

# Les Routes ASP.NET

## Crée des routes

Chaque action de chaque contrôleur a maintenant un filtre de route.

```
[Route("Programme", Name = "Index-programme")]  
0 références | Carl Gauthier, il y a 9 jours | 2 auteurs, 4 modifications | 0 requête | 0 exceptions  
public ActionResult Index()  
{  
    return View(db.Programme.ToList());  
}
```

Le premier paramètre c'est ce qu'on souhaite faire apparaître dans l'URL après le nom de domaine.

Le deuxième paramètre est le nom de la route. **Elle doit être unique**. Ce nom sera utilisé pour appeler la route.

Bien sûr on peut aussi ajouter des paramètres pour recevoir des ID ou autres.

```
[Route("Programme/{id:int?}", Name = "Info-programme")]  
0 références | lordfred23, il y a 9 jours | 2 auteurs, 3 modifications | 0 requête | 0 exceptions  
public ActionResult Info(int? id)  
{  
    if (id == null)  
    {  
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  
    }  
    Programme programme = db.Programme.Find(id);  
}
```

Le ou les paramètres à passer se rajoutent dans le premier paramètre de la route dans les accolades {nom du paramètre :type}. La slash barre est optionnelle : ASP.NET va rajouter un ? avant le paramètre.

## Utilisation des routes

L'appel de la route ne se fait plus avec le `html.action`, car il est maintenant impossible de faire cette action quand l'action a un filtre de route. On doit maintenant utiliser ***Url.RouteUrl*** comme ci-dessous.

Le premier paramètre est le nom de la route qu'on souhaite accéder que nous avons choisi auparavant.

Le Deuxième paramètre est optionnel c'est les id et autres paramètres. Il s'agit simplement de faire comme ci-dessous avec les bons noms de paramètre.

```
<a class="info-child clearfix" href="@Url.RouteUrl("Info-programme", new { id = programme.idProgramme })">
  <p class="gauche">@Html.DisplayFor(ec => programme.EnteteProgramme.commentaire)</p>
  <p class="droite">@Html.DisplayFor(model => programme.codeProgramme)-@Html.DisplayFor(model => programme)
</a>
```

Pour que les `BeginForm` envoie maintenant les infos aux bon contrôleur et action nous devons maintenant lui spécifier le contrôleur et l'action souhaité en plus car ***il n'y pu d'Url de type contrôleur/action***. Pour certaine action il n'est pas nécessaire de l'ajouté c'est à vous de le tester sans avant.

```
@using (Html.BeginForm("Create", "ConsoleProgramme", FormMethod.Post))
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <p class="text-danger">@ViewBag.messageError</p>
```

Le Premier paramètre est l'action et le deuxième est le contrôleur ***le filtre de route de ces actions sert qu'à ne pas montrer le lien du contrôleur et de l'action*** en format `www.nomdomaine.com/contrôleur/action`