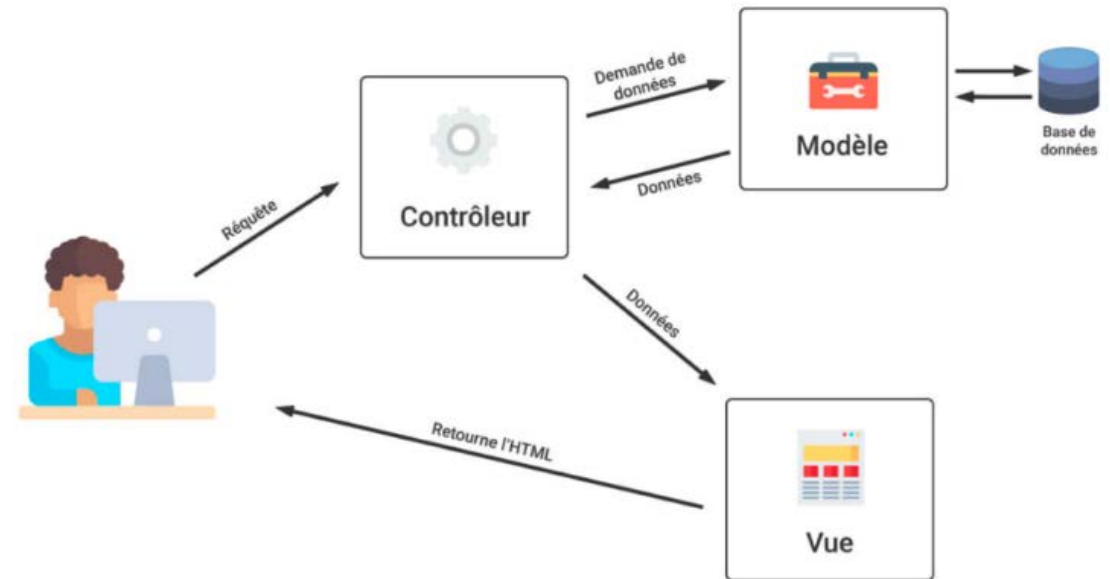


L'architecture d'une application web avec le modèle MVC et création d'un projet ASP.NET MVC

MOHAMED AIROUCHE

Le Modèle MVC en développement Web

- Le modèle MVC est un patron de conception utilisé pour réaliser des applications Web. Le but de modèle MVC est justement de séparer la logique du code en trois parties : le modèle (l'accès aux données), la vue (l'affichage des informations) et le contrôleur (les actions de l'utilisateur).

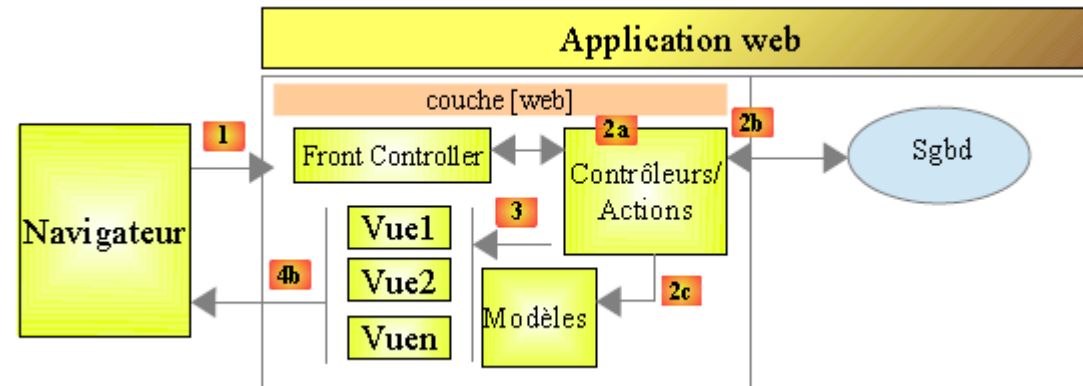


[Réf] <http://blogdummi.fr/dossier/architecture-mvc-diviser-pour-mieux-developper/>

- Modèle** : cette partie gère les *données* de l'application Web.
- Vue** : cette partie se concentre sur l'*affichage* (essentiellement du code HTML).
- Contrôleur** : est la partie principale de l'architecture, car c'est lui qui va faire le lien entre l'utilisateur, le modèle et la vue.

