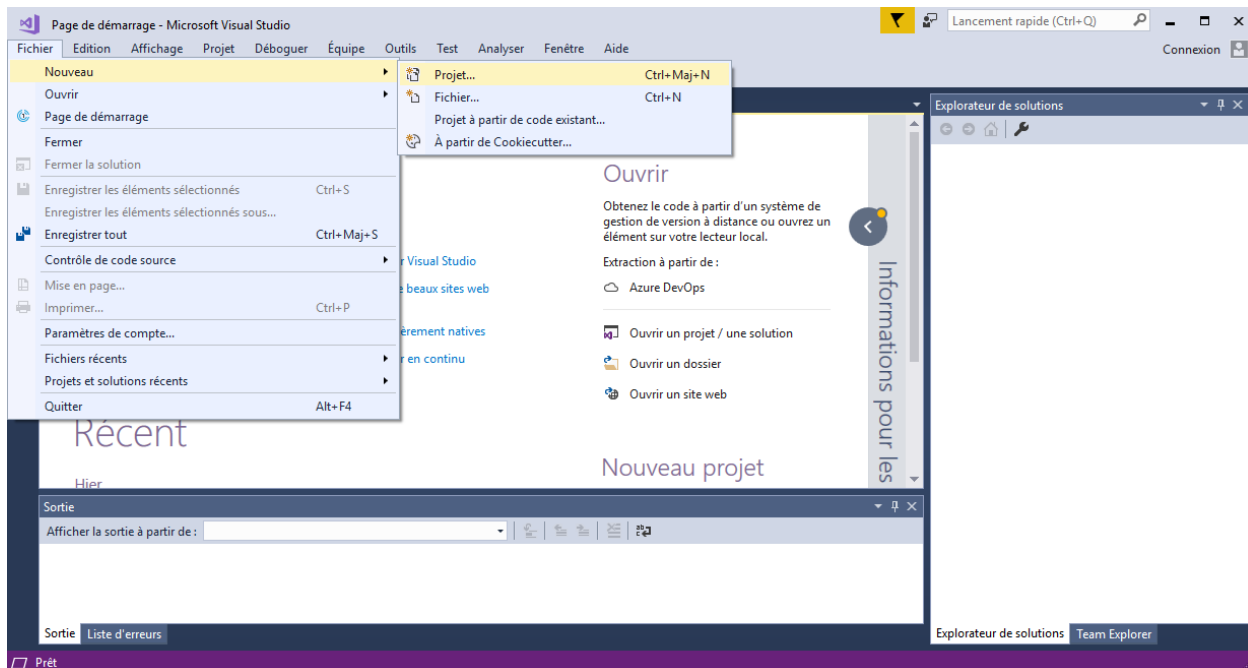


SÉRIE D'EXERCICES 1 (PARTIE 1)

