

Ligar Datos en Model Dinámicamente

Si tenemos un form al que constantemente agregamos nuevos campos y valores que necesitan ser validados con el sistema que provee MVC, entonces necesitaremos hacer un procedimiento un poco mas complicado para lograr nuestro objetivo.

1.- Requisitos Practica

```
namespace ListasDinamicas.Models
{
    public class Movie
    {
        [Required]
        public string Title { get; set; }
        public int Rating { get; set; }
    }
}
```

```
namespace ListasDinamicas.Models
{
    public class User
    {
        public int? Id { get; set; }
        [Required]
        public string Name { get; set; }
        public List<Movie> FavouriteMovies { get; set; }
    }
}
```

Models

- C# Movie.cs
- C# User.cs

- Dos clases Model, Uno será un Usuario con su nombre y una lista de sus películas favoritas.
- Películas lo único que tendrá es su titulo y su calificación

2.- Instalar Nuget

- Instalamos en Nuget el paquete BeginCollectionItem

Escriba 'get-help NuGet' para ver todos los comandos de NuGet disponibles.

```
PM> Install-Package BeginCollectionItem -Version 1.2.1
```

3.- Análisis de Lista

- Analizaremos la Lista que mandamos a nuestra vista
- A) Si es null podremos un HTML pre terminado.
- B) Por cada miembro en nuestra lista haremos un renderizado de una vista parcial a la que le mandaremos el modelo presente.

```
<fieldset>
  <legend>My Favourite Movies</legend>
  @if (Model.FavouriteMovies == null || Model.FavouriteMovies.Count == 0)
  {
    <p>None.</p>
  }
  else
  {
    <ul id="movieEditor" style="list-style-type: none">
      @foreach (Movie movie in Model.FavouriteMovies)
      {
        Html.RenderPartial("MovieEntryEditor", movie);
      }
    </ul>
    <p> <a id="addAnother" href="#">Add another</a></p>
  }
</fieldset>
```


4.- Partial View

```
@model ListasDinamicas.Models.Movie

<li style="padding-bottom:15px">
  @using (Html.BeginCollectionItem("FavouriteMovies"))
  {
    @Html.LabelFor(model => model.Title)
    @Html.EditorFor(model => model.Title)
    @Html.ValidationMessageFor(model => model.Title)

    @Html.LabelFor(model => model.Rating)
    @Html.EditorFor(model => model.Rating)
    @Html.ValidationMessageFor(model => model.Rating)

    <a href="#" onclick="$(this).parent().remove();">Delete</a>
  }
</li>

<script type="text/javascript">
  $('form').removeData('validator');
  $('form').removeData('unobtrusiveValidation');
  $.validator.unobtrusive.parse('form');
</script>
```

Tres Cosas importantes:

1.- Modelo referenciado.

2.- Nuestros campos dentro de @Using BeginCollectionItem

3.- Volver a validar el JQuery validador

5.- Javascript

```
inamicas Archivos de contenido JavaScript <global>
$(document).ready(function () {
    $("#addAnother").click(function () {
        $.post('/Home/MovieEntryRow', function (template) {
            $("#movieEditor").append(template);
        });
    });
});
```

```
public ActionResult MovieEntryRow()
{
    return PartialView("MovieEntryEditor");
}
```

- Añadir un nuevo Row al Form ocupamos JavaScript y Una acción en el server