



EBook Gratis

APRENDIZAJE

asp.net-mvc-5

Free unaffiliated eBook created from
Stack Overflow contributors.

#asp.net-
mvc-5

Tabla de contenido

Acerca de	1
Capítulo 1: Empezando con asp.net-mvc-5	2
Observaciones.....	2
Examples.....	2
Lo nuevo en ASP.NET MVC 5.....	2
Instalar MVC5 o actualizar a una versión específica.....	2
Capítulo 2: Atributo de enrutamiento en mvc-5	4
Sintaxis.....	4
Observaciones.....	4
Examples.....	4
Cómo implementar ruta de atributos.....	4
Parámetros URI opcionales y valores predeterminados.....	4
Prefijos de ruta.....	5
Ruta por defecto.....	6
Restricciones de ruta.....	6
Capítulo 3: Controlador asíncrono en MVC 5	8
Examples.....	8
La definición.....	8
Controlador asincrono.....	8
Capítulo 4: Crear ayudantes HTML	9
Introducción.....	9
Observaciones.....	9
Examples.....	9
Crea un ayudante simple - un div con un texto en él.....	9
Ayudante desechable (como Html.BeginForm).....	9
Creditos	11

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [asp-net-mvc-5](#)

It is an unofficial and free asp.net-mvc-5 ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official asp.net-mvc-5.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con asp.net-mvc-5

Observaciones

Esta sección proporciona una descripción general de qué es asp.net-mvc-5 y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de asp.net-mvc-5, y vincular a los temas relacionados. Dado que la Documentación para asp.net-mvc-5 es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Lo nuevo en ASP.NET MVC 5

1. Filtros de autenticación Son un nuevo tipo de filtro agregado en ASP.NET MVC 5.0. Se ejecutan antes de los filtros de autorización en el canal de ASP.NET MVC y le permiten especificar la lógica de autenticación por acción, por controlador o globalmente para todos los controladores. Los filtros de autenticación procesan las credenciales en la solicitud y proporcionan el principal correspondiente. Los filtros de autenticación también pueden agregar desafíos de autenticación en respuesta a solicitudes no autorizadas.
2. Anulaciones de filtro Ahora puede anular los filtros que se aplican a un método de acción o controlador determinado especificando un filtro de anulación.
3. Atributo de enrutamiento

Instalar MVC5 o actualizar a una versión específica

Para instalar / actualizar la versión MVC, siga estos pasos:

1. En Visual Studio, abra la consola del Administrador de paquetes (use CTRL + Q y escriba la consola del Administrador de paquetes)
2. En la consola que aparece, ingrese lo siguiente después del cursor de la consola que muestra `PM>` :

```
Install-Package Microsoft.AspNet.Mvc -Version 5.2.3
```

Nota: especifique la versión que desea. En el ejemplo anterior usamos 5.2.3 (la última versión cuando se escribieron estas instrucciones)

3. Verifique la instalación utilizando el siguiente comando en la consola del administrador de paquetes:

```
Get-Package -ListAvailable -Filter mvc
```

Lea Empezando con asp.net-mvc-5 en línea: <https://riptutorial.com/es/asp-net-mvc-5/topic/976/empezando-con-asp-net-mvc-5>

Capítulo 2: Atributo de enrutamiento en mvc-5

Sintaxis

1. {productId: int} / {productTitle} asignado a ProductsController.Show (int id)
2. {nombre de usuario} asignado a ProfilesController.Show (nombre de usuario de cadena)
3. {nombre de usuario} / catalogs / {catalogId: int} / {catalogTitle} asignado a CatalogsController.Show (cadena nombre de usuario, int catalogId)

Observaciones

El enrutamiento es cómo ASP.NET MVC hace coincidir un URI con una acción. MVC 5 admite un nuevo tipo de enrutamiento, llamado enrutamiento de atributos. Como su nombre lo indica, el enrutamiento de atributos utiliza atributos para definir rutas. El enrutamiento de atributos le brinda más control sobre los URI en su aplicación web.

El estilo anterior de enrutamiento, denominado enrutamiento basado en convenciones, todavía es totalmente compatible. De hecho, puedes combinar ambas técnicas en el mismo proyecto.

Examples

Cómo implementar ruta de atributos

Habilitación del enrutamiento de atributos Para habilitar el enrutamiento de atributos, llame a `MapMvcAttributeRoutes` durante la configuración.

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

    routes.MapMvcAttributeRoutes();

    routes.MapRoute(
        name: "Default",
        url: "{controller}/{action}/{id}",
        defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
    );
}
```

Parámetros URI opcionales y valores predeterminados

Puede hacer que un parámetro URI sea opcional agregando un signo de interrogación al parámetro de ruta. También puede especificar un valor predeterminado utilizando el parámetro

parámetro = valor.