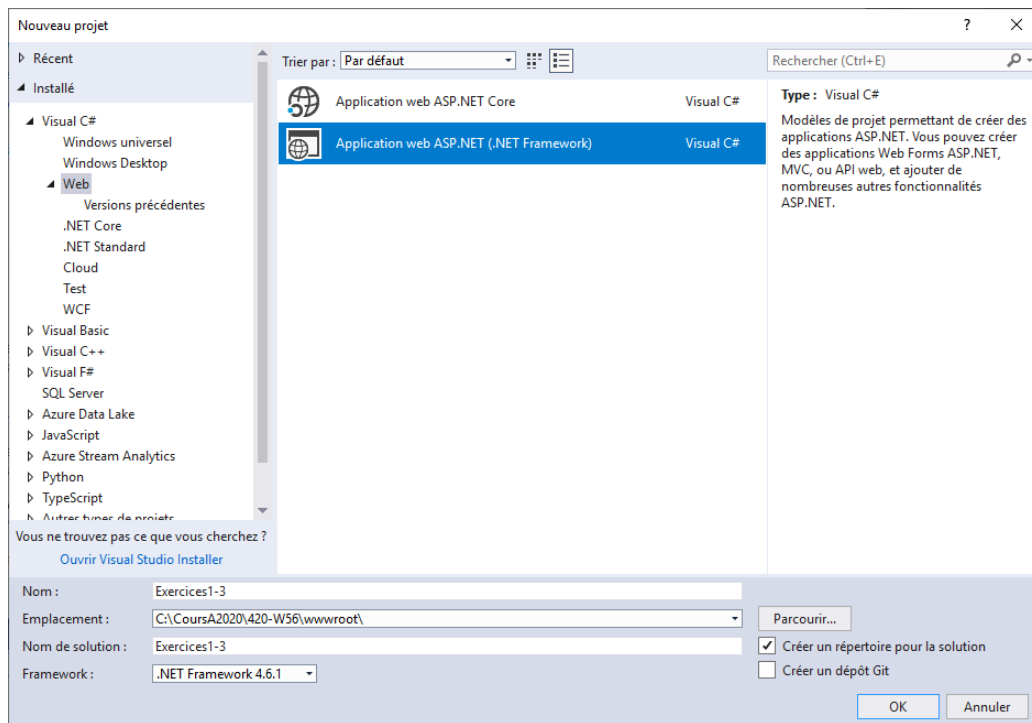
Premier contact avec l'environnement ASP.NET MVC

Pour commencer

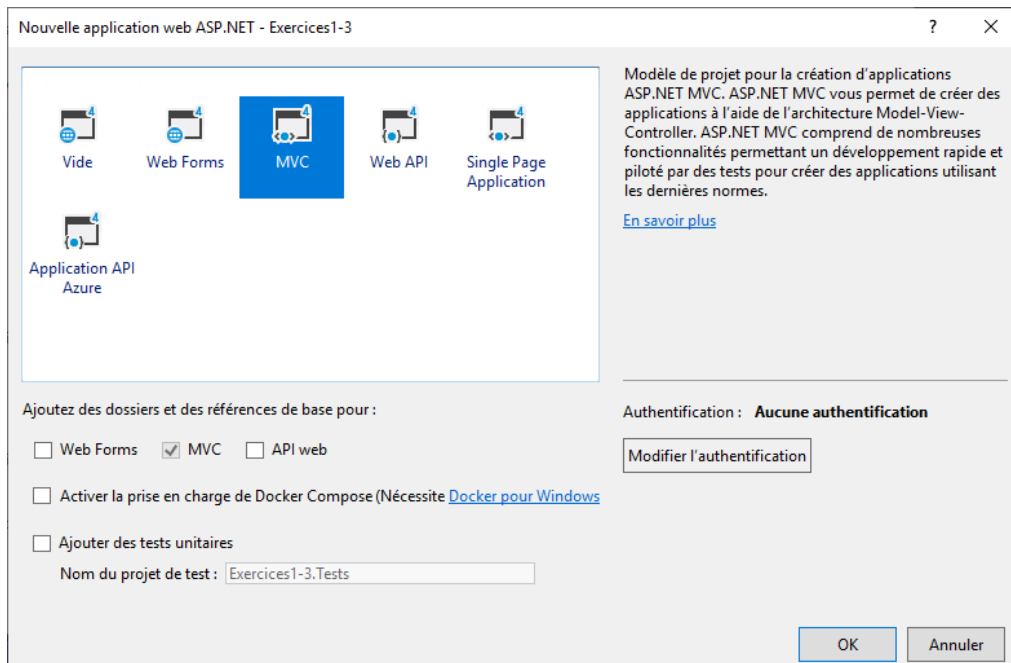
1. Créez un dossier **exercices-W56**.
2. Lancez Visual Studio 2017.
3. Créez un projet web MVC en utilisant le menu **Fichier->Nouveau->Projet**. Comme le montre la figure suivante :



