

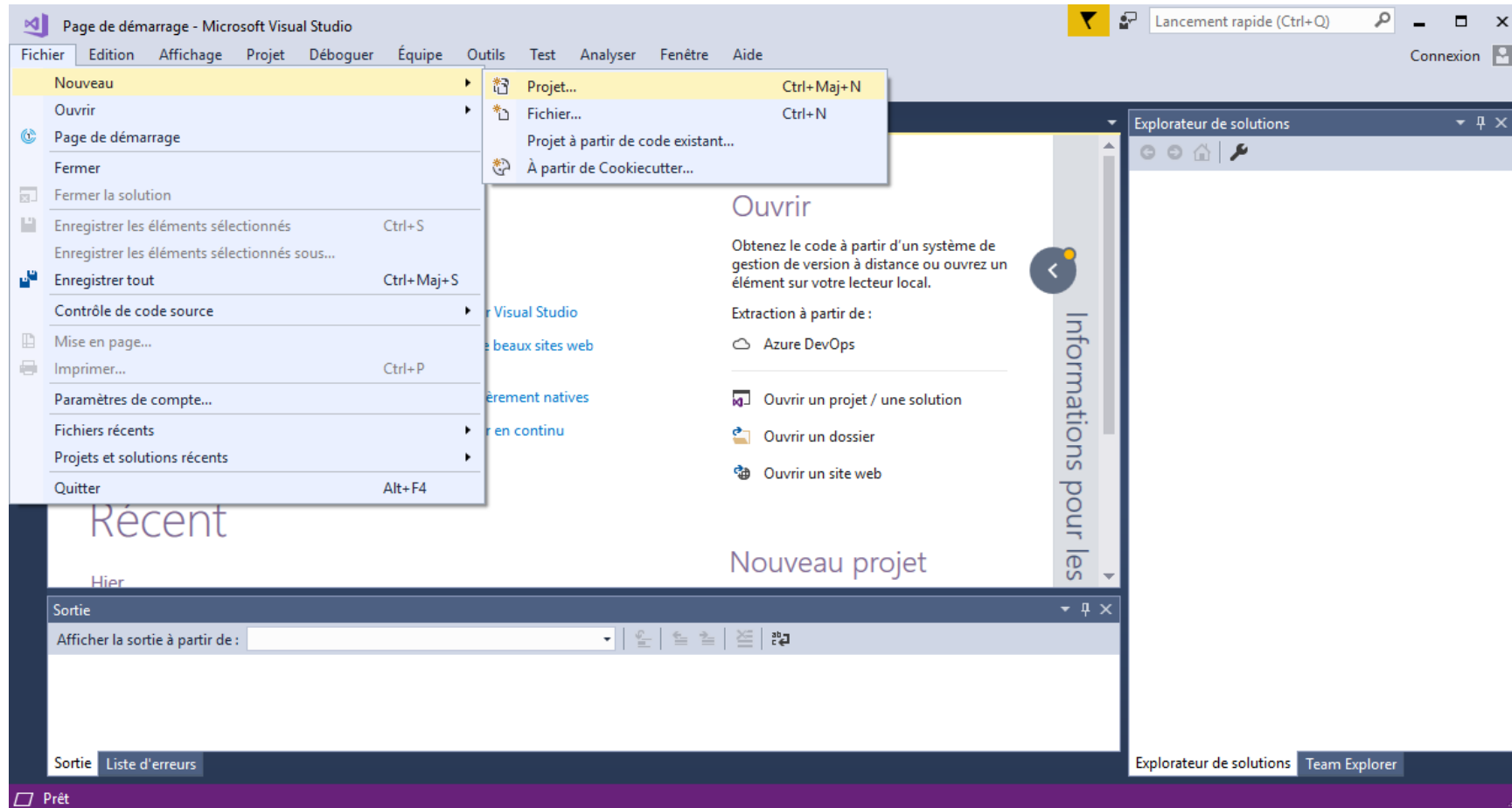
# Les contrôleurs, les méthodes d'action, les vues et le routage en ASP.NET MVC

