



eBook Gratuit

APPRENEZ

asp.net-mvc-5

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#asp.net-

mvc-5

Table des matières

À propos.....	1
Chapitre 1: Démarrer avec asp.net-mvc-5.....	2
Remarques.....	2
Exemples.....	2
Quoi de neuf dans ASP.NET MVC 5.....	2
Installer MVC5 ou mettre à jour vers une version spécifique.....	2
Chapitre 2: Contrôleur asynchrone dans MVC 5.....	4
Exemples.....	4
Définition.....	4
Contrôleur asynchrone.....	4
Chapitre 3: Créer des aides HTML.....	5
Introduction.....	5
Remarques.....	5
Exemples.....	5
Créer un assistant simple - une div avec un texte.....	5
Helper jetable (comme Html.BeginForm).....	5
Chapitre 4: Routage d'attribut dans mvc-5.....	7
Syntaxe.....	7
Remarques.....	7
Exemples.....	7
Comment implémenter la route d'attribut.....	7
Paramètres d'URI facultatifs et valeurs par défaut.....	7
Préfixes de route.....	8
Route par défaut.....	9
Contraintes de Route.....	9
Crédits.....	11

À propos

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

Chapitre 1: Démarrer avec asp.net-mvc-5

Remarques

Cette section fournit une vue d'ensemble de ce qu'est asp.net-mvc-5 et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets dans asp.net-mvc-5, et établir un lien avec les sujets connexes. La documentation pour asp.net-mvc-5 étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Exemples

Quoi de neuf dans ASP.NET MVC 5

1. Filtres d'authentification Un nouveau type de filtre est ajouté dans ASP.NET MVC 5.0. Il s'exécute avant les filtres d'autorisation dans le pipeline ASP.NET MVC et vous permet de spécifier la logique d'authentification par action, par contrôleur ou globalement pour tous les contrôleurs. L'authentification filtre les informations d'identification du processus dans la demande et fournit un principal correspondant. Les filtres d'authentification peuvent également ajouter des problèmes d'authentification en réponse à des demandes non autorisées.
2. Remplacement des filtres Vous pouvez désormais remplacer les filtres qui s'appliquent à une méthode d'action ou à un contrôleur donné en spécifiant un filtre de remplacement.
3. Routage d'attribut

Installer MVC5 ou mettre à jour vers une version spécifique

Pour installer / mettre à jour la version de MVC, procédez comme suit:

1. Dans Visual Studio, ouvrez la console du gestionnaire de packages (utilisez CTRL + Q et tapez la console du gestionnaire de packages)
2. Dans la console qui apparaît, entrez ce qui suit après le curseur de la console indiquant `PM>` :

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

Remarque: spécifiez la version souhaitée. Dans l'exemple ci-dessus, nous avons utilisé 5.2.3 (la dernière version lorsque ces instructions ont été écrites)

3. Vérifiez l'installation en utilisant la commande suivante dans la console du gestionnaire de packages:

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

Lire Démarrer avec asp.net-mvc-5 en ligne: <https://riptutorial.com/fr/asp-net-mvc-5/topic/976/demarrer-avec-asp-net-mvc-5>

Chapitre 2: Contrôleur asynchrone dans MVC 5

Exemples

Définition

Utilisation d'un contrôleur asynchrone dans ASP.NET MVC. La classe AsyncController vous permet d'écrire des méthodes d'action asynchrones. Vous pouvez utiliser des méthodes d'action asynchrones pour les demandes non liées à une unité centrale et de longue durée. Cela évite de bloquer l'exécution du travail du serveur Web pendant le traitement de la demande.

Contrôleur asynchrone

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

Lire Contrôleur asynchrone dans MVC 5 en ligne: <https://riptutorial.com/fr/asp-net-mvc-5/topic/7500/controleur-asynchrone-dans-mvc-5>

Chapitre 3: Créer des aides HTML

Introduction

Les aides HTML sont un moyen très utile de créer des éléments HTML dans les vues à l'aide du framework MVC. Avec un peu de temps, votre équipe peut vraiment tirer parti de leur utilisation. Cela aide à garder le code propre et sujet aux erreurs.

Remarques

Pour utiliser les helpers, vous devez d'abord ajouter une directive `@using` dans la vue ou ajouter l'espace de noms dans le fichier `Web.config` situé dans le dossier `Views`.

Exemples

Créer un assistant simple - une div avec un texte

```
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());
    }
}
```

Pour l'utiliser dans les vues:

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

Helper jetable (comme `Html.BeginForm`)

1. Commencez par créer une classe jetable:

```
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);
}
}

```

Cette classe hérite de `IDisposable` car nous voulons utiliser l'aide comme `Html.BeginForm(...)`. Lorsqu'il est disposé, il ferme la `div` créée lorsque nous avons appelé l'assistant dans la vue.

