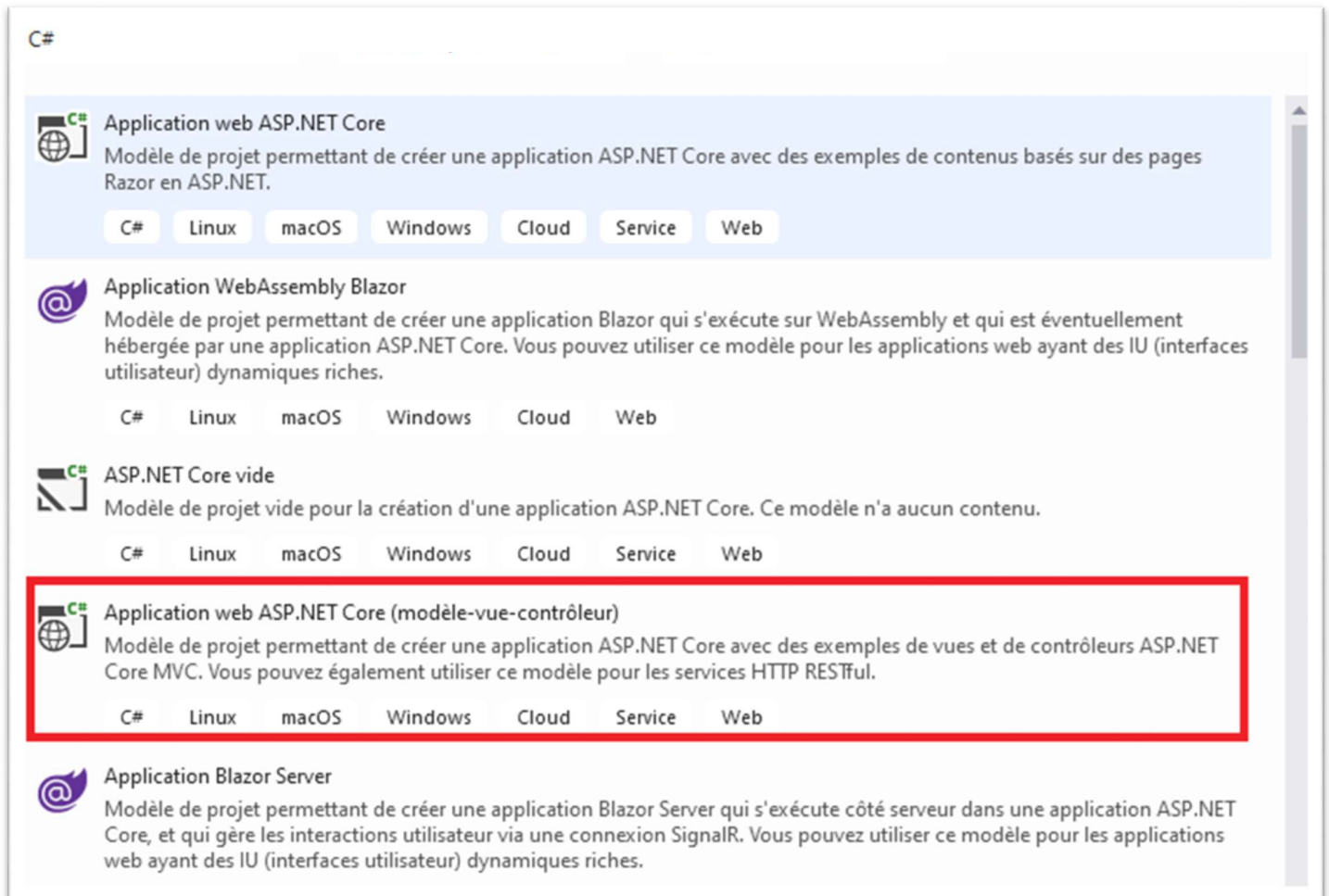


## *Tp Asp.net Core(Navigation)*

- Créer une Application Asp.net Core MVC :