---

MOHAMED AIROUCHE

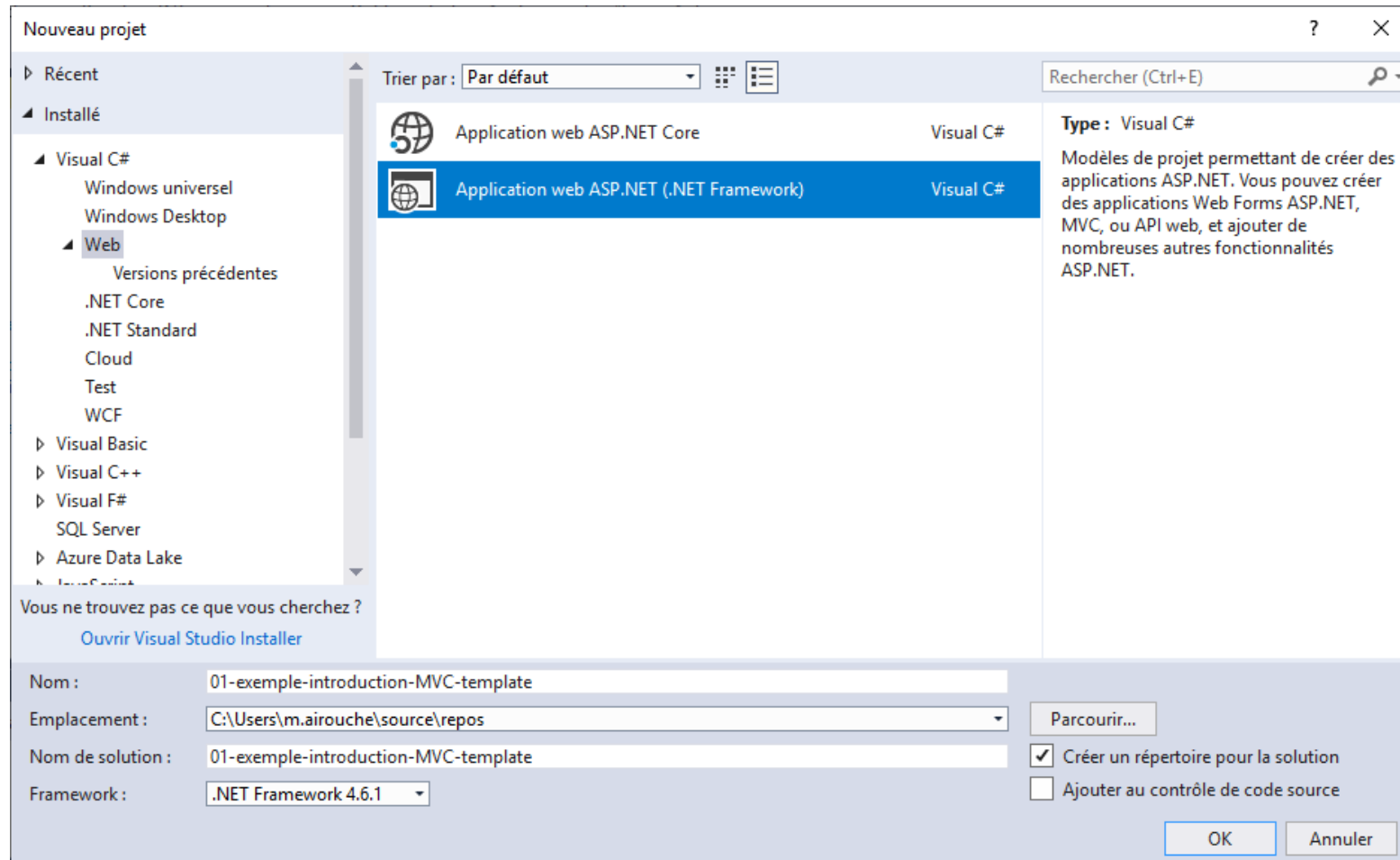
# Création d'un projet

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

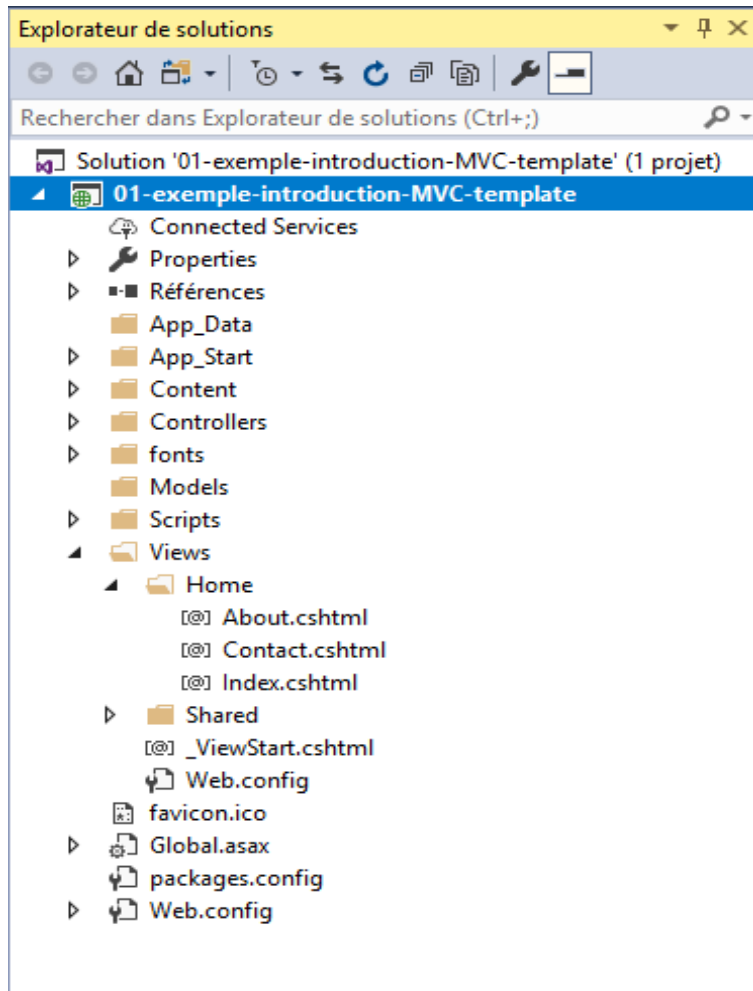
Authentification : **Aucune authentification**

