Frameworks de développement

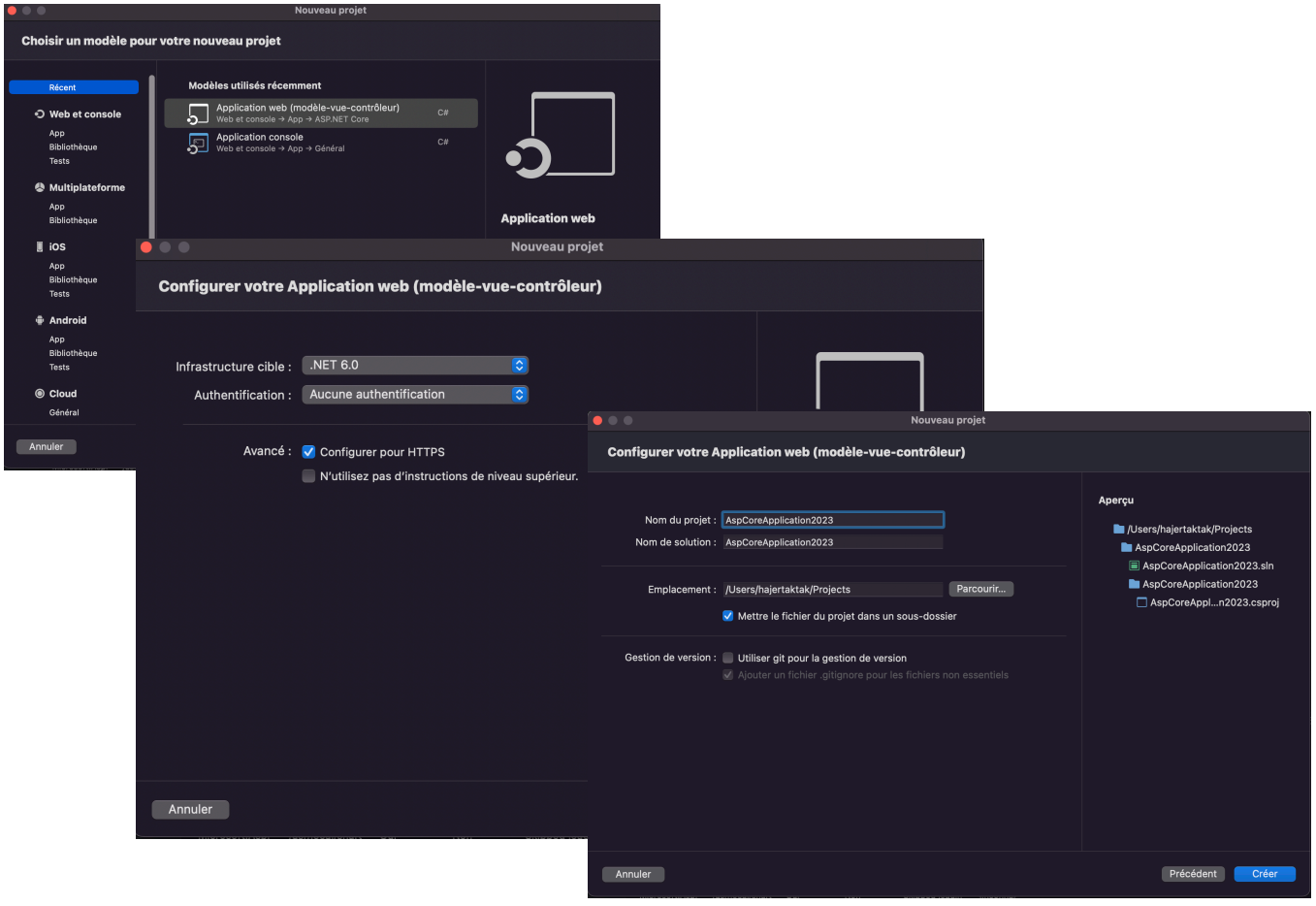
TP1 ASP.Net Core MVC & routage

MVC est synonyme de *modèle-vue-contrôleur*. MVC est un pattern pour développer des applications qui sont bien architecturé, testables et facile à entretenir. MVC veut dire :