- Dans le dossier Models Ajouter les classes suivantes :Etudiant,Groupe,Matiere,Inscription
- Cliquez avec le bouton droit sur le dossier *Models* > **Ajouter** > **Classe**.

```
public class Groupe
{
    public int Id { get; set; }
    [Required]
    [StringLength(30, MinimumLength = 3)]
    [Display(Name = "Groupe")]
    public string LibGroupe { get; set; }
    public virtual ICollection<Etudiant> Etudiants { get; set; }
}
```

```

public class Etudiant
{
    public int Id { get; set; }
    [Required]
    [StringLength(30, MinimumLength = 3)]
    public string Nom { get; set; }
    [Required(ErrorMessage = "Prénom Obligatoire")]
    [StringLength(30, MinimumLength = 3)]
    public string Prenom { get; set; }
    [Display(Name = "Date De Naissance")]
    public DateTime DateN { get; set; }
    [Display(Name = "Groupe")]
    public int GroupeId { get; set; }
    public virtual Groupe Groupe { get; set; }
    public virtual ICollection<Inscription> Inscriptions { get; set; }
}

```

```

public class Matiere
{
    public int Id { get; set; }
    [Display(Name = "Matière")]
    [Required, StringLength(30, MinimumLength = 3)]
    public string LibMatiere { get; set; }
    public virtual ICollection<Inscription> Inscriptions { get; set; }
}

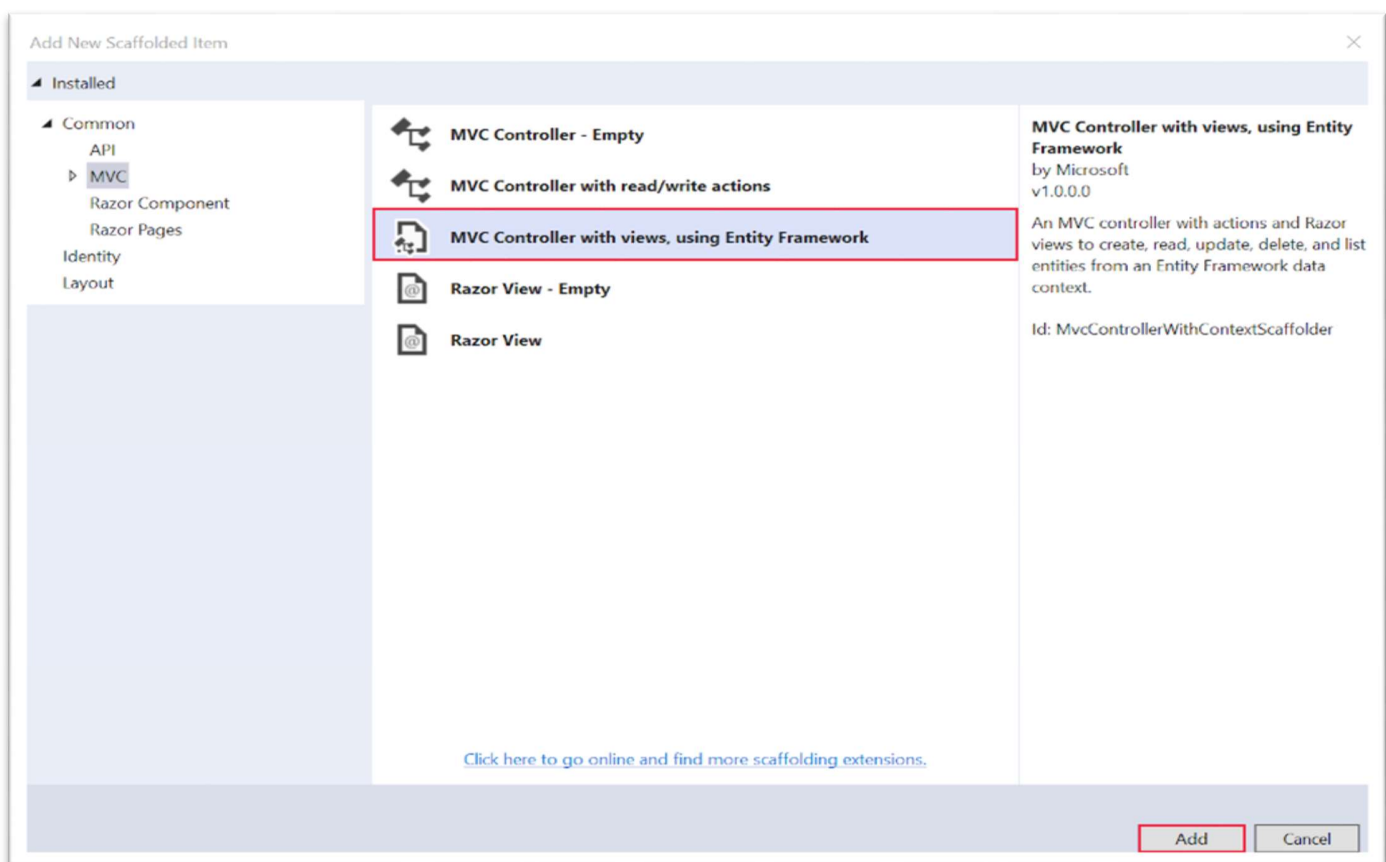
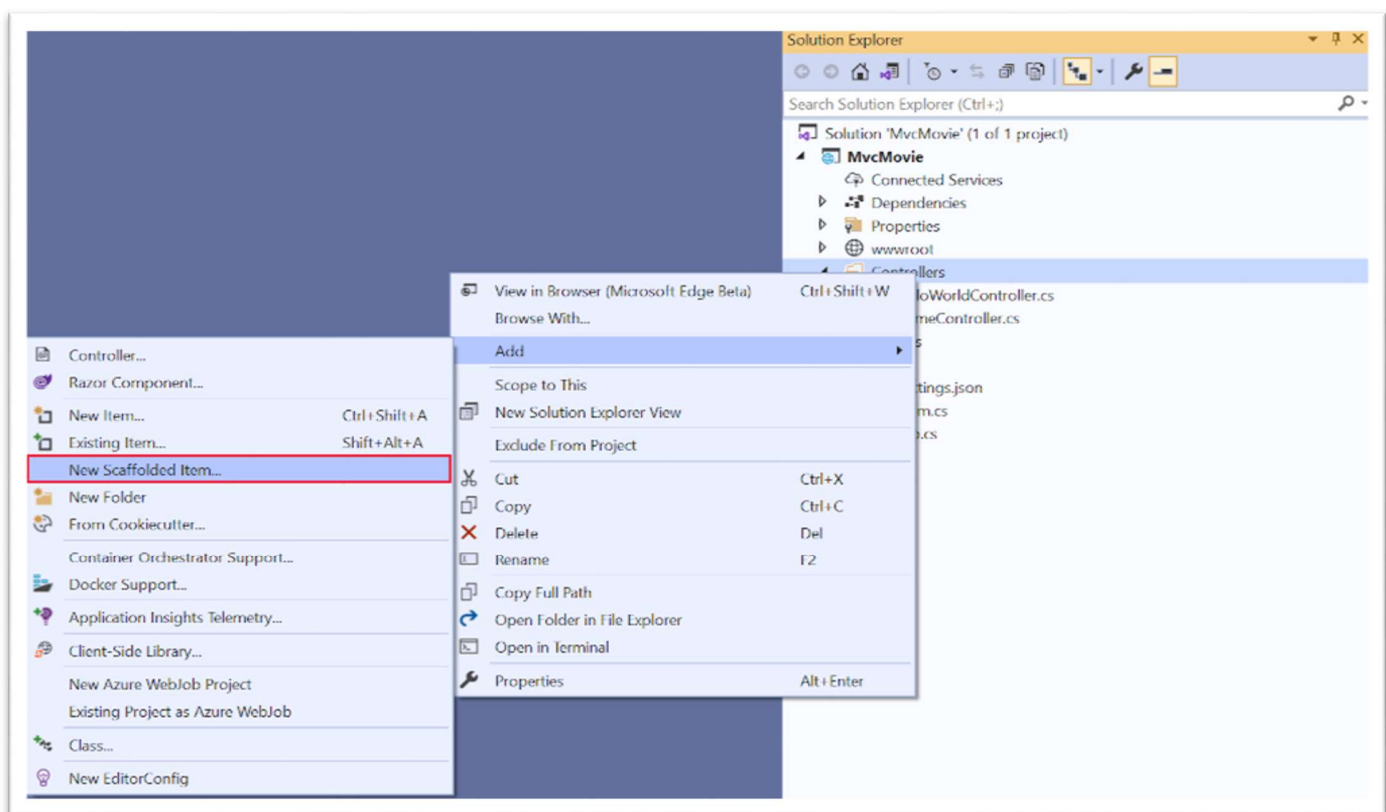
```

```

public class Inscription
{
    public int Id { get; set; }
    [Display(Name = "Matière")]
    public int MatiereId { get; set; }
    [Display(Name = "Etudiant")]
    public int EtudiantId { get; set; }
    public virtual Etudiant Etudiant { get; set; }
    public virtual Matiere Matiere { get; set; }
}