4. Sélectionnez le modèle de projet application Web ASP.NET (.NET Framework) et nommez le projet : **Exercices1-3** et Cliquez sur OK comme le montre la figure suivante :

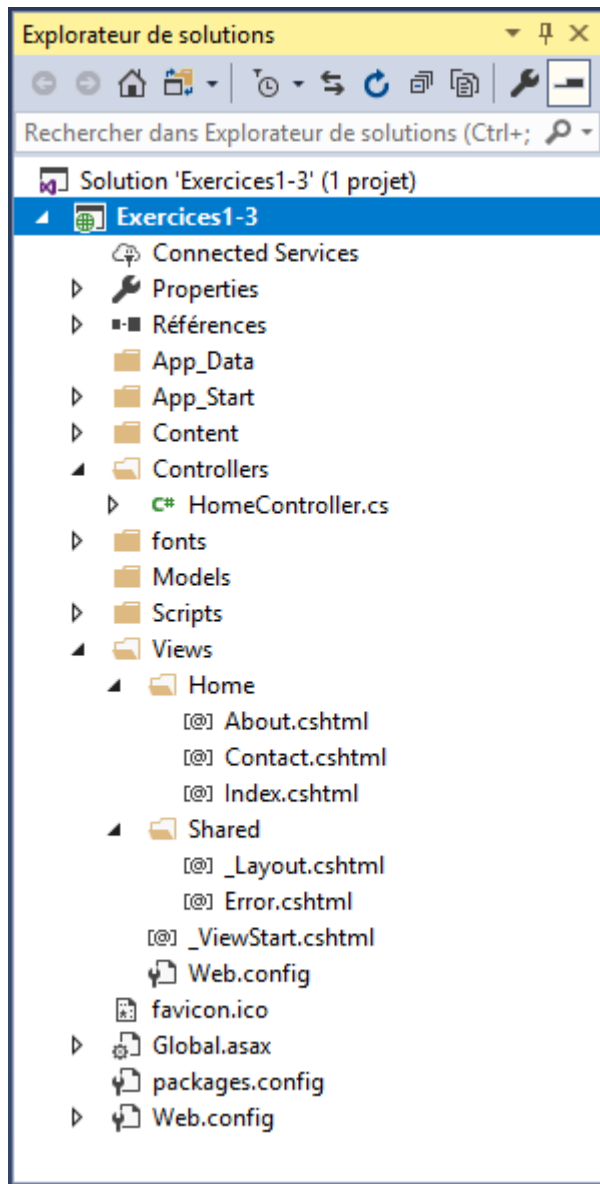


5. Sélectionnez dans la boîte de dialogue **nouvelle application Web ASP.net** le modèle MVC et l'option MVC et Cliquez sur OK.



Visual Studio a utilisé le modèle par défaut pour le projet MVC que vous venez de créer. Vous disposez maintenant d'une application fonctionnelle en entrant un nom de projet et en sélectionnant quelques options. Il s'agit d'un projet de démarrage de base.

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



- **App_Data** : Contient les fichiers de base des données ou les fichiers comme .xml, .txt etc...
- **App_Start** : Contient les fichiers de configuration exécutés 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** : Polices 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.

Les conventions utilisées dans une application ASP.NET MVC :

ASP.NET MVC utilise le principe de la convention sur la configuration (ou le codage par convention). Ceci est un terme qui, lorsqu'une convention, ou un modèle de code, est implémenté par un outil pour correspondre au comportement souhaité, se comporte comme prévu sans l'utilisation de fichiers de configuration. Cela facilite votre codage lorsque vous connaissez les conventions de l'emplacement de tout et que vous placez chaque composant à son emplacement correct.