2. Créez la méthode d'extension `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);
}

```

Ce qu'il faut noter ici, c'est l'appel à `Writer.Write` pour écrire dans la page de réponse un élément personnalisé. Le `TagRenderMode.StartTag` est utilisé pour informer le rédacteur de ne pas fermer le `div` pour le moment, car nous allons le fermer lors de la `MyDisposableHelper` classe

`MyDisposableHelper`.

3. Pour l'utiliser dans la vue:

```

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

```

Lire Créer des aides HTML en ligne: <https://riptutorial.com/fr/asp-net-mvc-5/topic/8922/creer-des-aides-html>

Chapitre 4: Routage d'attribut dans mvc-5

Syntaxe

1. {productId: int} / {productTitle} Mappé sur ProductsController.Show (int id)
2. {nom d'utilisateur} Mappé sur ProfilesController.Show (nom d'utilisateur de chaîne)
3. {username} / catalogs / {catalogId: int} / {catalogTitle} Mappé sur CatalogsController.Show (nom d'utilisateur de chaîne, int catalogId)

Remarques

Le routage est la manière dont ASP.NET MVC associe un URI à une action. MVC 5 prend en charge un nouveau type de routage, appelé routage d'attribut. Comme son nom l'indique, le routage d'attribut utilise des attributs pour définir des itinéraires. Le routage d'attributs vous permet de mieux contrôler les URI de votre application Web.

Le style de routage antérieur, appelé routage basé sur des conventions, est toujours pris en charge. En fait, vous pouvez combiner les deux techniques dans un même projet.

Exemples

Comment implémenter la route d'attribut

Activation du routage d'attribut Pour activer le routage d'attribut, appelez `MapMvcAttributeRoutes` pendant la configuration.

```
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 }
    );
}
```

Paramètres d'URI facultatifs et valeurs par défaut

Vous pouvez rendre un paramètre d'URI facultatif en ajoutant un point d'interrogation au paramètre de route. Vous pouvez également spécifier une valeur par défaut en utilisant le paramètre de formulaire = valeur.

```

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));
    }
}

```

Dans cet exemple, / books et / books / 1430210079 seront acheminés vers l'action «View», le premier entraînera la liste de tous les livres, et le second listera le livre spécifique. / Books / lang et / books / lang / en seront traités de la même manière.

Préfixes de route

Souvent, les itinéraires dans un contrôleur commencent tous par le même préfixe. Par exemple:

```

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) { ... }
}

```

Vous pouvez définir un préfixe commun pour un contrôleur entier en utilisant l'attribut `[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) { ... }
}
```

Utilisez un tilde (~) sur l'attribut `method` pour remplacer le préfixe de route si nécessaire:

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

    ...
}
```

Route par défaut

Vous pouvez également appliquer l'attribut `[Route]` au niveau du contrôleur, en capturant l'action en tant que paramètre. Cette route serait alors appliquée à toutes les actions du contrôleur, sauf si une `[Route]` spécifique a été définie pour une action spécifique, en remplacement du paramètre par défaut du contrôleur.

```
[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) { ... }
}
```

Contraintes de Route

Les contraintes de route vous permettent de limiter la correspondance entre les paramètres du modèle de route. La syntaxe générale est `{paramètre: contrainte}`. Par exemple:

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

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

Ici, le premier itinéraire ne sera sélectionné que si le segment "id" de l'URI est un entier. Sinon, le deuxième itinéraire sera choisi.

Const	Description (correspondances :)	Exemple
alpha	Caractères alphabétiques latins majuscules ou minuscules (az, AZ)	{x: alpha}
bool	Valeur booléenne	{x: bool}
datetime	Valeur DateTime.	{x: datetime}
décimal	Valeur décimale	{x: décimale}
double	Valeur à virgule flottante 64 bits.	{x: double}
flotte	Valeur à virgule flottante 32 bits.	{x: float}
guid	Valeur GUID.	{x: guid}
int	Valeur entière de 32 bits.	{x: int}
longueur	Chaîne avec la longueur spécifiée ou dans une plage de longueurs spécifiée.	{x: length (6)} {x: length (1,20)}
longue	Valeur entière 64 bits.	{x: long}
max	Entier avec une valeur maximale.	{x: max (10)}
longueur maximale	Chaîne avec une longueur maximale.	{x: maxlength (10)}
min	Entier avec une valeur minimale.	{x: min (10)}
Longueur minimale	String avec une longueur minimale.	{x: minlength (10)}
gamme	Entier dans une plage de valeurs.	{x: range (10,50)}
regex	Expression régulière.	{x: regex (^ \ d {3} - \ d {3} - \ d {4} \$)}

Lire Routage d'attribut dans mvc-5 en ligne: <https://riptutorial.com/fr/asp-net-mvc-5/topic/6370/routage-d-attribut-dans-mvc-5>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec asp.net-mvc-5	Community , Hamzaway , hasan , Meghraj , Rosdi Kasim , Twister1002
2	Contrôleur asynchrone dans MVC 5	vicky
3	Créer des aides HTML	Mihail Stancescu
4	Routage d'attribut dans mvc-5	Anik Saha , PedroSouki