[Modifier l'authentification](#)

OK Annuler

# Création d'un projet

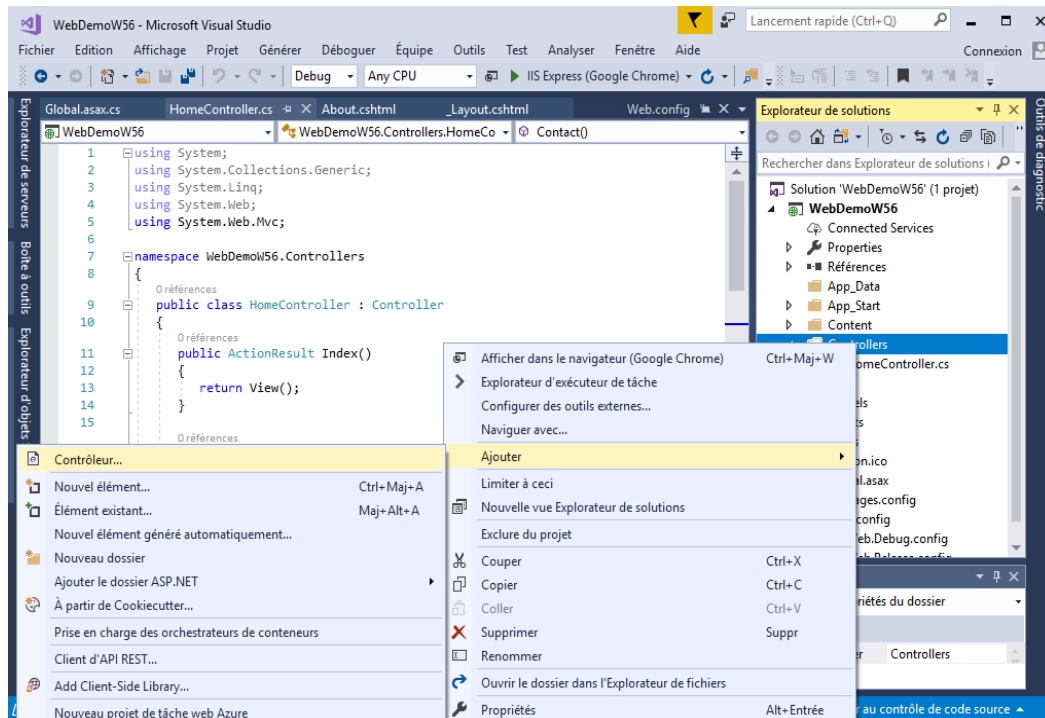
La solution du projet web contient les répertoires et les fichiers suivants : (quand le modèle MVC est choisie)



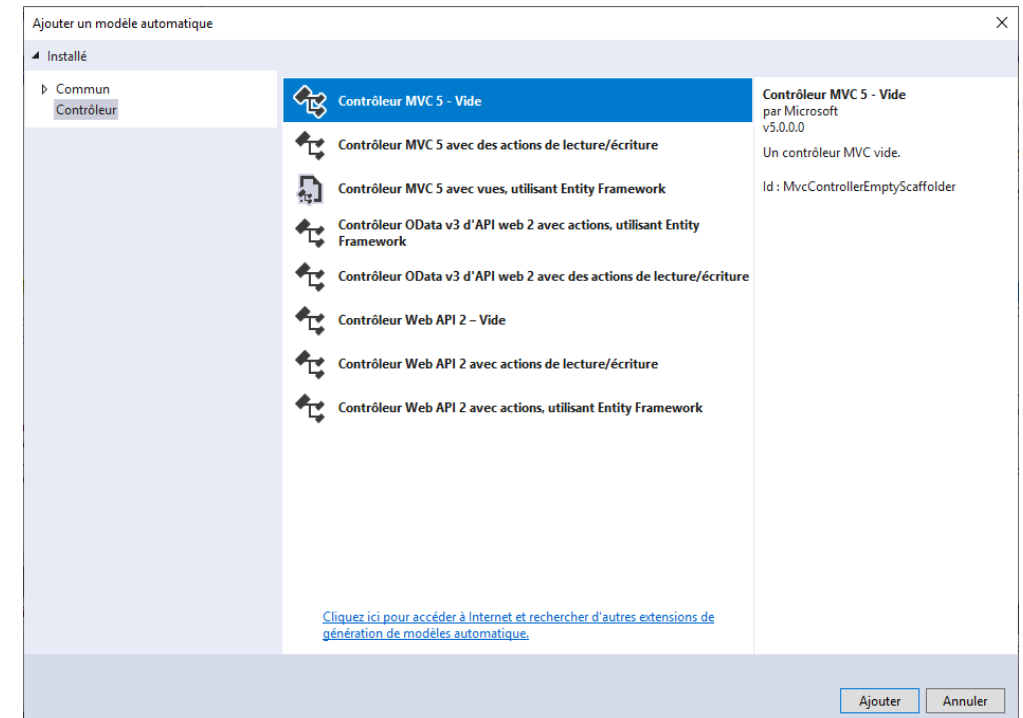
- **App\_Data** : Contient les fichiers de base des données ou les fichiers comme .xml, .txt etc...
- **App\_Start** : Contient les fichiers de configuration exécutées au démarrage de l'application
- **Content** : Contient les feuilles de style CSS et autres ressources partagées.
- **Controllers** : 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.