```
public class BooksController : Controller
{
    // eg: /books
    // eg: /books/1430210079
    [Route("books/{isbn?}")]
    public ActionResult View(string isbn)
    {
        if (!String.IsNullOrEmpty(isbn))
        {
            return View("OneBook", GetBook(isbn));
        }
        return View("AllBooks", GetBooks());
    }

    // eg: /books/lang
    // eg: /books/lang/en
    // eg: /books/lang/he
    [Route("books/lang/{lang=en}")]
    public ActionResult ViewByLanguage(string lang)
    {
        return View("OneBook", GetBooksByLanguage(lang));
    }
}
```

En este ejemplo, ambos `/books` y `/books/1430210079` se dirigirán a la acción "Ver", el primero dará como resultado una lista de todos los libros, y el segundo mostrará el libro específico. Ambos `/books/lang` y `/books/lang/en` serán tratados de la misma manera.

Prefijos de ruta

A menudo, las rutas en un controlador comienzan todas con el mismo prefijo. Por ejemplo:

```
public class ReviewsController : Controller
{
    // eg: /reviews
    [Route("reviews")]
    public ActionResult Index() { ... }
    // eg: /reviews/5
    [Route("reviews/{reviewId}")]
    public ActionResult Show(int reviewId) { ... }
    // eg: /reviews/5/edit
    [Route("reviews/{reviewId}/edit")]
    public ActionResult Edit(int reviewId) { ... }
}
```

Puede establecer un prefijo común para un controlador completo utilizando el atributo `[RoutePrefix]`:

```
[RoutePrefix("reviews")]
public class ReviewsController : Controller
{
    // eg.: /reviews
    [Route]
    public ActionResult Index() { ... }
    // eg.: /reviews/5
```

```

[Route("{reviewId}")]
public ActionResult Show(int reviewId) { ... }
// eg.: /reviews/5/edit
[Route("{reviewId}/edit")]
public ActionResult Edit(int reviewId) { ... }
}

```

Use una tilde (~) en el atributo del método para anular el prefijo de ruta si es necesario:

```

[RoutePrefix("reviews")]
public class ReviewsController : Controller
{
    // eg.: /spotlight-review
    [Route("~/spotlight-review")]
    public ActionResult ShowSpotlight() { ... }

    ...
}

```

Ruta por defecto

También puede aplicar el atributo [Ruta] en el nivel del controlador, capturando la acción como un parámetro. Esa ruta se aplicaría a todas las acciones en el controlador, a menos que se haya definido una [Ruta] específica en una acción específica, anulando el conjunto predeterminado en el controlador.

```

[RoutePrefix("promotions")]
[Route("{action=index}")]
public class ReviewsController : Controller
{
    // eg.: /promotions
    public ActionResult Index() { ... }

    // eg.: /promotions/archive
    public ActionResult Archive() { ... }

    // eg.: /promotions/new
    public ActionResult New() { ... }

    // eg.: /promotions/edit/5
    [Route("edit/{promoId:int}")]
    public ActionResult Edit(int promoId) { ... }
}

```

Restricciones de ruta

Las restricciones de ruta le permiten restringir cómo se comparan los parámetros en la plantilla de ruta. La sintaxis general es {parámetro: restricción}. Por ejemplo:

```

// eg: /users/5
[Route("users/{id:int}")]
public ActionResult GetUserById(int id) { ... }

// eg: users/ken

```



```
[Route("users/{name}")]
public ActionResult GetUserByName(string name) { ... }
```

Aquí, la primera ruta solo se seleccionará si el segmento "id" de la URI es un número entero. De lo contrario, se elegirá la segunda ruta.

Const	Descripción (coincidencias :)	Ejemplo
alfa	Caracteres del alfabeto latino en mayúsculas o minúsculas (az, AZ)	{x: alfa}
bool	Valor booleano.	{x: bool}
fecha y hora	Valor de fecha y hora.	{x: datetime}
decimal	Valor decimal.	{x: decimal}
doble	Valor de punto flotante de 64 bits.	{x: doble}
flotador	Valor de punto flotante de 32 bits.	{x: flotar}
guid	Valor GUID.	{x: guid}
Ent	Valor entero de 32 bits.	{x: int}
longitud	Cadena con la longitud especificada o dentro de un rango especificado de longitudes.	{x: longitud (6)} {x: longitud (1,20)}
largo	Valor entero de 64 bits.	{x: largo}
max	Entero con un valor máximo.	{x: max (10)}
longitud máxima	Cadena con una longitud máxima.	{x: longitud máxima (10)}
min	Entero con un valor mínimo.	{x: min (10)}
longitud mínima	Cadena con una longitud mínima.	{x: longitud mínima (10)}
distancia	Entero dentro de un rango de valores.	{x: rango (10,50)}
expresiones regulares	Expresión regular.	{x: regex (^ \ d {3} - \ d {3} - \ d {4} \$)}

Lea Atributo de enrutamiento en mvc-5 en línea: <https://riptutorial.com/es/asp-net-mvc-5/topic/6370/atributo-de-enrutamiento-en-mvc-5>

Capítulo 3: Controlador asíncrono en MVC 5

Examples

La definicion

Uso de un controlador asíncrono en ASP.NET MVC. La clase AsyncController le permite escribir métodos de acción asíncronos. Puede usar métodos de acción asíncrona para solicitudes de larga duración que no estén vinculadas a la CPU. Esto evita que el servidor web no realice el trabajo mientras se procesa la solicitud.

Controlador asincrono