Voici une liste rapide des conventions simples implémentées dans ASP.NET MVC:

- La structure du projet reflète son modèle MVC :
 - Les contrôleurs C seront placés dans le dossier [Controllers],
 - Les modèles de données M seront placés dans le dossier [Models],
 - Les vues V seront placées dans le dossier [Views],
- Lorsque vous créez un nouveau contrôleur, par défaut, il place le mot «Controller» comme suffixe sur votre nom de contrôleur. Tous les contrôleurs ont Controller comme suffixe ajouté.
- Il doit y avoir au moins une méthode effectuant le GET et une méthode facultative effectuant le POST si vous renvoyez des données au contrôleur.
- Chaque page de vue (HTML) se trouve dans un dossier nommé d'après le nom du contrôleur. Sur la base de l'exemple par défaut avec Visual studio, nous avons un HomeController (avec trois actions : Index, About et Contact), et sous le répertoire Views, vous verrez un dossier appelé Home avec trois fichiers (About.chhtml, Contact.cshtml, Index.cshtml) dans ce dossier.
- Si vous avez un fichier .cshtml commun (ou une vue partielle comme nous l'appelons dans MVC), il peut être placé dans le dossier Shared situé sous le dossier Views.
- Il y a toujours un _Layout.cshtml et ceci est considéré comme votre page maître et se trouve dans le dossier Views / Shared.
- Une route sert à faire le lien entre un modèle d'URL et une action de contrôleur. Ces routes sont définies dans le dossier [App_Start] du projet par les classes [FilterConfig, RouteConfig, BundleConfig].
- Il peut y avoir plusieurs routes possibles (= plusieurs modèles d'URL possibles). Elles se distinguent entre-elles par leur nom. La forme des URL de la route par défaut aura trois composantes : {controller}/{action}/{id}
 - {controller} : le nom d'une classe dérivée de [Controller]. Elle sera cherchée dans le dossier [Controllers] du projet. Par convention, si l'URL est /X/Y/Z, le contrôleur chargé de traiter cette URL sera la classe XController. Le suffixe Controller est ajouté au nom du contrôleur présent dans l'URL ; Y est le nom de l'action à exécuter.
 - {action} : le nom d'une méthode dans le contrôleur désigné ci-dessus. C'est elle qui va recevoir les paramètres qui accompagnent l'URL et qui va les traiter.
 - {id} : un paramètre qui sera transmis à l'action. Pour cela l'action devra avoir un paramètre nommé id.

- Des valeurs par défaut lorsque l'URL n'a pas la forme attendue `/controller/action/id`. Elle indique également que le paramètre `{id}` dans l'URL est facultatif. Voici une liste d'URL incomplètes et l'URL complétée avec les valeurs par défaut :

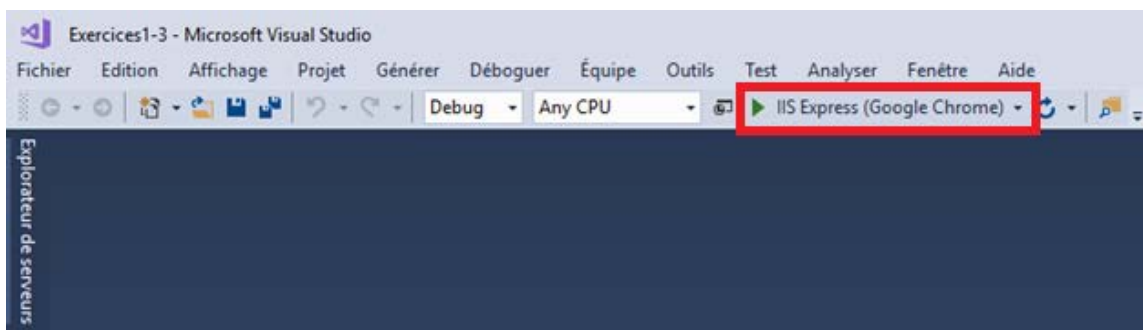
URL d'origine	URL complétée
/	/Home/Index
/Do	/Do/Index
/Do/Something	/Do/Something
/Do/Something/4	/Do/Something/4
/Do/Something/x/y/z	URL non routée

Nous tenons pour acquis certaines conventions lors du codage. Cela devient juste une seconde nature pour eux sans même y penser.

