# Vues

# Retourner une vue

Avant de retourner une vue, il faut d'abord créer un dossier dans `Views` pour notre controleur qui correspond à son nom. Ici notre controleur s'appelle `HelloWorldController`, notre dossier va donc s'appeller `HelloWorld`.

On va ensuite créer dedans un fichier `hello-world.cshtml`, qui sera notre template :

```
<h1>Hello, world</h1>
```

Pour le retourner depuis une méthode endpoint, il faut utiliser la fonction `View()` du controleur :

```
    [HttpGet]
    public IActionResult HelloWorld()
    {
        return View("hello-world");
    }
```

On peut ensuite constater le résultat en redémarrant l'app (CTRL + C dans la console puis `dotnet run`). Et en allant à https://localhost:7229/hello-world.

# Templater une vue

## Avec une classe modèle

On peut templater une vue avec un modèle de façon fortement typée, ce qui est pratique car cela permet de répérer d'eventuelles erreurs plus vites.

Soit la classe de modèle (dans le dossier `Model`, et n'oublier pas de décomenter l'import du namespace dans `_ViewImports.cshtml`) :

```
namespace net_web_tuto.Models {
    public class Person {
        public string FirstName { get; set; }
        public string LastName { get; set; }
    }
}
```

Dans la méthode endpoint, on ajouter le modèle dans l'appel à `View` :

```
    [HttpGet]
    public IActionResult HelloWorld()
    {
        var person = new Person(){ FirstName = "John", LastName = "Shepard"};
        return View("hello-world", person);
    }
```

Et pour le template, la syntaxe est la suivante :

```
<h1>Hello, @Model.FirstName @Model.LastName</h1>
```

On peut ensuite constater le résultat en redémarrant l'app (CTRL + C dans la console puis `dotnet run`). Et en allant à https://localhost:7229/hello-world.

# Avec d'autres données

Pour templater la vue avec d'autres infos que celles d'un modèle, on peut utiliser le `ViewBag`, exemple :

Dans la méthode endpoint, on ajouter le modèle dans l'appel à `View` :

```
    [HttpGet]
    public IActionResult HelloWorld()
    {
        var person = new Person(){ FirstName = "John", LastName = "Shepard"};
      ViewBag.Today = DateTime.Now;
        return View("hello-world", person);
    }
```

```
<h1>Hello, @Model.FirstName @Model.LastName, The Date is : @ViewBag.Today</h1>
```

On peut ensuite constater le résultat en redémarrant l'app (CTRL + C dans la console puis `dotnet run`). Et en allant à https://localhost:7229/hello-world.

# Rendu conditionnel

Pour faire du rendu conditionnel il faut utiliser `@if` :

```
@if((20 % 2) == 0) {
    <p> 20 is even </p>
}

@if((21 % 2) == 0) {
    <p> 21 is not even </p>
}
```

Seulement "20 is even" sera rendu sur la page.

## Boucles

Pour faire du rendu en boucle, il faut utiliser `@foreach`, exemple :

```
@foreach (var person in Model.Persons)

    <div>@item.FirstName @item.LastName</div>
}
```

---

Revision #1
Created 2022-09-28 12:54:54 UTC by Arsène Lapostolet
Updated 2022-09-28 17:36:46 UTC by Arsène Lapostolet