```

- Création du contrôleur **Groupes**
- Dans **Explorateur de solutions**, cliquez avec le bouton droit sur le dossier *Controllers* et sélectionnez **Ajouter > nouvel élément** généré automatiquement.



- Complétez la boîte de dialogue **Ajouter un contrôleur MVC avec des affichages**, à l'aide de **Entity Framework** :
- Dans la liste déroulante **classe de modèle** , sélectionnez **Groupe** .
- Dans la ligne **Classe du contexte de données**, sélectionnez le signe + (plus).
  - ✓ Dans la boîte de dialogue **Ajouter un contexte de données** , le nom de la classe est `:ScolariteContext` .
  - ✓ Sélectionnez **Ajouter**.
- **Affichages** et **nom du contrôleur**: conservez la valeur par défaut.
- Sélectionnez **Ajouter**.

Add MVC Controller with views, using Entity Framework

Model class

Data context class  +

Views

☒ Generate views

☒ Reference script libraries

☒ Use a layout page

...

(Leave empty if it is set in a Razor \_viewstart file)

Controller name

Add Cancel

La génération de modèles automatique met à jour les éléments suivants :

- ✓ Insère les références de package requises dans le fichier projet *ScolariteApp.csproj* .
- ✓ Inscrit le contexte de base de données dans `Startup.ConfigureServices` du fichier *Startup.cs* .
- ✓ Ajoute une chaîne de connexion de base de données au fichier *appsettings.json* .

La génération de modèles automatique crée les éléments suivants :

- ✓ Un contrôleur de films : *Controllers/GrouperController.cs*
  - ✓ Razor afficher les fichiers pour les pages de **création**, de **suppression**, de **Détails**, de **modification** et d' **index** : *Views/Grouper/\*.cshtml*
  - ✓ Classe de contexte de base de données : *Data/ScolariteContext.cs*
- Dans le au fichier *appsettings.json* ,Changer le nom de la base de données généré automatiquement en **ScolariteDb**

```
'ConnectionStrings': {  
  "ScolariteContext": "Server=(localdb)\\mssqllocaldb;Database=ScolariteDb;Trusted_Connection=True;MultipleActiveResultSets=true"  
}
```

- Vérifier que la classe du contexte **ScolariteContext** Contient toutes les **DbSet**

```
public class ScolariteContext : DbContext  
{  
    public ScolariteContext (DbContextOptions<ScolariteContext> options)  
        : base(options)  
    {  
    }  
    public DbSet<Groupe> Groupe { get; set; }  
  
    public DbSet<Etudiant> Etudiant { get; set; }  
  
    public DbSet<Inscription> Inscription { get; set; }  
  
    public DbSet<Matiere> Matiere { get; set; }  
}
```

