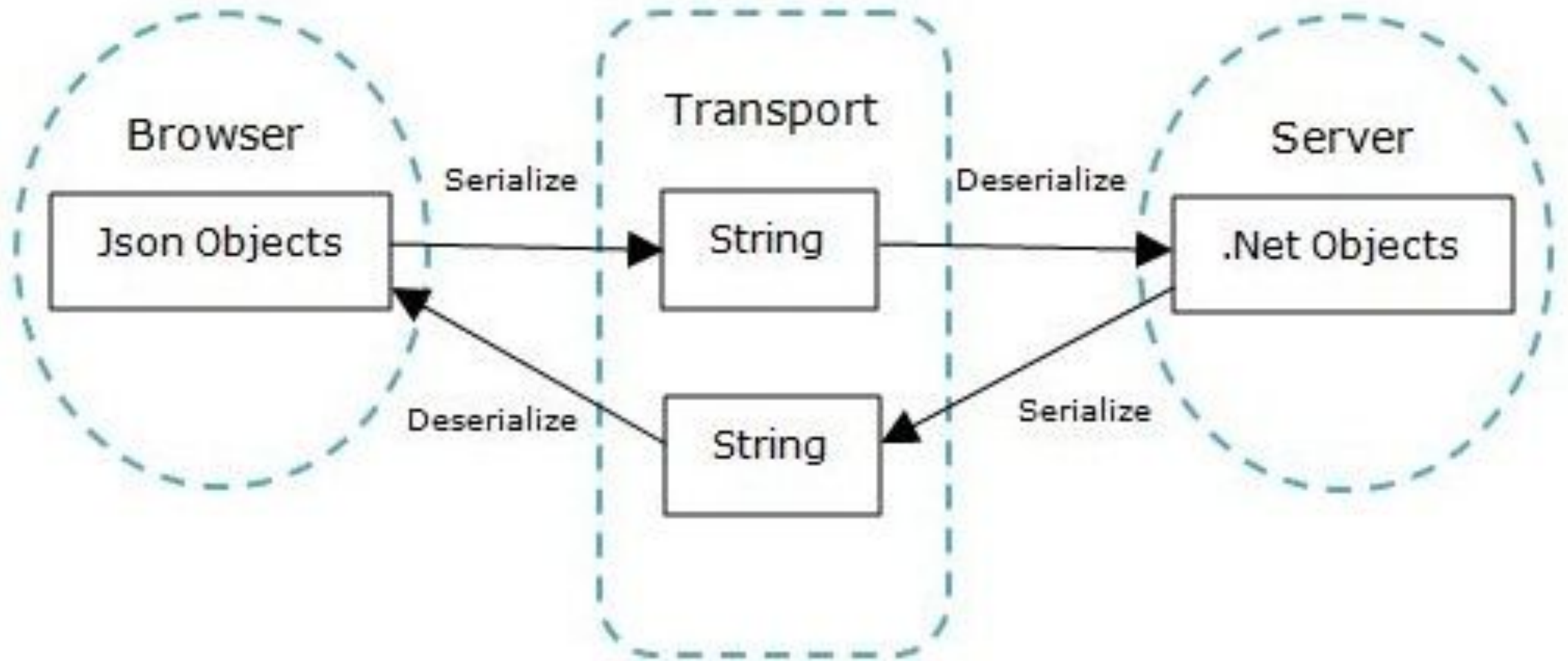
ActionResult BaseClass

- Derived Classes:
- ContentResult - Content()
- FileResult - File()
- HttpNotFoundResult - HttpNotFound()
- **JsonResult - Json()**
- **It represent a base class that returns JSON formatted content to response.**

What is JSON?

- JSON *Java Script Object Notation* is a very familiar and commonly used concept.
- It is a **data interchange** medium and is **very lightweight**.
- It is one kind of a **syntax for storing and passing data**.
- Since it is Java script object notation, it uses the java script style of syntax, but actually is **text only**.
- It also is language **independent**.

Serialization and Deserialization (with .NET)



At the browser side, the data is stored and manipulated as "JavaScript" "JSON objects". At the server side, if ASP.NET MVC is used, the data is stored and manipulated as ".NET objects".

1. When the browser loads data from the server, the .NET objects need to be "serialized" into "JSON strings" and passed to the browser. The browser will "de-serialize" the "JSON strings" into easy to use JavaScript "JSON objects".
2. When the browser sends data to the server, the "JSON objects" need to be "serialized" into "JSON strings" and then "de-serialized" into ".NET objects" at the server side.

JSON syntax

```
{  
  "name": "JOHN",  
  "age": 18,  
  "streetAddress": "100 Internet Dr",  
  "city": "JavaTown",  
  "phoneNumbers": [  
    { "Mobile": "111-111-1111" },  
    { "Home": "222-222-2222" }  
  ]  
}
```

JSON in MVC

Controller:

```
public JsonResult ReadJSON()
{
    return Json(db.students.ToList(),
    JsonRequestBehavior.AllowGet);
}
```

Output:

```
[{"ID":2,"EnrollmentNo":"IU1002","Name":"Pooja","Gender":1}, {"ID":4,"EnrollmentNo":"iu1`0002","Name":"anc","Gender":0}, {"ID":5,"EnrollmentNo":"iu14541","Name":"ABCE","Gender":0}, {"ID":6,"EnrollmentNo":"IU1004","Name":"Tina","Gender":1}, {"ID":7,"EnrollmentNo":"IU188383001","Name":"Kajal","Gender":1}, {"ID":8,"EnrollmentNo":"IU188383002","Name":"Hiren","Gender":0}, {"ID":9,"EnrollmentNo":"IU188282001","Name":"Priyanshu","Gender":0}, {"ID":10,"EnrollmentNo":"IU188282002","Name":"Kajal Oza","Gender":1}, {"ID":11,"EnrollmentNo":"IU178383001","Name":"Keyur ","Gender":0}, {"ID":12,"EnrollmentNo":"IU178383002","Name":"Tina Patel","Gender":1}, {"ID":13,"EnrollmentNo":"IU168383012","Name":"Tina Tukadiya","Gender":1}, {"ID":14,"EnrollmentNo":"IU168383011","Name":"Tina Tukadiya","Gender":1}]
```

REFERENCES

Web Site:

<https://docs.microsoft.com/en-us/aspnet/mvc/>

https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

<https://www.guru99.com/mvc-tutorial.html>

<https://en.wikipedia.org/wiki/ModelViewController>

<https://www.guru99.com/mvc-tutorial.html>

<https://www.geeksforgeeks.org/mvc-design-pattern/>

Book:

Pro ASP.NET MVC 5.0