L'architecture MVC en ASP.NET

1. Le client envoie une requête HTTP vers le serveur.
2. Le contrôleur va lancer une action pour faire des traitements.
3. Le contrôleur récupère les résultats et les met dans le modèle et retourne la vue
4. Le serveur fourni de HTML pour le client qui affiche la vue. Le client n'a pas besoin de faire des calculs supplémentaires.



[Réf] Introduction à ASP.NET MVC par l'exemple, Serge Tahe, novembre 2013. <https://tahe.developpez.com/>

Le principe de fonctionnement de l'ASP.NET MVC

- Lorsqu'un utilisateur demande une page web depuis son navigateur, une requête HTTP est envoyée vers le serveur.
- Le premier défi du serveur sera de comprendre cette requête pour répondre à la demande de l'utilisateur. Cette requête est dans un premier temps traitée par le serveur en analysant les diverses informations transportées par cette requête.
- Le point d'entrée pour chaque application **MVC** commence avec **le routage**.
- ASP.NET MVC appelle différentes classes de contrôleur (et différentes méthodes d'action qu'ils contiennent) en fonction de l'URL entrante. Pour déterminer le code à appeler *ASP.NET MVC* utilise la logique de routage d'URL par défaut suivante :
`/[Controller]/[ActionName]/[Parameters]`

<https://docs.microsoft.com/fr-ca/aspnet/mvc/overview/getting-started/introduction/adding-a-controller>

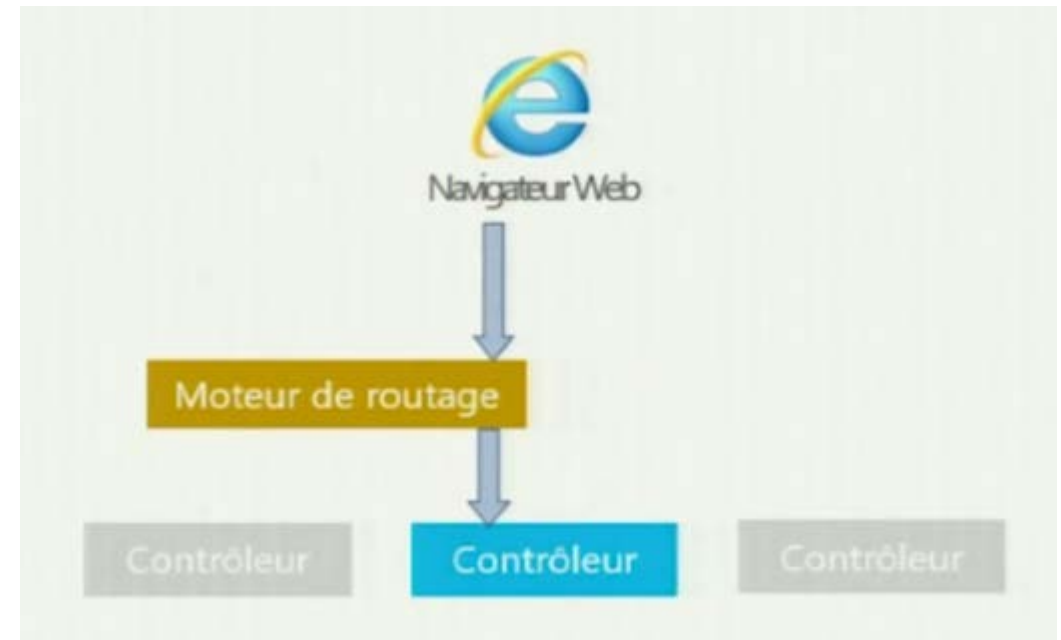


[Réf] <https://channel9.msdn.com/Blogs/Initiation-a-ASPNET-5/02-Les-nouveauts-MVC-6>

Le principe de fonctionnement de l'ASP.NET MVC

- Le routage **ASP.NET MVC** décide comment les URL correspondent aux contrôleurs et aux actions.
- Le **[Front Controller]** utilise un fichier de configuration pour "router" la demande vers le bon contrôleur et la bonne action au sein de ce contrôleur.
- Le **C** de **MVC** en ASP.NET regroupe **[FrontController, Contrôleur, Action]**
- Les **contrôleurs** sont des classes C# héritant de **System.Web.Mvc.Controller**, qui est la classe de base du contrôleur intégré.