# Ajout d'un contrôleur au projet

- Dans Explorateur de solutions, cliquez avec le bouton droit sur le dossier Controllers
- Puis cliquez sur Ajouter, puis sur Contrôleur.



- Dans la boîte de dialogue **Ajout un modèle automatique**, cliquez sur Contrôleur MVC 5 – Vide, puis cliquez sur Ajouter.

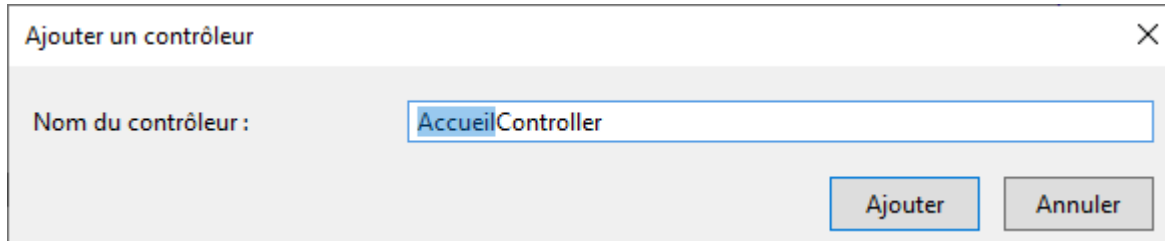


<https://docs.microsoft.com/fr-ca/aspnet/mvc/overview/getting-started/introduction/adding-a-controller>

# Ajout d'un contrôleur au projet

---

Dans la boîte de dialogue **Ajouter Contrôleur**, entrez le nom `AccueilController`



Ajouter un contrôleur

Nom du contrôleur :

Ajouter Annuler

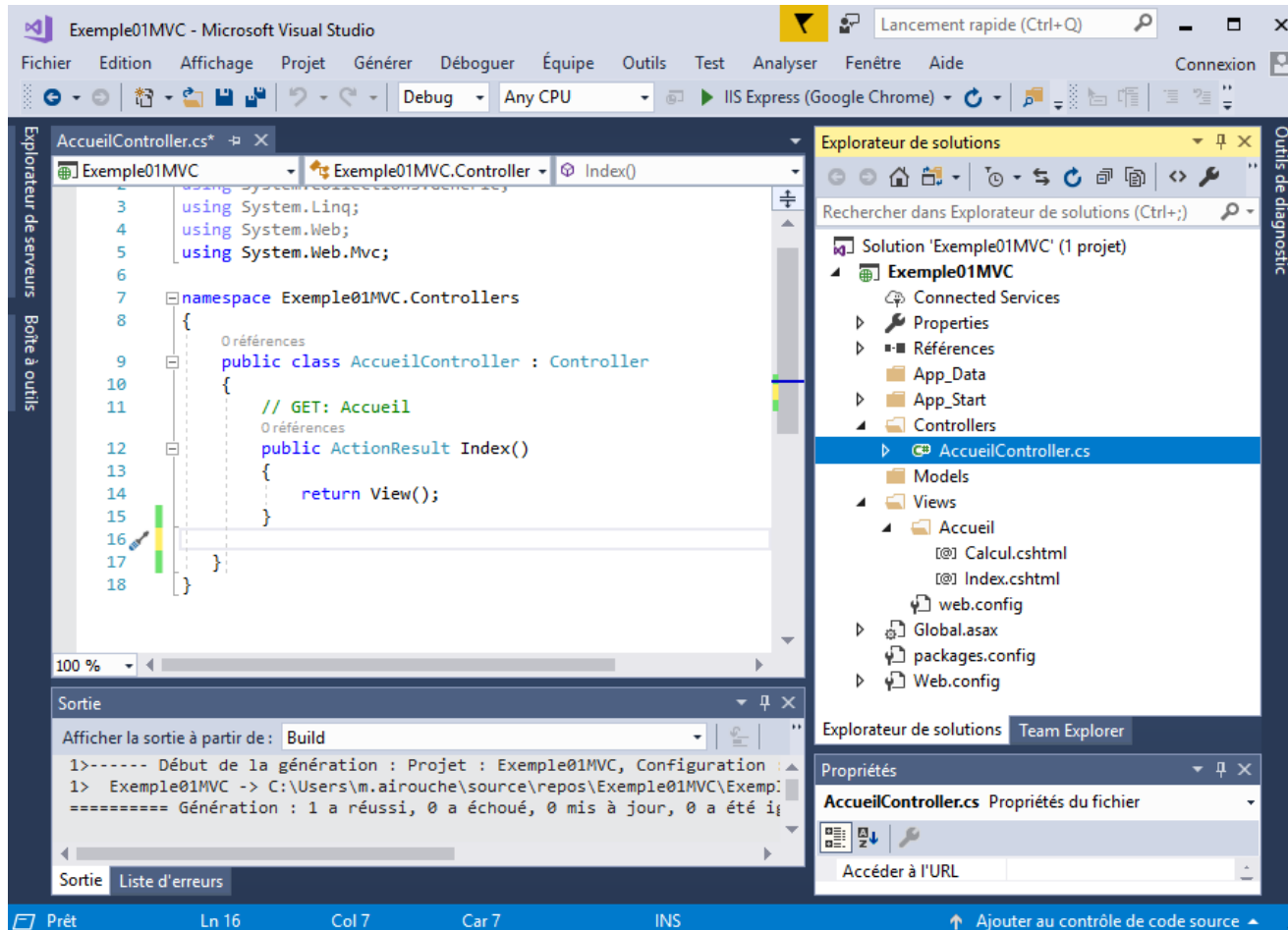
Pour plus de détails:

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

<https://www.tutorialsteacher.com/mvc/mvc-controller>

# Ajout d'un contrôleur au projet

Le Contrôleur AccueilController est ajouté au projet avec une méthode Index()





# Les méthodes d'action en ASP.NET MVC

---

Toutes les méthodes publiques d'une classe Controller sont appelées méthodes d'action. Elles ont les restrictions suivantes:

- La méthode d'action doit être publique. Il ne peut pas être privé ou protégé.
- La méthode d'action ne peut pas être surchargée.
- La méthode d'action ne peut pas être une méthode statique.

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**.

Pour plus de détails:

<https://www.tutorialsteacher.com/mvc/action-method-in-mvc>

<https://sergetahe.com/cours-tutoriels-de-programmation/wp-content/uploads/documents/Introduction%20%c3%a0%20ASP.NET%20MVC%20par%20l'exemple%20-%202013.pdf>

# Les méthodes d'action en ASP.NET MVC

---

Le Framework MVC inclut diverses classes de résultats, qui peuvent être renvoyées à partir d'une méthode d'action. Ces classes de résultats représentent différents types de réponses telles que HTML, fichier, chaîne, Json, Javascript, etc.

La classe **ActionResult** est une classe de base de toutes les classes de résultats qui peuvent être retournés par une méthode d'action.

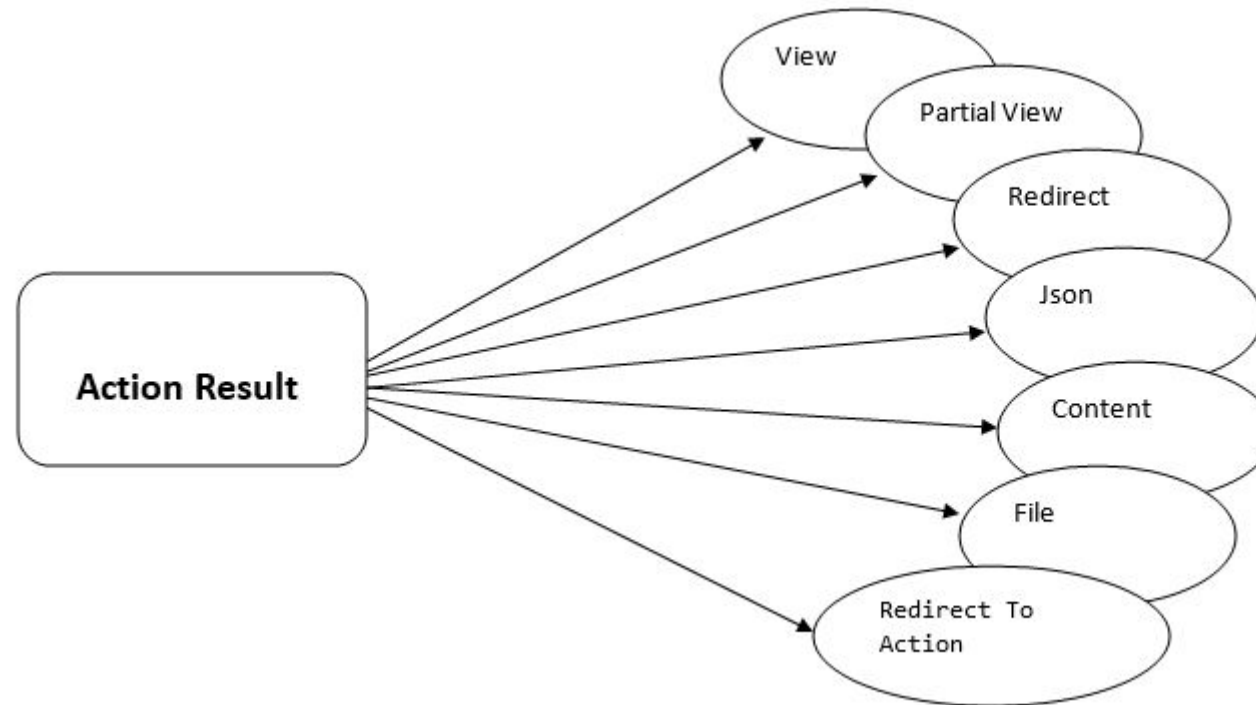
Vous pouvez donc renvoyer le type de méthode d'action qui renvoie tout type de résultat qui hérite de la classe de base **ActionResult**.

