

Agregar CheckBoxes con función JQuery y JSON « afelipelc Blog

Continuando con el proyecto Películas (ASP.NET MVC 2), la tabla película tiene un campo llamado Disponible (Byte = true o false), en nuestro proyecto en lugar de mostrar True o False, anteriormente agregamos el código que muestra Si o No dependiendo del valor del campo, lo que haremos ahora es: En el módulo de administración, convertir la columna Disponible en una columna con CheckBoxes donde el administrador podrá modificar el estado de cada película, el resultado será similar a:

Catalogo de todas las peliculas en la coleccion.

| | ID | Titulo | Genero | Fecha de Lanzamiento | Precio | Disponible |
|---|----|--------------------------|------------|----------------------|--------|-------------------------------------|
| Editar Eliminar | 6 | El caballero de la noche | Drama | 25/05/2008 | 90.00 | <input checked="" type="checkbox"/> |
| Editar Eliminar | 7 | Super Man | Accion | 13/04/2009 | 85.00 | <input checked="" type="checkbox"/> |
| Editar Eliminar | 74 | Como Perros y Gatos II | Accion | 25/05/2009 | 90.00 | <input checked="" type="checkbox"/> |
| Editar Eliminar | 76 | This is not a Movie | Documental | 25/09/2010 | 120.00 | <input checked="" type="checkbox"/> |
| Editar Eliminar | 80 | Y Tu Cuanto Cuestas | Documental | 01/05/2007 | 90.00 | <input checked="" type="checkbox"/> |
| Editar Eliminar | 81 | El plan perfecto | Accion | 24/07/2006 | 80.00 | <input checked="" type="checkbox"/> |
| Editar Eliminar | 82 | Nuevo Titulo 1810 | Accion | 16/05/2010 | 80.00 | <input type="checkbox"/> |
| Editar Eliminar | 83 | Prueba nueva | Accion | 18/10/2010 | 80.00 | <input type="checkbox"/> |

Img 1.- Resultado con CheckBoxes

¿Qué necesitamos y como lo haremos?

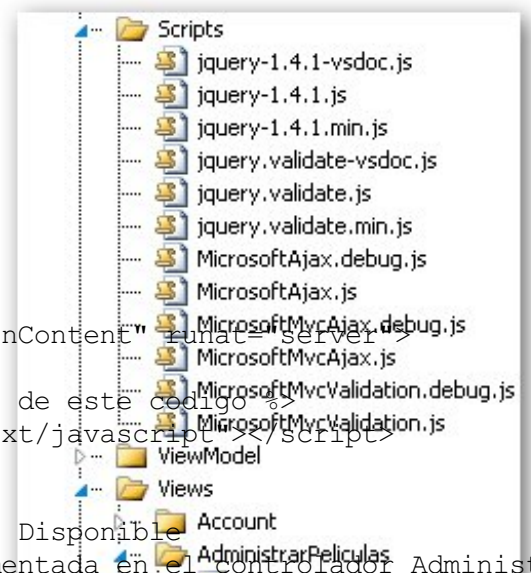
Para realizarlo lo hacemos por medio de **JQuery** (<http://jquery.com>) que es una Biblioteca o Framework de JavaScript que permite simplificar la manera de interactuar con documentos HTML utilizando la tecnología **AJAX** (acrónimo de Asynchronous JavaScript And XML “*JavaScript asíncrono y XML*”) para manipular los DOM (Document Object Model “*modelo en objetos para la representación de documentos*”).

Entonces al grano, la librería JQuery se encuentra ya integrada en nuestro proyecto (en el explorador de soluciones / Carpeta Scripts / jquery-1.4.1.js).

Entonces en la vista donde se va a utilizar JQuery, agregamos el código que incluirá el archivo JS y la función que realizara la llamada a la acción que actualizara el estado Disponible, el resultado de la consulta es devuelto en formato JSON (<http://api.jquery.com/jquery.getJSON>), en nuestro caso es en la vista Index.aspx del controlador AdministrarPeliculas.

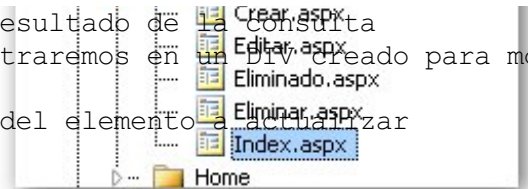
Agregamos el código dentro del bloque MainContent en la vista Index.aspx que deberá quedar así:

```
<% //Esta línea de código ya existe. %>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">
<% //Esta línea es la que se debe agregar y el resto de este código %>
<script src="../../../Scripts/jquery-1.4.1.js" type="text/javascript"></script>
<script type="text/javascript">
//Esta función ejecutara la actualización del estado Disponible
//la función llama a la acción DisponibleSiNo implementada en el controlador AdministrarPeliculas
</script>
</asp:Content>
```



```
//como [ActionName("DisponibleSiNo)], devuelve el resultado de la consulta
//en formato JSON (www.json.org) el resultado lo mostraremos en un DIV creado para nosotros

//la función JavaScript recibe como parámetro el ID del elemento a actualizar
function Disponible(Id) {
    //Generamos la consulta JSON
    //y procesamos los resultados devueltos con function(data) { //Procesar resultados
        $.getJSON("<%= Url.Action("DisponibleSiNo") %>"
            + "/" + Id,
            function(data) {
                if(data.Id != 0){
                    $("#Id"+Id).empty();
                    $("#Id"+Id).append(data.Disponible);
                }else alert("No se encontro el registro: "+Id);
            });
    }
}
</script>
```