<https://dotnet.microsoft.com/apps/aspnet/mvc>



[Réf] <https://channel9.msdn.com/Blogs/Initiation-a-ASPNET-5/02-Les-nouveaux-MVC-6>

Le principe de fonctionnement de l'ASP.NET MVC

Exemple d'un contrôleur

```
public class PeopleController : Controller
{
    private readonly AddressBookContext _context;
    public PeopleController(AddressBookContext context)
    {
        _context = context;
    }

    // GET: /people
    public async Task Index()
    {
        return View(await _context.People.ToListAsync());
    }

    // GET: /people/details/5
    public async Task Details(int id)
    {
        var person = await _context.People.Find(id);

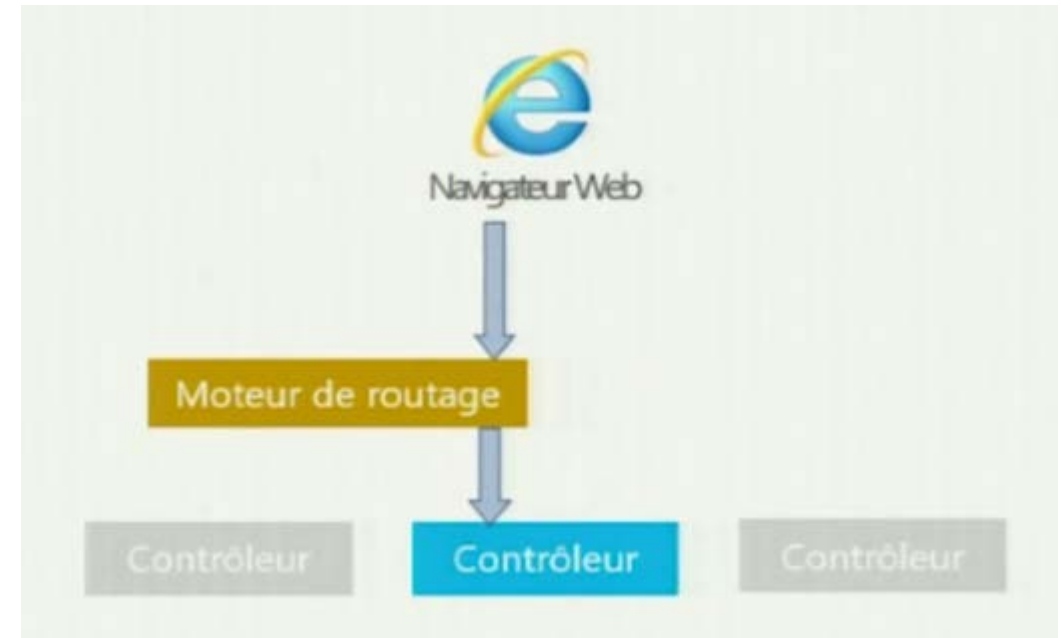
        if (person == null)
        {
            return NotFound();
        }

        return View(person);
    }
}
```

<https://dotnet.microsoft.com/apps/aspnet/mvc>

Exemple d'une URL

People/Details/1



[Réf] <https://channel9.msdn.com/Blogs/Initiation-a-ASPNET-5/02-Les-nouveauts-MVC-6>

Le principe de fonctionnement de l'ASP.NET MVC

- Le contrôleur va lancer une action pour faire des traitements.
- Le contrôleur récupère les résultats des traitements et les met dans le modèle et retourne la vue
- **Le modèle** est une collection de classes qui modélisent les données utilisées dans une application Web. L'action peut récupérer des données stockées en utilisant un modèle.