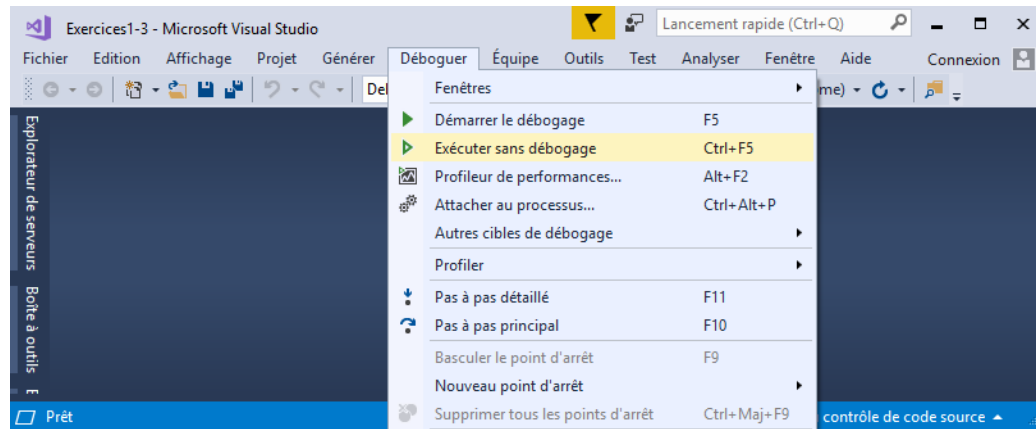
L'exécution d'une application ASP.NET MVC :

Appuyez sur F5 pour démarrer le débogage ou appuyez sur **Ctrl-F5** pour exécuter l'application en mode non-débogage.

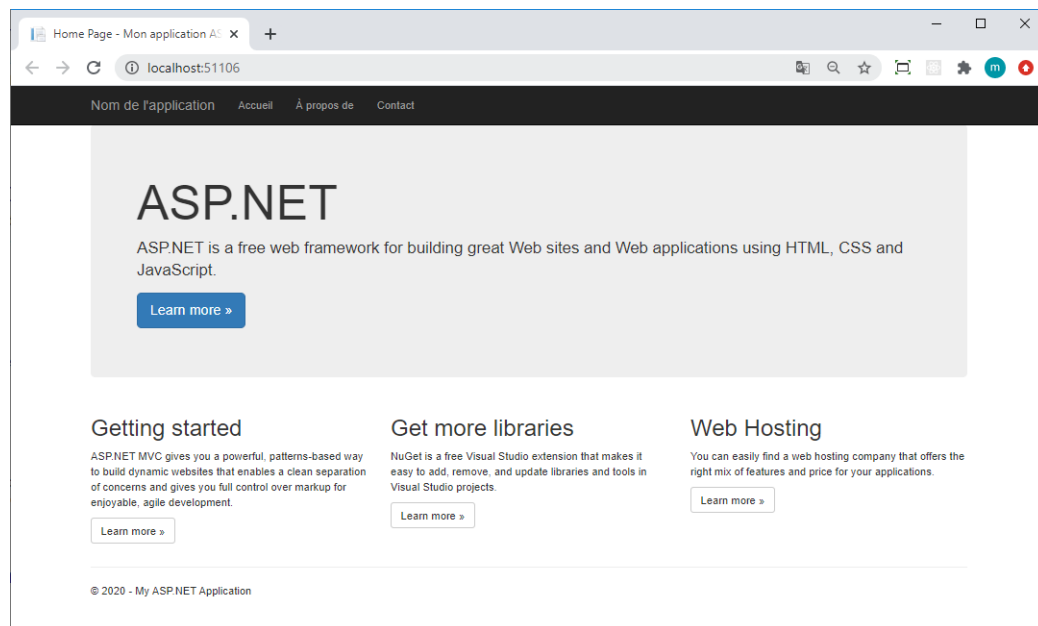
- Visual Studio démarre IIS Express et exécute l'application. Notez que la barre d'adresse affiche `localhost:port#`, et non quelque chose comme `exemple.com`. En effet, `localhost` est le nom d'hôte standard de votre ordinateur local. Quand Visual Studio crée un projet web, un port aléatoire est utilisé pour le serveur web.
- Si vous lancez l'application avec **Ctrl+F5** (mode sans débogage), vous pouvez apporter des modifications au code, enregistrer le fichier, actualiser le navigateur et examiner les modifications apportées au code. Beaucoup de développeurs préfèrent utiliser ce mode pour lancer rapidement l'application et voir les modifications.
- Vous pouvez déboguer l'application en sélectionnant le bouton **IIS Express**