```
public async Task<ActionResult> Index()
{
    return View("View", await db.UserMasers.ToListAsync());
}
```

Lea Controlador asíncrono en MVC 5 en línea: <https://riptutorial.com/es/asp-net-mvc-5/topic/7500/controlador-asincrono-en-mvc-5>

Capítulo 4: Crear ayudantes HTML

Introducción

Los ayudantes HTML son una forma muy útil de crear elementos HTML en vistas utilizando el marco MVC. Con un poco de tiempo, su equipo puede beneficiarse realmente de su uso. Ayuda a mantener el código limpio y propenso a errores.

Observaciones

Para usar los ayudantes, primero debe agregar una directiva `@using` dentro de la vista o agregar el espacio de nombres dentro del archivo `Web.config` ubicado en la carpeta `Views`.

Examples

Creando un ayudante simple - un div con un texto en él

```
public static class MyHelpers
{
    public static MvcHtmlString MyCustomDiv(this HtmlHelper htmlHelper, string text,
        object htmlAttributes = null)
    {
        var mainTag = new TagBuilder("div");
        mainTag.MergeAttributes(htmlAttributes);
        mainTag.AddCssClass("some custom class");
        mainTag.SetInnerHtml(text);
        return MvcHtmlString.Create(mainTag.ToString());
    }
}
```

Para usarlo en las vistas:

```
@Html.MyCustomDiv("Test inside custom div");
@Html.MyCustomDiv("Test inside custom div", new { @class="some class for the div element" });
```

Ayudante desechable (como `Html.BeginForm`)

1. Primero crea una clase desechable:

```
public class MyDisposableHelper: IDisposable
{
    private bool _disposed;
    private readonly ViewContext _viewContext;

    public MyDisposableHelper(ViewContext viewContext)
    {
        if (viewContext == null)
        {
```

```

        throw new ArgumentNullException(nameof(viewContext));
    }
    _viewContext = viewContext;
}

public void Dispose()
{
    Dispose(true);
    GC.SuppressFinalize(this);
}

protected virtual void Dispose(bool disposing)
{
    if (_disposed)
        return;
    _disposed = true;
    _viewContext.Writer.Write("</div>");
}

public void EndForm()
{
    Dispose(true);
}
}

```

Esta clase hereda `IDisposable` porque queremos usar el helper como `Html.BeginForm(...)`. Cuando se elimina, cierra el `div` creado cuando llamamos al ayudante en la vista.

2. Crea el método de extensión `HtmlHelper` :

```

public static MyDisposableHelper BeginContainingHelper(this HtmlHelper htmlHelper)
{
    var containingTag = new TagBuilder("div");
    //add default css classes, attributes as needed
    htmlHelper.ViewContext.Writer.Write(containingTag.ToString(TagRenderMode.StartTag));
    return new MyDisposableHelper (htmlHelper.ViewContext);
}

```

Lo que debe notar aquí es la llamada a `Writer.Write` para escribir en la página de respuesta un elemento personalizado. `TagRenderMode.StartTag` se usa para informar al escritor que no cierre el `div` todavía, porque lo vamos a cerrar cuando `MyDisposableHelper` clase `MyDisposableHelper`.

3. Para usarlo en la vista:

```

@using (Html.BeginContainingHelper()) {
    <div>element inside our custom element</div>
}

```

Lea Crear ayudantes HTML en línea: <https://riptutorial.com/es/asp-net-mvc-5/topic/8922/crear-ayudantes-html>

Creditos

S. No	Capítulos	Contributors
1	Empezando con asp.net-mvc-5	Community , Hamzaway , hasan , Meghraj , Rosdi Kasim , Twister1002
2	Atributo de enrutamiento en mvc-5	Anik Saha , PedroSouki
3	Controlador asíncrono en MVC 5	vicky
4	Crear ayudantes HTML	Mihail Stancescu