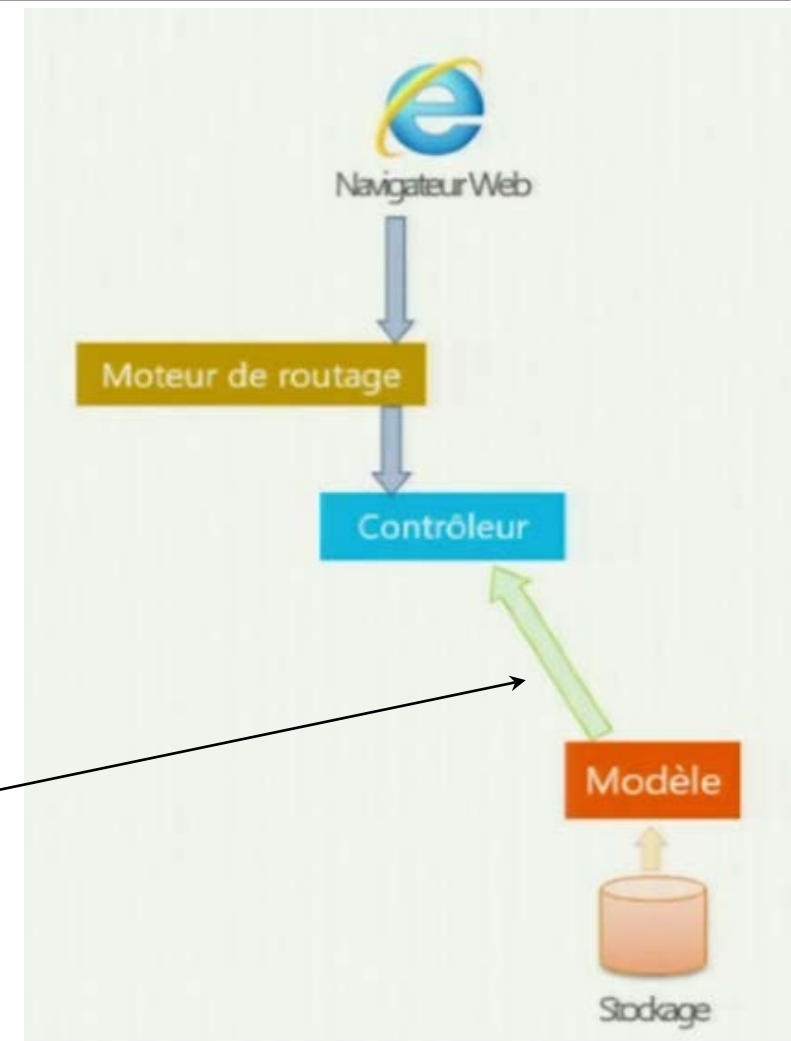
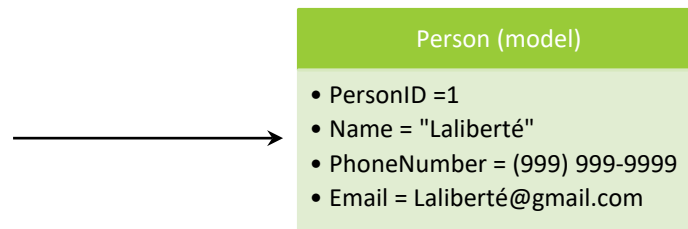
Exemple d'un modèle

```
public class Person
{
    public int PersonId { get; set; }

    [Required]
    [MinLength(2)]
    public string Name { get; set; }

    [Phone]
    public string PhoneNumber { get; set; }

    [EmailAddress]
    public string Email { get; set; }
}
```



<https://dotnet.microsoft.com/apps/aspnet/mvc>

<https://channel9.msdn.com/Blogs/Initiation-a-ASPNET-5/02-Les-nouveauts-MVC-6>

Le principe de fonctionnement de l'ASP.NET MVC

- L'action demande à une certaine vue de s'afficher. Cette vue peut afficher des données récupérées en utilisant un certain modèle.
- Une page ASP.NET, connue comme la **Vue**, est une page **HTML** qui peut contenir du code exécuté côté serveur et qui génère certaines parties de la page.

