
CONFIGURATION DES CHEMINS D'ACCÈS (ROUTAGE)

- Le moteur de routage ASP.NET MVC
- Configuration du routage
- Passer des paramètres en utilisant le routage
- Test unitaire sur les règles de routage

Le moteur de routage ASP.NET MVC

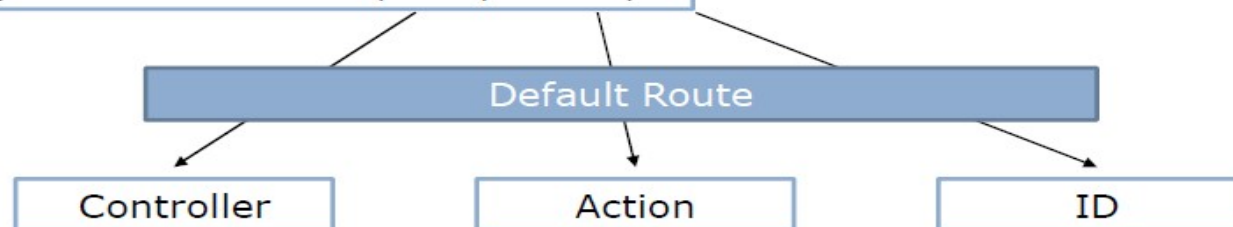
- Le routage décrit la manière dont les URL sont formulées et comment ils sont liés aux contrôleurs et aux actions.
- Le Routage ne concerne pas le protocole, serveur, domaine, ou le numéro de port d'une URL
- Il configure seulement la manière d'appeler les contrôleurs et leurs éventuelles actions.
- Lors de la création d'une application Web MVC dans Visual Studio à l'aide d'un Template de projet, l'application a une route par défaut.
- On doit comprendre cette route par défaut avant d'envisager d'ajouter des routes supplémentaires pour manipuler les URL d'une application.

LE MOTEUR DE ROUTAGE ASP.NET MVC

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

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

http://www. ASP.net .com/todo/details/1



LE MOTEUR DE ROUTAGE ASP.NET MVC

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

- «Default », permet de réagir à des requêtes de type /xxx/yyy/zzz
- Ces requêtes vont instancier le contrôleurxxxController , et appeler la méthode yyy() en lui passant en paramètre la valeur zzz

une requête /Home/Index/123 permettra d'appeler la méthode suivante :

Ce qui affichera la chaîne :
HomeController.Index 123

```
public class HomeController : Controller  
{  
    // references  
    public string Index(string id)  
    {  
        return "HomeController.Index " + id;  
    }  
}
```

LE MOTEUR DE ROUTAGE ASP.NET MVC : CRÉER DES ROUTES

```
routes.MapRoute(  
    name: "Default",  
    url: "{action}/{valeur1}/{valeur2}",  
    defaults: new { controller = "Calculateur", action = "Ajouter", valeur1 = 0, valeur2 = 0 }));
```

```
public class CalculateurController : Controller  
{  
    0 references  
    public string Ajouter(int valeur1, int valeur2)  
    {  
        int resultat = valeur1 + valeur2;  
        return resultat.ToString();  
    }  
}
```

/Ajouter/5/11 et donne le résultat 16

LE MOTEUR DE ROUTAGE ASP.NET MVC : DÉFINIR PLUSIEURS ROUTES

```
routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
routes.MapRoute(
    name: "Default",
    url: "{action}/{valeur1}/{valeur2}",
    defaults: new { controller = "Calculateur", action = "Ajouter", valeur1 = 0, valeur2 = 0 });

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

- l'ordre dans lequel les définitions des routes sont ajoutées est important et détermine l'ordre dans lequel seront testées les différentes routes.

LE MOTEUR DE ROUTAGE ASP.NET MVC : LES CONTRAINTES DE ROUTE

- Les contraintes sont des règles sur les segments d'URL
- Toutes les contraintes sont des expressions régulières compatible avec la classe **Regex**

```
// 2013/01/29/Blog-title
routes.MapRoute(
    name: "Blog",
    url: "{year}/{month}/{day}",
    defaults: new { controller = "Blog", action = "ByDate" },
    constraints: new { year=@"\d{4}", month=@"\d{2}", day=@"\d{2}" }
);
```