- Dans le menu **outils** , sélectionnez **Gestionnaire de package NuGet > Console du Gestionnaire de package**.
- Dans la console du Gestionnaire de package, entrez les commandes suivantes :

```
Add-Migration mg1  
Update-Database
```

- Une fois la base de données est créée Ajouter les contrôleurs restants en utilisant le même principe
- Exécuter l'application et tester toute l'application(les 4 Contrôleurs)

Dans la Vue **Create** du contrôleur **Inscription**, on peut vérifier que la liste déroulante Matière Contient l'**Id** Matière et non pas le libellé, aussi la liste déroulante Etudiant contient seulement le **Nom**.

- Dans la classe Etudiant Ajouter la Propriété NomPrenom Suivante :

```
public string NomPrenom {
    get
    {
        return Nom + " " + Prenom;
    }
}
```

- Dans l'action **Create** du contrôleur **Inscriptions** modifier les deux **SelectList** comme suit :

```
ViewData["EtudiantId"] = new SelectList(_context.Etudiant, "Id", "NomPrenom");

ViewData["MatiereId"] = new SelectList(_context.Matiere, "Id", "LibMatiere");
```

- Tester le contrôleur **Inscriptions** :

- Accédez au page maitre du projet ayant le chemin suivant « **Views/Shared/\_Layout.cshtml** » :
- Ajoutez les actions Index du contrôleurs Etudiants, Groupes, Matieres, Inscriptions au menu principale de l'application.

```
<li class="nav-item">
  <a class="nav-link text-dark" asp-area="" asp-controller="Etudiants" asp-
action="Index"> Etudiant
</a>
</li>
```

- On envisage afficher la liste des étudiants inscrit dans chaque groupe.
- Accéder à la vue « **Vues/Groupes/Details.cshtml** »
- Ajouter le code suivant dans cette vue, juste au-dessus de la dernière balise **Div** :

```
<table class="table">
  <tr>
    <th>Nom</th>
    <th>prénom</th>
    <th>Date De Naissance</th>
  </tr>
  @foreach (var item in Model.Etudiants)
  {
    <tr>
      <td>@Html.DisplayFor(x => item.Nom)</td>
      <td>@Html.DisplayFor(x => item.Prenom)</td>
      <td>@Html.DisplayFor(x => item.DateN)</td>
    </tr>
  }
</table>
```

Modifier l'action **Details** du contrôleur Groupes de la façon suivante :

```
var groupe = await _context.Groupe.Include(x=>x.Etudiants)
    .FirstOrDefaultAsync(m => m.Id == id);
```

- On envisage afficher la liste des étudiants inscrit dans chaque Matière ainsi que leurs Groupes.
- Accéder à la vue « **Vues/Matieres/Details.cshtml** »
- Ajouter le code suivant dans cette vue, juste au-dessus de la dernière balise **Div**

```
<table class="table">
  <tr>
    <th>Nom</th>
    <th>prénom</th>
    <th>Date De Naissance</th>
    <th>Groupe</th>
  </tr>
  @foreach (var item in Model.Inscriptions)
  {
    <tr>
      <td>@Html.DisplayFor(x => item.Etudiant.Nom)</td>
      <td>@Html.DisplayFor(x => item.Etudiant.Prenom)</td>
      <td>@Html.DisplayFor(x => item.Etudiant.DateN)</td>
      <td>@Html.DisplayFor(x => item.Etudiant.Groupe.LibGroupe)</td>
    </tr>
  }
</table>
```

- Modifier l'action **Details** du contrôleur Matieres de la façon suivante :

```
var matiere = await _context.Matiere.Include(x=>x.Inscriptions)
    .ThenInclude(x=>x.Etudiant).ThenInclude(x=>x.Groupe)
    .FirstOrDefaultAsync(m => m.Id == id);
```