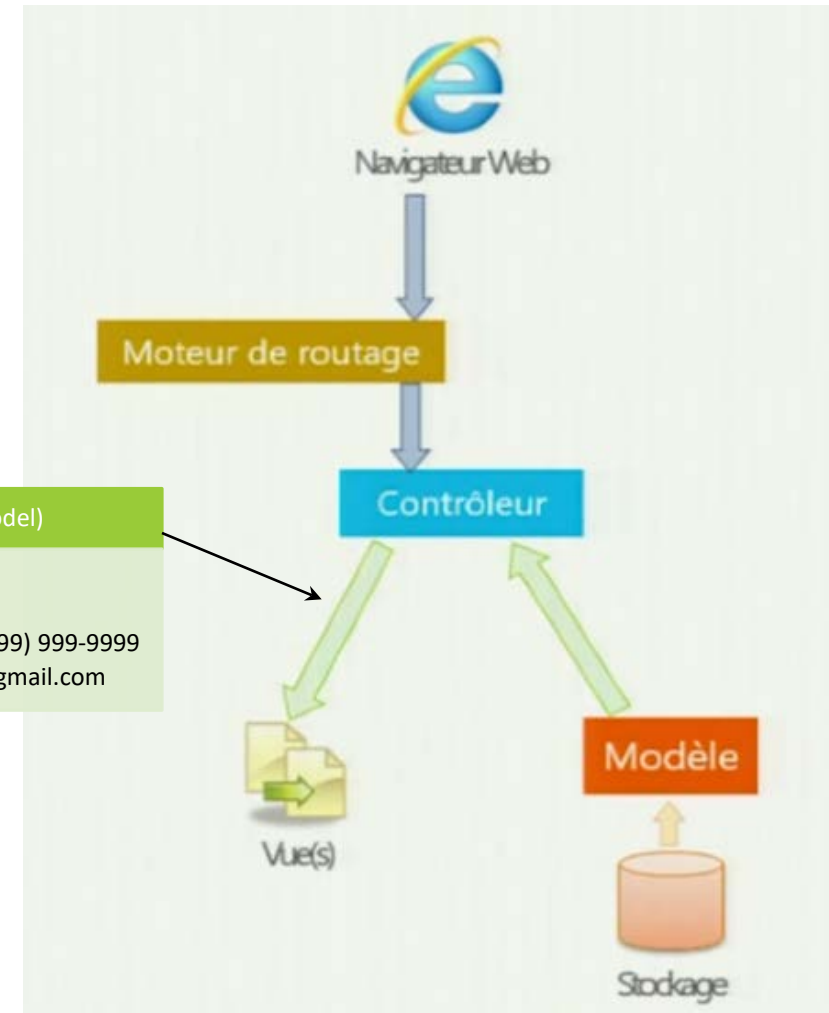
Exemple d'une vue Razor

```
<table class="table">
  <thead>
    <tr>
      <th>@Html.DisplayNameFor(model => model.Name)</th>
      <th>@Html.DisplayNameFor(model => model.PhoneNumber)</th>
      <th>@Html.DisplayNameFor(model => model.Email)</th>
    </tr>
  </thead>
  <tbody>
    @foreach (var item in Model) {
      <tr>
        <td>@Html.DisplayFor(modelItem => item.Name)</td>
        <td>@Html.DisplayFor(modelItem => item.PhoneNumber)</td>
        <td>@Html.DisplayFor(modelItem => item.Email)</td>
      </tr>
    }
  </tbody>
</table>
```

<https://dotnet.microsoft.com/apps/aspnet/mvc>

Person (model)

- PersonID = 1
- Name = "Laliberté"
- PhoneNumber = (999) 999-9999
- Email = Laliberté@gmail.com



<https://channel9.msdn.com/Blogs/Initiation-a-ASPNET-5/02-Les-nouveauts-MVC-6>

Utilisation de l'ASP.NET MVC

Dans le contexte de développement en Visual studio avec C# :

- **Modèle** : le langage C# est utilisé (les fichiers ont une extension .CS).
- **Vue** : Le moteur de vue Razor est utilisé (Combinaison de C# et de HTML - Les fichiers ont une extension .CSHTML).
- **Contrôleur** : le langage C# est utilisé (les fichiers ont une extension .CS).
- **Une action** est simplement une méthode d'une classe de contrôleur.
- **Le routage** est utilisé en ASP.NET MVC pour sélectionner une action. Par défaut, l'URL respecte la forme suivante : **nom du contrôleur/nom de l'action.**
- **La requête va passer par des filtres** : une fois que le serveur sélectionne l'action à appeler, il va vérifier s'il a le droit de l'appeler.