- Vous pouvez lancer l'application en mode débogage ou non-débogage à partir de l'élément de menu **Déboguer** :



- La figure suivante montre la page affichée après l'exécution de l'application :



Pour plus de détails sur la création d'un projet en ASP.NET MVC:
<https://docs.microsoft.com/fr-fr/aspnet/mvc/overview/getting-started/introduction/getting-started>

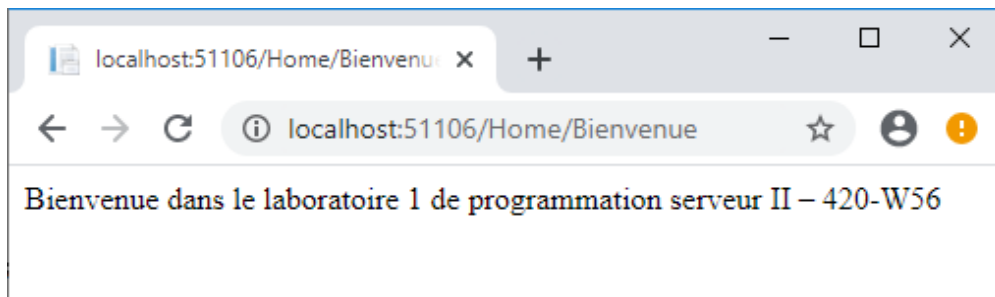
Utilisation des pages maitre :
<https://www.tutorialsteacher.com/mvc/layout-view-in-asp.net-mvc>

Travail à effectuer :

1. Ajoutez l'action suivante au contrôleur Home qui retourne une chaîne de caractère (string) "Bienvenue dans le laboratoire 1 de programmation serveur II – 420-W56" :

```
public string Bienvenue()
{
    return "Bienvenue dans le laboratoire 1 de programmation serveur II – 420-W56";
}
```

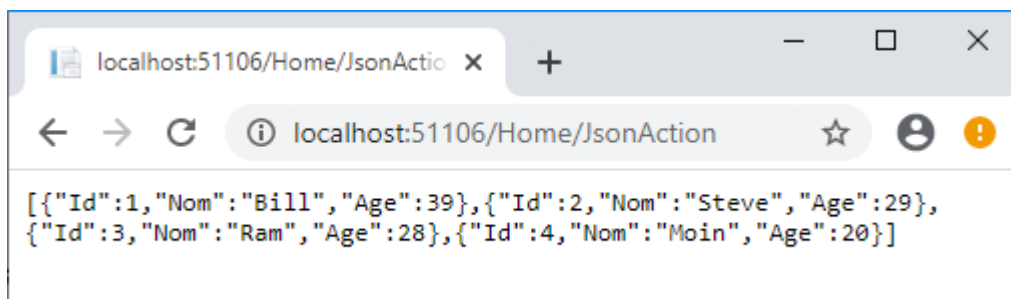
2. Exécutez l'application avec l'URL <https://localhost:51106/Home/Bienvenue> pour obtenir une page comme le montre la figure suivante :



