

Créer des contraintes de route pour vos applications MVC

par Louis-Guillaume MORAND (Traduction) ([Page perso de Louis-Guillaume MORAND](#))


Date de publication : mai 2009

Dernière mise à jour :

Dans ce tutoriel, Stephen Walther démontre comment vous pouvez contrôler comment les requêtes navigateur correspondent à des routes particulières, en créant de contraintes de routing avec des expressions régulières.

Traduction.....	3
Création d'une contrainte de route.....	3

Traduction

Cet article est la traduction la plus fidèle possible de l'article original :  **Creating a Route Constraint**

Création d'une contrainte de route

Vous utilisez des contraintes de route pour restreindre les requêtes navigateur qui correspondent à une route particulière. Vous pouvez utiliser les expressions régulières pour spécifier une contrainte de route.

Par exemple, imaginez que vous avez défini la route suivante dans votre fichier Global.asax.

Global.asax.cs

```
routes.MapRoute( "Product", "Product/{productId}", new {controller="Product", action="Details"} );
```

Le code précédent contient une route nommée Product. Vous pouvez utiliser la route Product pour mapper les requêtes navigateur vers le ProductController présenté dans le code suivant.

Controllers\ProductController.cs

```
using System.Web.Mvc;  
namespace MvcApplication1.Controllers  
{  
    public class ProductController : Controller  
    {  
        public ActionResult Details(int productId) { return View(); }  
    }  
}
```

Notez que l'action Details exposée par le contrôleur Product accepte un unique argument nommé productId. Ce paramètre est de type entier.

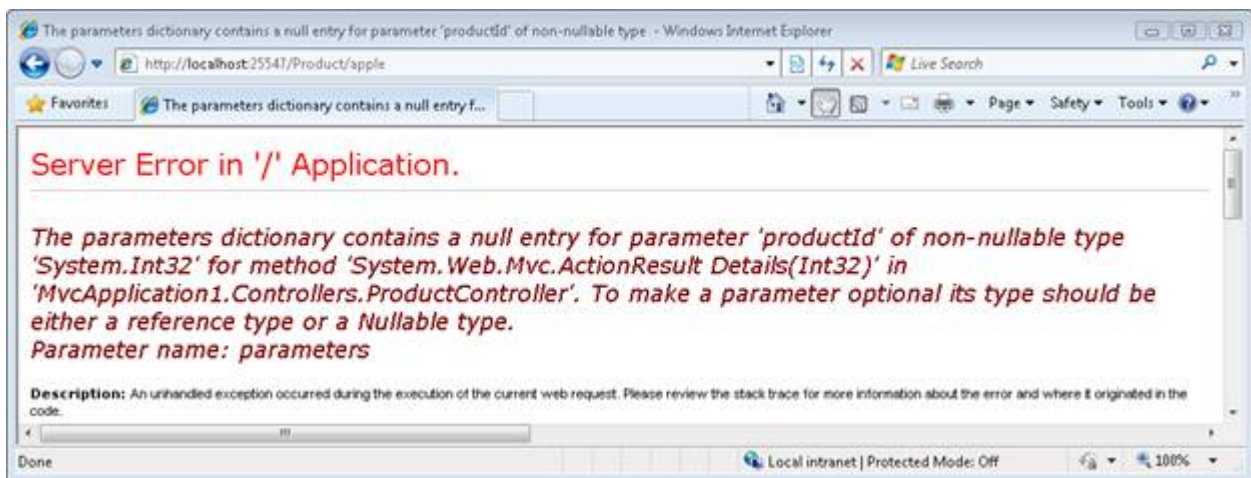
La route définie dans le fichier Global.asax correspondra à n'importe laquelle des adresses suivantes :

- /Product/23
- /Product/7

Malheureusement, cela va aussi correspondre aux URLs suivantes :

- /Product/blah
- /Product/apple

Parce que l'action Details attend un paramètre entier, faire une requête qui contient un paramètre qui soit autre chose qu'un entier causera une erreur. Par exemple, si vous tapez l'URL /Product/apple dans votre navigateur, alors vous allez obtenir la page d'erreur suivante :



Ce que vous devez vraiment faire est simplement de prendre en compte les URLs qui contiennent un productId qui soit un entier. Vous pouvez utiliser une contrainte de route justement pour restreindre les bonnes URLs. La route Product modifiée suivante contient une expression régulière pour ne prendre en compte que des entiers.

```
Global.asax.cs
```

```
routes.MapRoute( "Product", "Product/{productId}", new {controller="Product",  
    action="Details"}, new {productId = @"\d+" } );
```

L'expression régulière `ld+` correspond à un ou plusieurs entiers. Cette contrainte force la route Product à correspondre à l'une des URLs suivantes :

- /Product/3
- /Product/8999

Mais pas à ces adresses-là :

- /Product/apple
- /Product

Ces requêtes navigateur vont être récupérées par une autre route ou, s'il n'y a pas de route correspondant, une erreur "The resource could not be found" sera retournée.