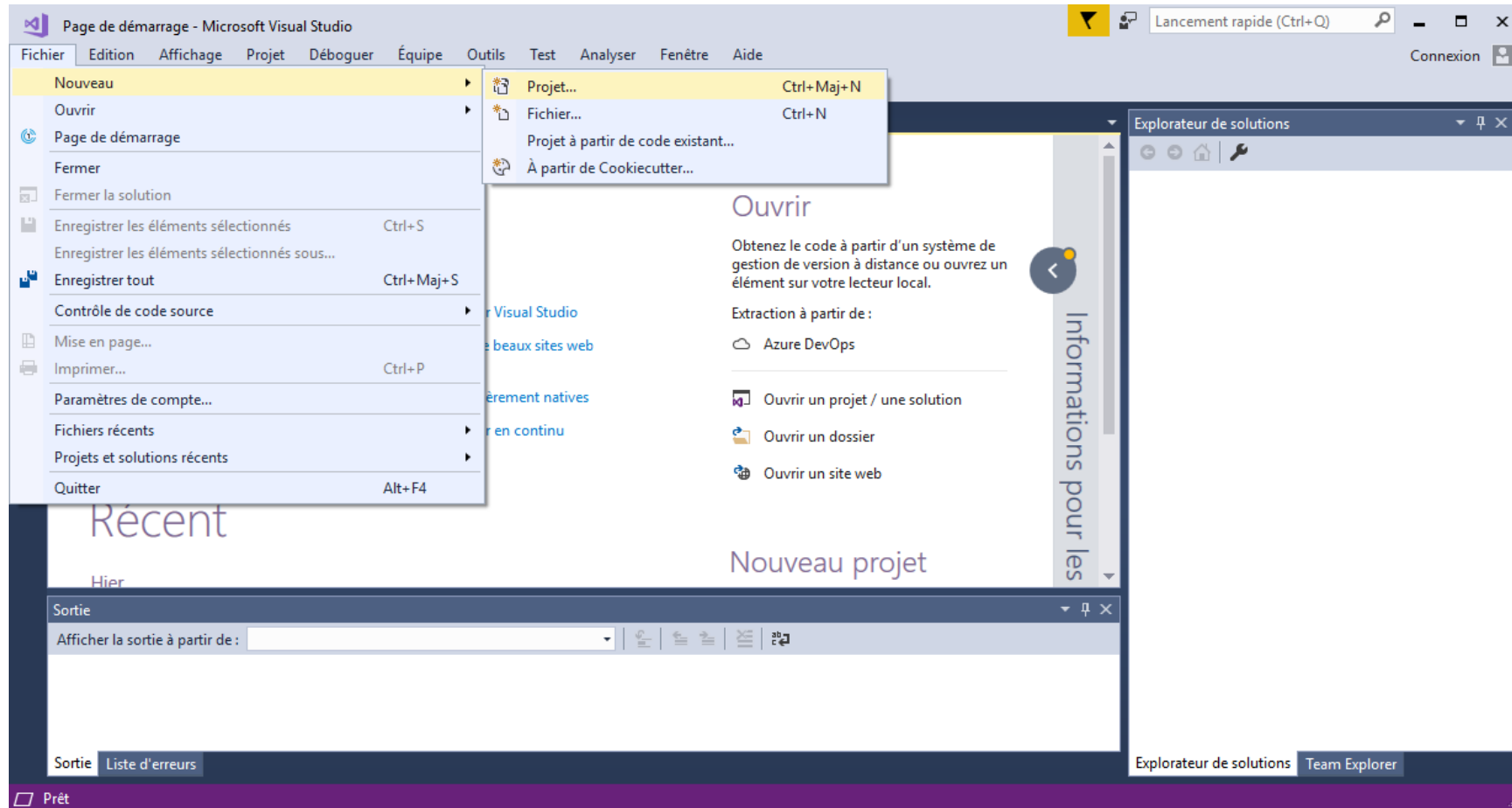
Création d'un projet

Création de projet en ASP.NET MVC

- Projet MVC avec modèle (Template)
- Projet MVC vide

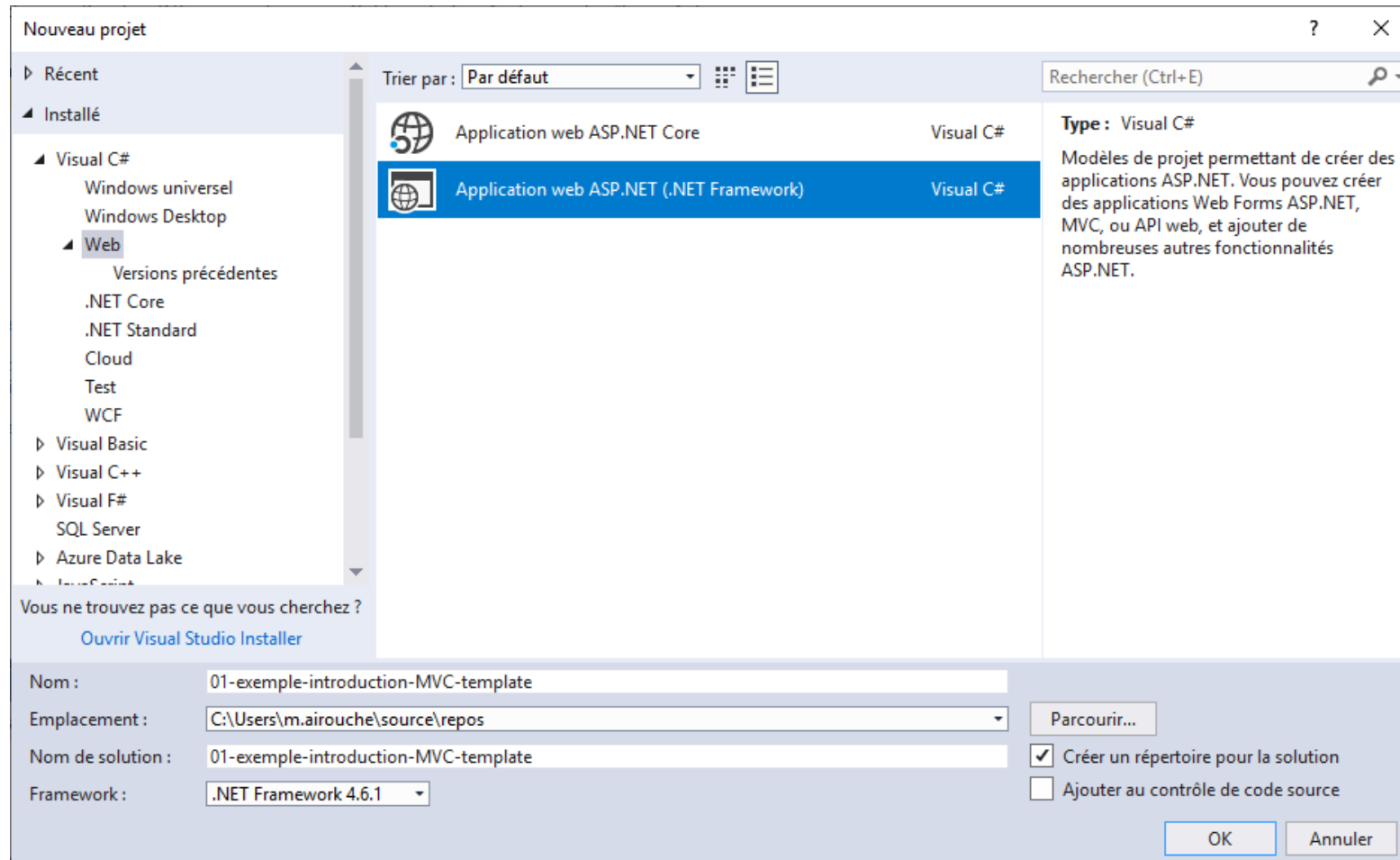
Création d'un projet

La création d'un projet web MVC s'effectue via le menu **Fichier-Nouveau Projet**



Création d'un projet


Sélectionnez (.NET Framework) et nommez le projet : **01-exemple-introduction-MVC-template**





Création d'un projet


Sélectionnez le modèle MVC et l'option MVC


Nouvelle application web ASP.NET - 01-exemple-introduction-MVC-template



Vide


Web Forms


MVC


Web API


Single Page Application


Application API Azure