- **M**odel : Classes qui représentent les données de l'application et qui utilisent une logique de validation pour fairerespecter les règles de métier.
- **V**iew : fichiers de modèle que votre application utilise pour générer dynamiquement les réponses HTML.
- **C**ontroller : Classes qui gèrent les demandes entrantes de navigateur, récupérer des données de modèle etensuite spécifier des modèles de vue qui retournent une réponse au navigateur.

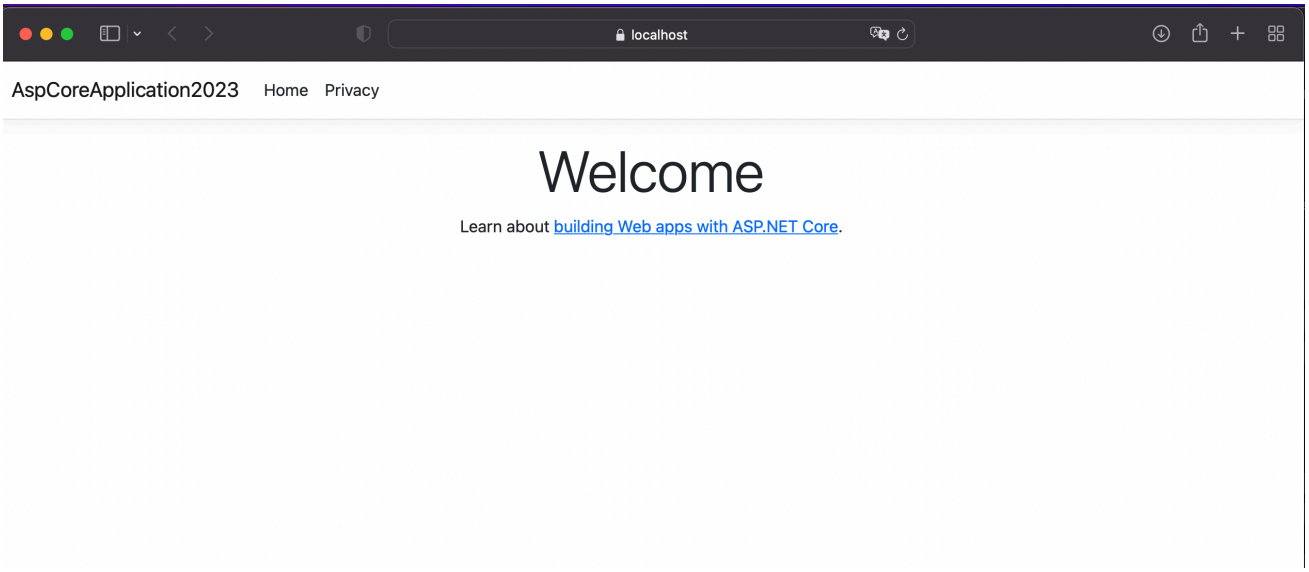
Ce TP explique comment se fait le routage en ASP.Net Core MVC.

- Cliquez sur **New Project**, puis sélectionnez Visual C#, puis **Web** et puis sélectionnez **ASP.NET Core Application Web**. Nommez votre projet «**ASPCoreApplication2023**» et puis cliquez sur **OK**.



- Cliquez sur F5 pour démarrer le débogage. F5 oblige Visual Studio à lancer IIS Express et exécuter votre application web Visual Studio puis lance un navigateur et ouvre la page d'accueil de l'application.