Vous pouvez spécifier la classe de résultat appropriée en tant que type de retour de méthode d'action.

# Les méthodes d'action en ASP.NET MVC

---

La figure suivante montre quelques types qui héritent de la classe ActionResult :



# Les méthodes d'action en ASP.NET MVC

---

Le tableau suivant montre les détails des types retournés par les méthodes de Contrôleur :

Les méthodes d'action	
<b>ContentResult</b>	Renvoie une chaîne de caractère
<b>FileContentResult</b>	Retourne le contenu du fichier
<b>RedirectToResult</b>	Redirige vers l'URL spécifiée
<b>RedirectToRouteResult</b>	Redirige vers différentes actions / différentes actions du contrôleur
<b>JsonResult</b>	Renvoie des données au format JSON
<b>EmptyResult</b>	Ne retourne rien
<b>ViewResult</b>	Reçu en réponse pour le moteur de vue
<b>PartialViewResult</b>	Reçu en réponse pour le moteur de vue

<https://www.tutorialsteacher.com/mvc/action-method-in-mvc>

# Les méthodes d'action en ASP.NET MVC

---

## Paramètres de méthode d'action

Toutes les méthodes d'action peuvent avoir des paramètres d'entrée comme méthodes normales. Il peut s'agir de paramètres de type de données primitif ou de type complexe.

<https://www.tutorialsteacher.com/mvc/action-method-in-mvc>

@\*

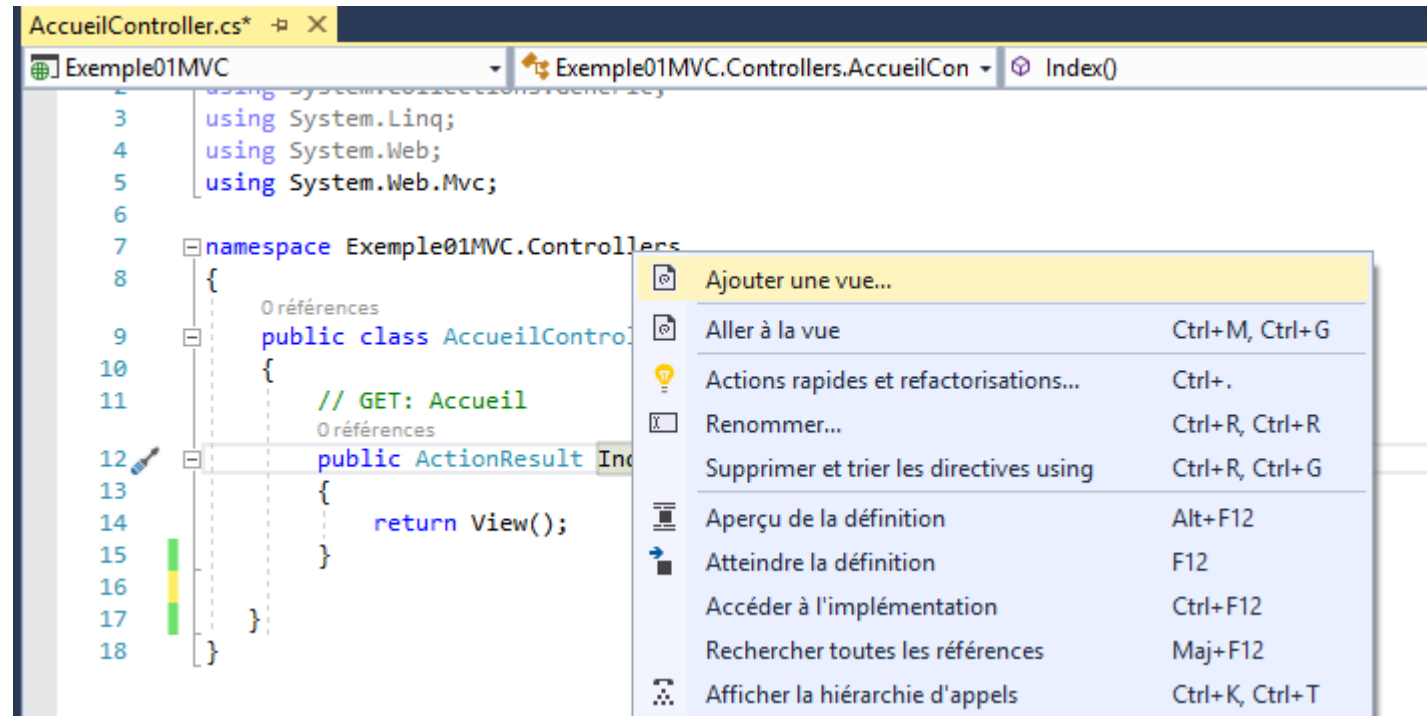
Exemple :

