

TP N° 4

ASP.NET Core MVC : Contrôleur, Vue, Routage

Créer une application ASP.NET Core MVC nommée PremiereAppWeb permettant :

- L'affichage d'une liste des étudiants
- L'affichage d'une liste des professeurs

1. Créer le projet PremiereAppWeb (via Visual Studio)

Avant de commencer, assurez-vous d'avoir installé les outils suivants :

- ✓ Ouvrir Visual Studio
- ✓ Aller dans Fichier > Nouveau > Projet
- ✓ Choisir : ASP.NET Core Web App (Model-View-Controller)
- ✓ Cliquer sur Suivant
- ✓ Nom du projet : PremiereAppWeb
- ✓ Choisir un emplacement de sauvegarde
- ✓ Cliquer sur Créer
- ✓ Choisir le Framework .NET 6 ou supérieur
- ✓ Laisser les options par défaut, puis cliquer sur Créer

2. Créer les modèles

- Dossier : Models
- Créer le modèle Etudiant
- Fichiers : Etudiant.cs

```
public class Etudiant
{
    public int Id { get; set; }
    public string Nom { get; set; }
    public string Prenom { get; set; }
    public int Age { get; set; }

}
```

- Dossier : Models
- Créer le modèle Prof
- Fichiers : Prof.cs

```

public class Prof
{
    public int Id { get; set; }
    public string Nom { get; set; }
    public string Matiere { get; set; }
    public int Experience { get; set; }

}

```

3. Créer les contrôleurs

- ✓ Clic droit sur le dossier Controllers > Ajouter > Contrôleur
- ✓ Choisir **Contrôleur MVC – Vide**
- ✓ Nom : *EtudiantsController.cs*
- ✓ Refaire les mêmes étapes pour *ProfsController.cs*

```

public class EtudiantsController : Controller
{
    public IActionResult Index()
    {
        var listeEtudiants = new List<Etudiant>
        {
            new Etudiant { Id = 1, Nom = "Benali", Prenom = "Amine", Age = 21 },
            new Etudiant { Id = 2, Nom = "Nouri", Prenom = "Salma", Age = 22 },
            new Etudiant { Id = 3, Nom = "Zerouali", Prenom = "Said", Age = 20 }
        };

        return View(listeEtudiants);
    }
}

```

⇒ Ajouter le fichier *ProfsController.cs* de même logique

Élément	Rôle
IActionResult	Type de retour MVC, permet de retourner une vue, un message, ou une redirection
Index()	Action qui retourne la liste des données
View(...)	Appelle une vue et lui passe un modèle

4. Créer les vues html Index.cshtml

Chemins:

- Views/Etudiants/Index.cshtml
- Views/Profs/Index.cshtml

Exemple de vue Views/Etudiants/Index.cshtml :

```

@model List<ApplicationWeb.Models.Etudiant>

 @{
    ViewData["Title"] = "Liste des étudiants";
}

```

```

        }

        <h2>@ ViewData["Title"]</h2>

        <table border="1" cellpadding="5">
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Nom</th>
                    <th>Prénom</th>
                    <th>Âge</th>
                </tr>
            </thead>
            <tbody>
                @foreach (var etudiant in Model)
                {
                    <tr>
                        <td>@etudiant.Id</td>
                        <td>@etudiant.Nom</td>
                        <td>@etudiant.Prenom</td>
                        <td>@etudiant.Age</td>
                    </tr>
                }
            </tbody>
        </table>
    
```

Élément Razor	Signification
@model	Déclare le type du modèle utilisé dans la vue
@Model.Propriété	Accès aux données passées depuis le contrôleur
@foreach	Boucle sur une liste pour afficher plusieurs lignes
@{}	Bloc de code C# intégré dans HTML
cshtml	Fichiers Razor : mélangeant HTML + C#

- ⇒ Ajoutez la vue Prof.cshtml de même logique
- ⇒ Ajouter une vue pour afficher les détails d'un étudiant (par id)

```

public IActionResult Details(int id)
{
    /*
    *
    *
    */
}
    
```

5. Ajouter les routages

Dans le programme : Program.cs

```

app.MapControllerRoute(
    name: "prof(alias",
    pattern: "nos-profs/{action=Index}/{id?}",
    defaults: new { controller = "Profs" });

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Etudiants}/{action=Index}/{id?}");
    
```

Élément	Rôle
MapControllerRoute(...)	Configure le lien URL → contrôleur/action
pattern: "{controller}/{action}/{id?}"	Modèle de l'URL
defaults:	Valeurs par défaut si rien n'est précisé
Routage multiple	Permet de créer des alias personnalisés

6. Exécuter le projet