Comme vous remarquez la barre d'adresse du navigateur contient : **localhost:port#** et non pas quelque chose comme **example.com**. C'est parce que **localhost** pointe toujours vers votre ordinateur local, qui dans ce cas exécute l'application que vous venez de créer. Lorsque Visual Studio exécute un projet web, un port aléatoire est utilisé pour le serveur web. Dans l'image ci-dessus, le numéro de port est 7292. Un port est assigné à chaque application

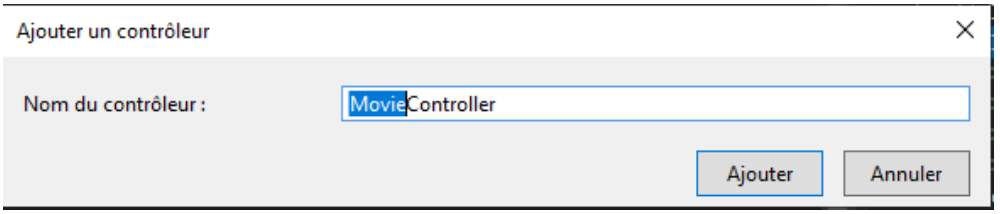


Création des classes du domaine (Model)

- Créer une classe Movie (int Id, string Name)

Création d'un contrôleur

- Nous allons commencer par créer une classe de contrôleur. Dans de **L'Explorateur de solutions**,faites un clic droit sur le dossier controllers, puis cliquez sur **Add**, puis **controller**.
- Puis cliquer sur Controller – Emptyensuite sur **Add**



Nommez

le nouveau contrôleur

MovieController

Dans **L'Explorateur** un nouveau fichier a été créé nommé ***MovieController.cs*** et un nouveau dossier ***Views\Movie***. Le contrôleur est ouvert dans l'IDE.

```
namespace AspCoreApplication2023.Controllers
{
   public class MovieController : Controller
   {
      // GET: /<controller>/
      public IActionResult Index()
      {
         return View();
      }
   }
}
```

- Remplacez le contenu du fichier avec le code suivant.

```
public IActionResult Index()
     {
       Movie movie = new Movie() { Name =
       "movie 1" };
       List<Movie> movies = new List<Movie>()
       {
         new Movie{Name="movie 2"},
         new Movie{Name="movie 3"},
       };
       return View(movies);
     }
```

Le contrôleur est appelé **MovieController** et la première méthode ou **Action** est nommée **Index**. Nous allons l'appeler à partir d'un navigateur.

Configuration des routes

MapControllerRoute est utilisé pour créer une route unique. La route unique est nommé default route. La plupart des applications avec des contrôleurs et des vues utilisent un modèle de route default similaire à la route. Les API doivent utiliser le routage d'attributs.

- La première partie de l'URL détermine le contrôleur qu'on veut exécuter. Si ***/Movie*** alors la classe **MovieController** sera appelée.
- La deuxième partie de l'URL détermine la méthode d'action à Exécuter.Si ***/Movie/index*** alors la méthode Index de la classe MovieController sera exécutée.

Une méthode nommée Index est la méthode par défaut et elle sera appelée sur un contrôleur si on n'a pas spécifié la méthode explicitement.

- La troisième partie du segment URL (Parameters) c'est pour acheminer les données.

> ⓘ Important
>
> Le routage est configuré à l'aide de l'intergiciel UseRouting et .UseEndpoints Pour utiliser des contrôleurs :
>
> - Appel MapControllers à des contrôleurs routés d'attribut de mappage.
> - Appelez MapControllerRoute ou MapAreaControllerRoute, pour mapper à la fois les contrôleurs routés de manière conventionnelle et les contrôleurs routés d'attribut .
>
> Les applications n'ont généralement pas besoin d'appeler `UseRouting` ou `UseEndpoints`. WebApplicationBuilder configure un pipeline d'intergiciels qui encapsule l'intergiciel ajouté dans `Program.cs` avec `UseRouting` et `UseEndpoints`. Pour plus d'informations, consultez Routage dans ASP.NET Core.

- Le mappage de MVC par défaut est /[Controller]/[ActionName]/[Parameters]. Pour cette URL, le contrôleur est Movie et Index est l'action. Nous n'avons pas utilisé les paramètres [Parameters] dans l'URL.