3. Ajoutez l'action suivante au contrôleur Home qui retourne des données Json :

```
public JsonResult ActionJson()
{
    var list = new []
    {
        new { Id=1, Nom="Bill", Age = 39},
        new { Id=2, Nom="Steve", Age = 29},
        new { Id=3, Nom="Ram", Age = 28},
        new { Id=4, Nom="Moin", Age = 20}
    };
    return Json(list, JsonRequestBehavior.AllowGet);
}
```

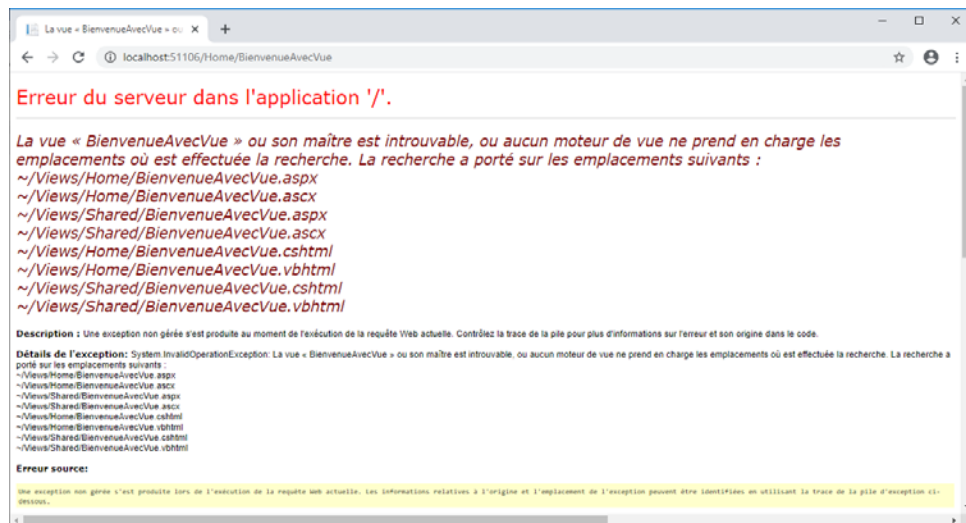
4. Exécutez l'application avec l'URL <https://localhost:51106/Home/JsonAction> pour obtenir une page comme le montre la figure suivante :



5. Ajoutez l'action suivante au contrôleur Home :

```
public ActionResult BienvenueAvecVue()
{
    return View();
}
```

6. Exécutez l'application avec l'URL <https://localhost:51106/Home/BienvenueAvecVue> pour obtenir une page comme le montre la figure suivante :



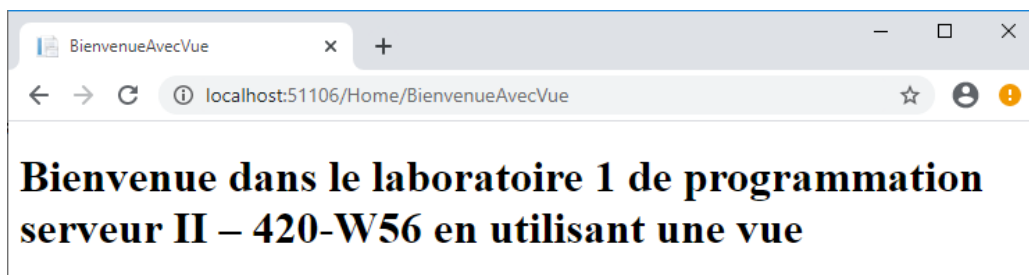
7. Ajoutez la vue suivante (BienvenueAvecVue.cshtml) dans le dossier Views/Home :