\*@

```
public string DemoParametres(string id)
{
    return "La valeur de l'id est : "+id;
}
public string DemoParametres2(string id, string id2)
{
    return "La valeur de l'id est : " + id+ "      La valeur de l'id est : " + id2;
}
```

# Ajout d'une vue au projet ASP.NET

Cliquez droit sur l'action Index pour ajouter la vue correspondante:



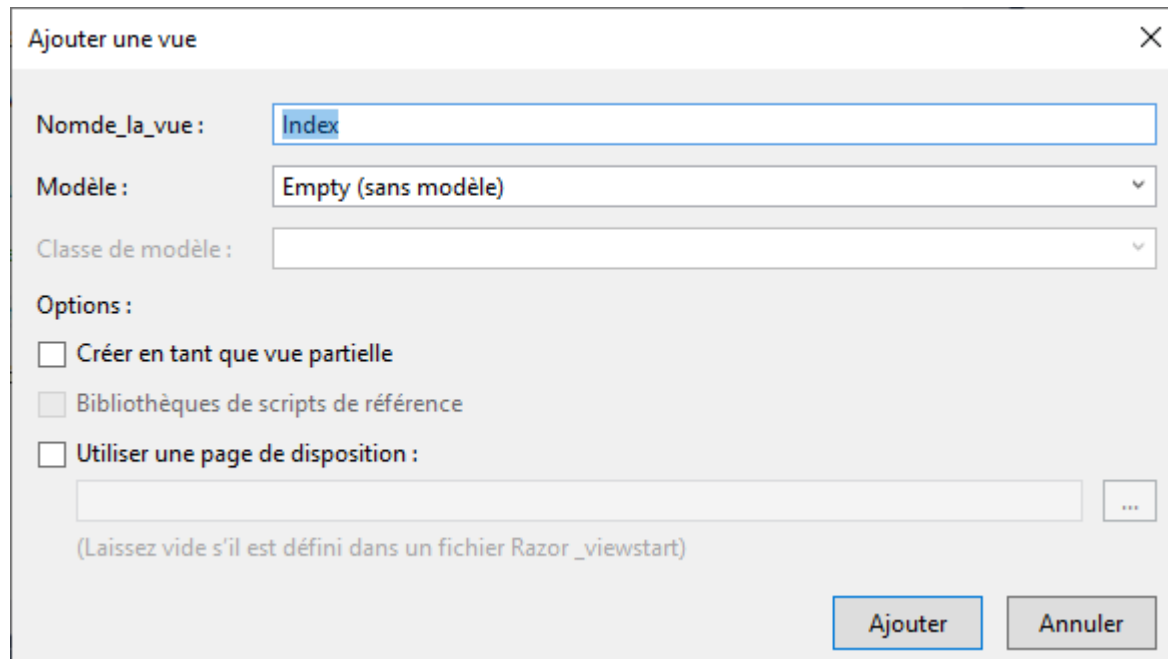
<https://www.tutorialsteacher.com/mvc/mvc-view>

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

# Ajout d'une vue au projet ASP.NET

---

Une fenêtre vous propose de choisir les paramètres de la vue. Acceptez les paramètres par défaut comme le montre la figure suivante et cliquez sur Ajoutez :



Ajouter une vue

Nom de la vue :

Modèle :

Classe de modèle :

Options :