Nous allons à présent ajouter un Action à notre exemple afin de passer des informations de paramètre de l'URL au contrôleur (par exemple, */Movie/Edit/1*.

- Ajouter une méthode Edit pour inclure un paramètre « Id » indiqué ci-dessous

```
public IActionResult Edit(int id)
    {
       return Content("Test Id" + id);
    }
```

Dans l'exemple ci-dessus, le segment d'URL (Parameters) est utilisé et est transmis sous forme de paramètres de requête.

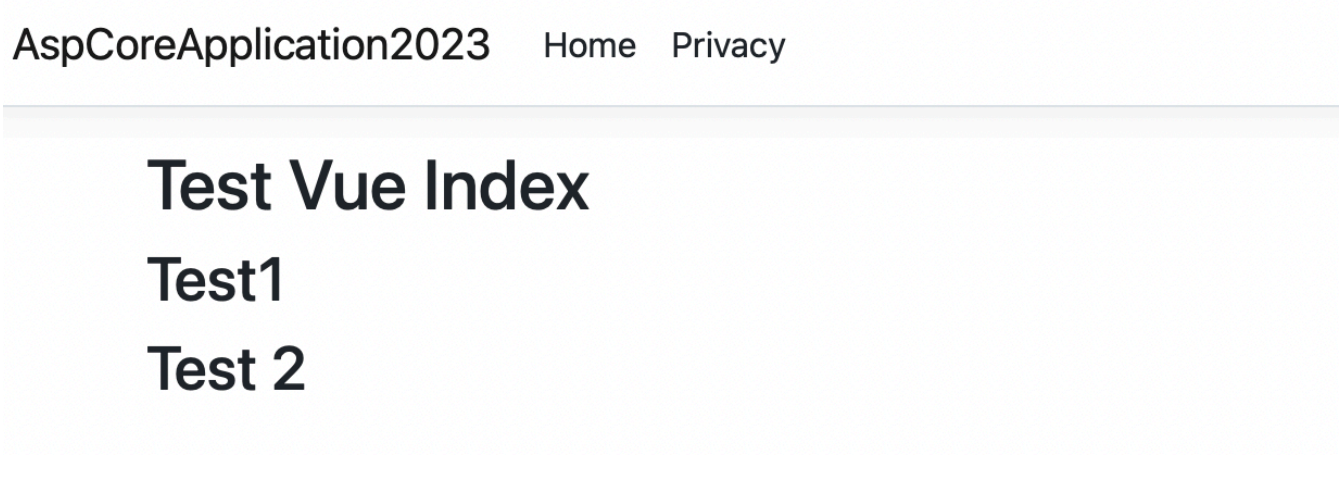
- Créer la vue pour accéder à http://localhost:xxxx/Movie/Index**.**

```
@model IEnumerable<AspCoreApplication2023.Models.Movie>

<h2>Test Vue Index</h2>

@foreach (var item in Model)
  {
     <h3>@item.Name</h3>
  }
```

La méthode d'action Index s'exécute et retourne les noms des films



C’est à vous?

1. **Modifiez le route pour inclure l’URL Movie/released/2020/03 pour une Action ByRelease ayant comme arguments (month, year) retournant juste un contenu.**
2. **Exécutez. Que remarquez vous? Quel changement à faire pour que le système de routage prend en charge cet route? Expliquez les différentes éventualités rencontrées.**
3. **Présentez avec des exemples les différents systèmes de routage si vous travaillez avec le framework de développement web ASP.Net.**
4. **On veut à présent passer deux modèles à la vue (Movie et Customer)**

- **Ajouter une classe Customer (Id, Name)**
- **On veut passer à la vue un film et une Liste de Clients (Penser à ajouter ViewModel)**
- **Lister (statique) les enregistrements au niveau du contrôleur (retourner le VM)**
- **Changer le modèle assigné à la vue**
- **Récupérer les détails du client par son Id.**