Ahora en nuestra vista donde se muestra Si o No dependiendo del valor del campo Disponible, modificaremos el código:

```
<% try
    {
        if ((bool) (item.Disponible))
        { %>
            <% //Aquí posteriormente en vez de mostrar SI, mostraremos
            //a una acción que actualice el estado de la película %>
            Si
            <%
        }
        else
        { %>
            No
            <% }
        }
        catch
        {
            %>
            No <% //Si Disponible es NULL, se producirá un error, entonces lo
        <%
        } %>
    }
</try>
```

Cambiaremos Si y No por un CheckBox con el evento Clic que llamada a la función Disponible de nuestro JavaScript, delante del CheckBox agregaremos un DIV donde se mostrara el resultado como notificación

El código modificado quedara así:

```
<% try
    {
        if ((bool) (item.Disponible))
        { %>
            <% //como el estado es TRUE => SI, el checkbox debe estar
            <input type="checkbox" checked="checked" onClick="javascri
            <%
        }
        else
        { %>
            <% // Si es False, colocamos el checkbox NO Activado %>
            <input type="checkbox" onClick="javascript: Disponible('<
            <% }
        }
        catch
        {
            %>
        }
    }
</try>
```

```

<% //Si Disponible es NULL, se producira un error, entonces lo toma
<input type="checkbox" onClick="javascript: Disponible('<%= item.I
<%
} %>

<%//Ahora agregamos el DIV delante del CheckBox
//Donde el ID = Id#IdPelicula %>
<div id="Id<%= item.PeliculaId %>" ></div>

```

Hasta aquí ya se encuentra la vista. Si ejecutamos nuestro proyecto, AdministrarPeliculas se verá así:

| | ID | Titulo | Fecha de Lanzamiento | Genero | Precio | Disponible |
|---|----|-----------------------|----------------------|---------|----------|-------------------------------------|
| Editar Eliminar | 1 | Identidad Sustituta | 25/09/2009 | Ficcion | \$ 80.00 | <input checked="" type="checkbox"/> |
| Editar Eliminar | 3 | Dejavu | 22/02/2004 | Accion | \$ 70.00 | <input checked="" type="checkbox"/> |
| Editar Eliminar | 5 | El principe de Persia | 12/05/2010 | Ficcion | \$ 87.00 | <input type="checkbox"/> |
| Editar Eliminar | 6 | Y tu cuanto cuestas? | 01/05/2007 | Accion | \$ 90.00 | <input type="checkbox"/> |
| Editar Eliminar | 7 | El Plan Perfecto | 28/08/2006 | Accion | \$ 90.00 | <input checked="" type="checkbox"/> |

Img 3.- Vista previa con los cambios realizados.

Si activamos o desactivamos los checkbox, no se producirá ningún efecto sobre los datos, ya que la acción **DisponibleSiNo** en el controlador aún no se ha implementado.

La acción DisponibleSiNo en el controlador AdministrarPeliculas será la siguiente:

```

//Respuesta a la acción invocada por AJAX
//La acción devuelve un resultado JSON
[ActionName("DisponibleSiNo")]
public JsonResult Disponible(int id)
{
    //Seleccionamos la Pelicula a actualizar (Disponible)
    var pelicula = DB.Pelicula.Single(p => p.PeliculaId == id);

    //Creamos una variable para almacenar SI o NO dependiendo del estado Dispc
    //Para devolverlo como resultado
    string disponibletxt;

    //comprobamos el valor del campo Disponible para cambiarlo
    if (pelicula.Disponible == true)
    {
        pelicula.Disponible = false;
        disponibletxt = "No";
    }
    else
    {
        pelicula.Disponible = true;
        disponibletxt = "Si";
    }

    //Actualizamos el modelo
    UpdateModel(pelicula, "Pelicula");
}

```

```

//Guardamos los cambios realizados
DB.SaveChanges();

//Creamos un resultado JSON con los datos necesarios :D, indicando que
//El cliente (navegador) pueda recuperar esos datos
//Revisar http://api.jquery.com/jquery.getJSON para comprender JSON
return this.Json(new { Id = pelicula.PeliculaId, Disponible = disponible});
}

```

Y probando la funcionalidad de nuestro proyecto, verificamos que funcione correctamente

Cuando desactivemos algún checkbox, delante se mostrara No, cuando activemos algún otro, delante se mostrara Si.

| | ID | Título | Fecha de Lanzamiento | Genero | Precio | Disponible |
|---|----|-----------------------|----------------------|---------|----------|--|
| Editar Eliminar | 1 | Identidad Sustituta | 25/09/2009 | Ficcion | \$ 80.00 | <input checked="" type="checkbox"/> |
| Editar Eliminar | 3 | Dejavu | 22/02/2004 | Accion | \$ 70.00 | <input type="checkbox"/> No |
| Editar Eliminar | 5 | El principe de Persia | 12/05/2010 | Ficcion | \$ 87.00 | <input checked="" type="checkbox"/> Si |
| Editar Eliminar | 6 | Y tu cuanto cuestas? | 01/05/2007 | Accion | \$ 90.00 | <input type="checkbox"/> |
| Editar Eliminar | 7 | El Plan Perfecto | 28/08/2006 | Accion | \$ 90.00 | <input checked="" type="checkbox"/> |

[Registrar nueva pelicula](#)

Img 4.-Resultado final

Aunque no veamos como el navegador realiza la llamada, los cambios ya se han realizado, esa es la magia de AJAX, para comprobarlo, prestamos atención a los registros que modificamos (Disponible) y actualizamos la página



Pues bien, hasta aquí ha sido todo lo que respecta a los checkboxes.

Este ejemplo puede implementarse para rellenar un formulario cargando los datos con una función similar con resultados JSON ya sea al presionar alguna tecla o cuando se produzca algún otro evento como clic, etc. Para implementarlo revisa la documentación de la función Bind() de JQuery en

<http://api.jquery.com/bind/>