Modèle de projet pour la création d'applications ASP.NET MVC. ASP.NET MVC vous permet de créer des applications à l'aide de l'architecture Model-View-Controller. ASP.NET MVC comprend de nombreuses fonctionnalités permettant un développement rapide et piloté par des tests pour créer des applications utilisant les dernières normes.

[En savoir plus](#)

Ajoutez des dossiers et des références de base pour :

☐ Web Forms ☒ MVC ☐ API web

☐ Activer la prise en charge de Docker Compose (Nécessite [Docker pour Windows](#))

☐ Ajouter des tests unitaires

Nom du projet de test : 01-exemple-introduction-MVC-template.Tes

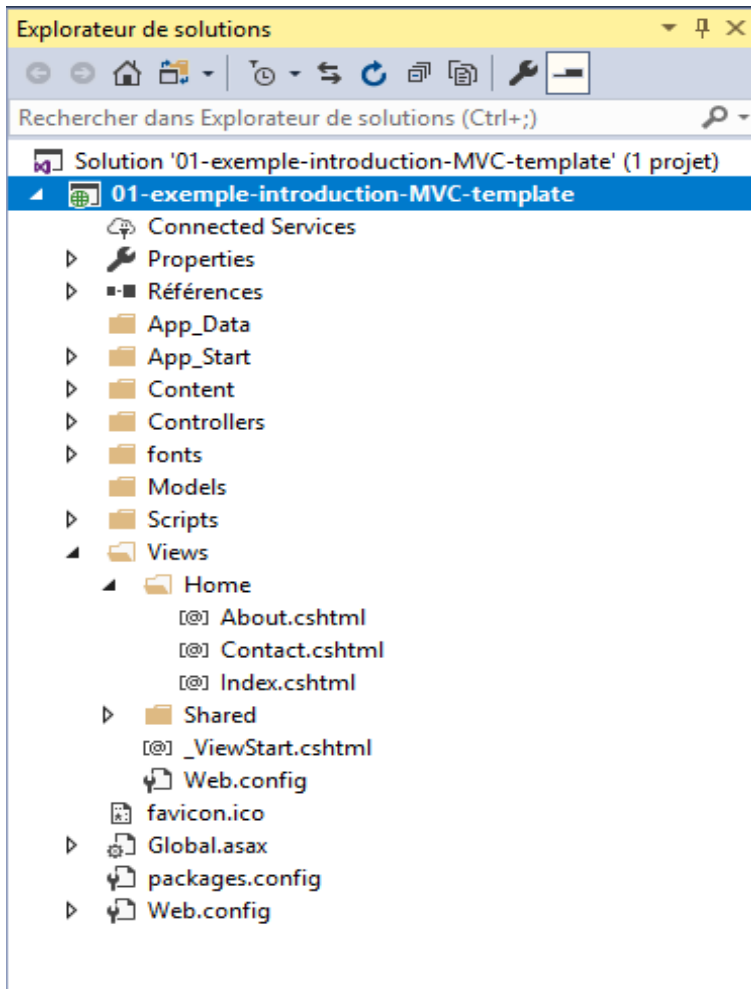
Authentification : **Aucune authentification**