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>BienvenueAvecVue</title>
</head>
<body>
    <h1>Bienvenue dans le laboratoire 1 de programmation serveur II - 420-W56 en utilisant une vue</h1>
</body>
</html>
```

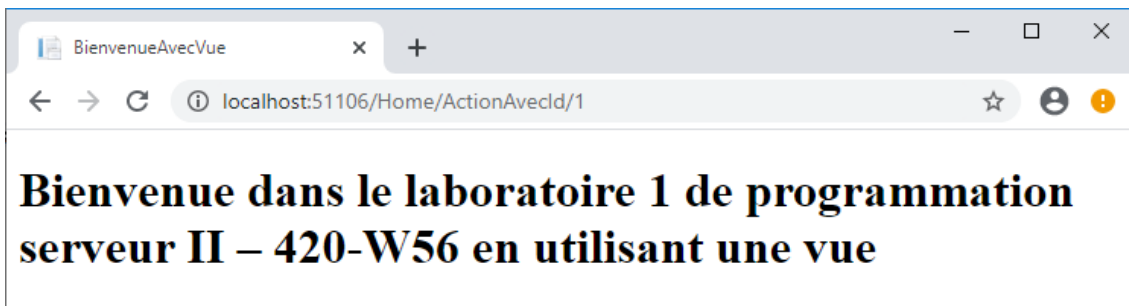
8. Re-Exécutez l'application avec l'URL <https://localhost:51106/Home/BienvenueAvecVue> pour obtenir une page comme le montre la figure suivante :



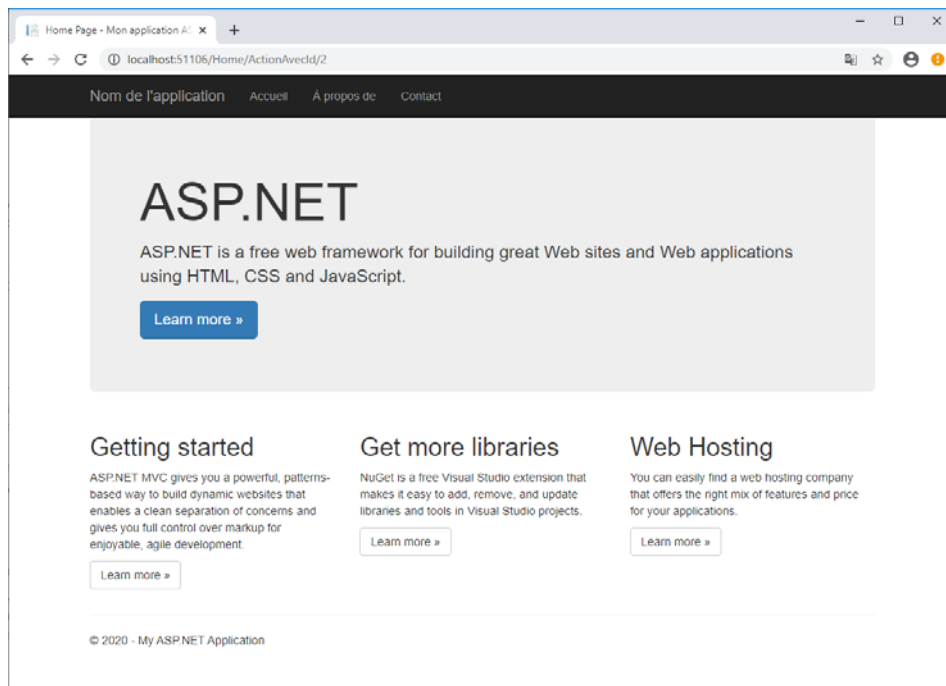
9. Ajoutez une action `ActionAvecId` comme le montre la figure suivante :

```
public ActionResult ActionAvecId(string id)
{
    if (id=="1")
        return View("BienvenueAvecVue");
    else
        return View("Index");
}
```

10. Exécutez l'application avec l'URL <https://localhost:51106/Home/ActionAvecId/1> pour obtenir la figure suivante :



11. Exécutez l'application avec l'URL <https://localhost:51106/Home/ActionAvecId/2> pour obtenir la figure suivante :



Remise :

- À suivre... (ne pas remettre tout de suite)