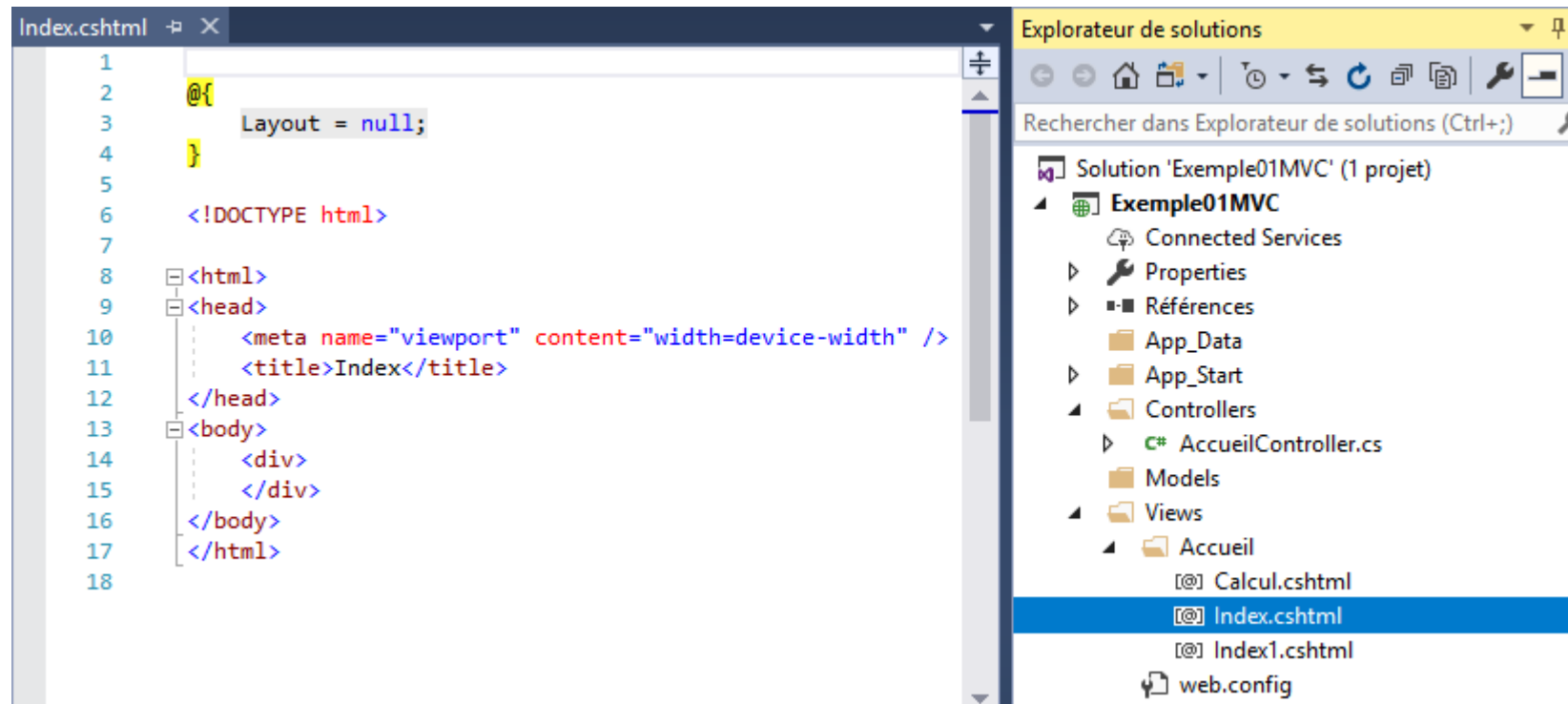
- ☐ Créer en tant que vue partielle
- ☐ Bibliothèques de scripts de référence
- ☐ Utiliser une page de disposition :

...

(Laissez vide s'il est défini dans un fichier Razor \_viewstart)

# Ajout d'une vue au projet ASP.NET

La vue est créée dans le dossier Views/Accueil avec le nom Index.cshtml:





# Le routage en ASP.NET MVC

---

Le routage est un sujet très important en développement web. En effet, les routes sont la base de la navigation pour un site web, et pour tout l'Internet de manière générale.

Le routage est le processus consistant à diriger une demande HTTP vers une action d'un contrôleur et la fonctionnalité de ce traitement est implémentée dans **System.Web.Routing**.

**System.Web.Routing** est utilisé par le Framework MVC. Le Framework MVC utilise le routage pour diriger une demande vers une action d'un contrôleur.

Le fichier RouteConfig.cs dans le dossier App\_start est la partie de votre application, dans laquelle vous définissez l'itinéraire pour votre application.

Le fichier **RouteConfig.cs** est référencé dans le code de l'événement de démarrage de l'application dans Global.asax.cs dans l'application MVC.

Le [**Front Controller**] devait router une URL vers l'action chargée de la traiter. Une route sert à faire le lien entre un modèle d'URL et une action. Ces routes sont définies dans le dossier [App\_Start] du projet par les classes [FilterConfig, RouteConfig, BundleConfig] :

<https://www.tutorialsteacher.com/mvc/routing-in-mvc>

# Le routage en ASP.NET MVC

---

➤ Voici le fichier Global.asax.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Optimization;
using System.Web.Routing;

namespace _01_exemple_introduction_MVC_template
{
    public class MvcApplication : System.Web.HttpApplication
    {
        protected void Application_Start()
        {
            AreaRegistration.RegisterAllAreas();
            FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);
        }
    }
}
```

<https://www.tutorialsteacher.com/mvc/routing-in-mvc>

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

# Le routage en ASP.NET MVC

---

➤ Voici le fichier RouteConfig.cs dans le dossier App\_start

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Routing;

namespace Exemple01MVC
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

            routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{id}",
                defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
            );
        }
    }
}
```

<https://www.tutorialsteacher.com/mvc/routing-in-mvc>

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

## Exercise:

Ajouter une vue au projet ASP.NET MVC qui permet d'afficher la page suivante :