[Modifier l'authentification](#)

OK Annuler

Création d'un projet

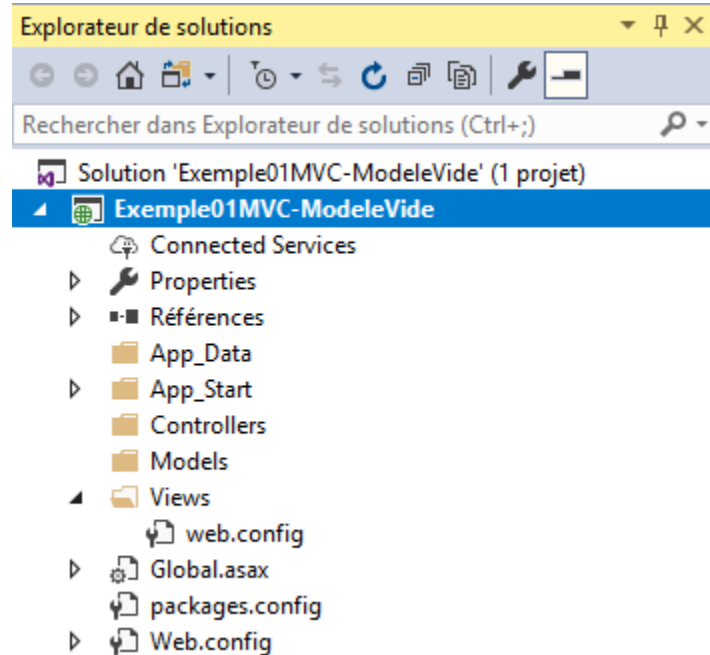
La solution du projet web contient les répertoires et les fichiers suivants : (quand le modèle MVC est choisie)



- **App_Data** : Contient les fichiers de base des données ou les fichiers comme .xml, .txt etc...
- **App_Start** : Contient les fichiers de configuration exécutées au démarrage de l'application
- **Content** : Contient les feuilles de style CSS et autres ressources partagées.
- **Controllers** : Regroupe les contrôleurs destinés à traiter les actions de l'application.
- **Font** : Police de caractères téléchargées par le navigateur.
- **Models** : Rassemble les modèles de l'application
- **Scripts** : Regroupe l'ensemble de modules JavaScript, jQuery et AJAX
- **Views** : Regroupe les Vues (.Cshtml) qui sont les pages Web de l'application
- **[Global.asax]** contient le code exécuté au démarrage de l'application.
- **[web.config]** est le fichier de configuration de l'application Web
- **[Layout]** définit la vue parente de toutes les vues. Elle correspond à la page maître du framework ASP.NET classique.

Création d'un projet

La solution du projet web contient les répertoires et les fichiers suivants : (quand le modèle vide est choisie)



- **App_Data** : Contient les fichiers de base des données ou les fichiers comme .xml, .txt etc...
- **App_Start** : Contient les fichiers de configuration exécutées au démarrage de l'application
- **Controllers** : Regroupe les contrôleurs destinés à traiter les actions de l'application.
- **Models** : Rassemble les modèles de l'application
- **Views** : Regroupe les Vues (.Cshtml) qui sont les pages Web de l'application
- **Global.asax** contient le code exécuté au démarrage de l'application.
- **Package.config** contient les informations de package à utiliser lors de déploiement de l'application
- **[web.config]** est le fichier de configuration de l'application Web