Module 1: Exploration d'ASP.NET MVC 5

# **Contenu:**

|  |  |
| --- | --- |
|  | [Aperçu du module](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C01.html#P1) |
| **Leçon 1:** | [Présentation des technologies Web Microsoft](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C01.html#P2) |
| **Leçon 2:** | [Vue d'ensemble d'ASP.NET](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C01.html#P3) |
| **Lecon 3:** | [Introduction à ASP.NET MVC 5](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C01.html#P4) |
| **Laboratoire:** | [Explorer ASP.NET MVC 5](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C01.html#P5) |
|  | [Examen du module et points à retenir](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C01.html#P6) |

# **Aperçu du module**

Microsoft ASP.NET MVC 5 et les autres technologies Web du .NET Framework vous aident à créer et à héberger des applications Web dynamiques, puissantes et extensibles. ASP.NET MVC 5 prend en charge le développement agile et piloté par les tests et les dernières normes Web telles que HTML 5. Pour créer des applications Web robustes, vous devez vous familiariser avec les technologies et les produits de la pile Web Microsoft. Vous devez également savoir comment les applications ASP.NET fonctionnent avec IIS, Visual Studio, SQL Server, Microsoft Azure et Microsoft Azure SQL Database pour fournir des pages Web attrayantes aux visiteurs du site. Pour choisir un langage de programmation qui convient le mieux à un ensemble d'exigences métier, vous devez savoir en quoi les applications Model-View-Controller (MVC) diffèrent des autres modèles de programmation ASP.NET: pages Web et formulaires Web.

L'application Web que vous allez concevoir et développer dans les laboratoires tout au long de ce cours vous aidera à développer des applications Web qui répondent aux besoins de l'entreprise.

### **Objectifs**

Après avoir terminé ce module, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Décrivez le rôle d'ASP.NET dans la pile des technologies Web et comment utiliser ASP.NET pour créer des applications Web. |
| • | Décrivez les trois modèles de programmation disponibles dans ASP.NET: pages Web, formulaires Web et MVC et sélectionnez un modèle approprié pour un projet donné. |
| • | Faites la distinction entre un modèle MVC, un contrôleur MVC et une vue MVC. |

# Leçon 1: Présentation des technologies Web Microsoft

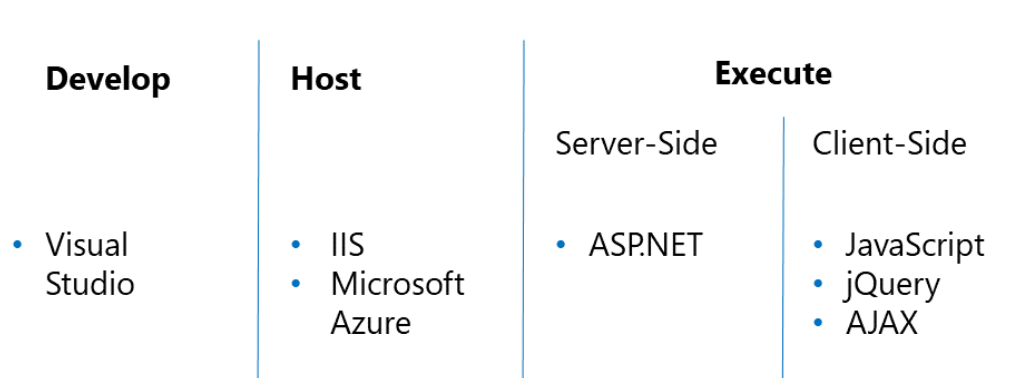
Avant d'utiliser ASP.NET MVC 5, vous devez savoir où se situe Microsoft ASP.NET dans le contexte de l'ensemble des technologies Web de Microsoft. Vous devez savoir comment les sites Web ASP.NET sont hébergés dans Internet Information Server (IIS) et Microsoft Azure, et comment ils exécutent le code côté serveur sur les serveurs Web et le code côté client sur les navigateurs Web, afin de fournir un contenu riche et convaincant.

## **Objectifs de la leçon**

Après avoir terminé cette leçon, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Fournissez un aperçu des technologies Web de Microsoft. |
| • | Fournissez une vue d'ensemble d'ASP.NET. |
| • | Fournit un aperçu des technologies Web côté client, y compris les bibliothèques AJAX et JavaScript. |
| • | Décrivez le rôle d'IIS dans la pile des technologies Web Microsoft. |
| • | Décrivez le rôle de Microsoft Azure dans la pile des technologies Web Microsoft. |

## **Introduction aux technologies Web de Microsoft**



Microsoft propose une large gamme de technologies que vous pouvez utiliser pour créer des applications Web riches et les publier sur des intranets et sur Internet. Outre la publication d'applications Web, vous pouvez utiliser ces technologies pour développer et héberger des sites Web, qui exécutent du code à la fois sur le serveur Web et sur le navigateur Web de l'utilisateur.

### ****Outils de développement****

Vous pouvez créer des sites Web simples avec du texte et des images à l'aide d'un éditeur de texte, tel que le Bloc-notes. Cependant, la plupart des sites Web nécessitent des actions complexes côté serveur, telles que les opérations de base de données, la livraison d'e-mails, les calculs complexes ou le rendu graphique. Pour créer rapidement et facilement des sites Web complexes, hautement fonctionnels et attrayants, Microsoft fournit l'outil Microsoft Visual Studio 2017. Vous pouvez utiliser Visual Studio 2017, un environnement de développement intégré (IDE), pour créer des applications personnalisées basées sur les technologies Microsoft, que ces applications s'exécutent sur le Web, sur des postes de travail, sur des appareils mobiles ou à l'aide des services cloud Microsoft. Visual Studio 2017 dispose de fonctionnalités riches pour la conception, le codage et le débogage de toute application Web ASP.NET, y compris les applications MVC.

### ****Technologies d'hébergement****

Quel que soit l'outil que vous utilisez pour créer une application Web, vous devez utiliser un serveur Web pour héberger l'application Web. Lorsque les utilisateurs visitent votre site, le serveur hôte répond en rendant le code HTML et en le renvoyant au navigateur de l'utilisateur pour affichage. Le serveur hôte peut interroger une base de données avant de créer le code HTML, et le serveur hôte peut effectuer d'autres actions telles que l'envoi d'e-mails ou l'enregistrement de fichiers téléchargés. Lorsque vous avez terminé de créer l'application Web et que vous la préparez pour que les utilisateurs puissent y accéder sur un intranet ou sur Internet, vous devez utiliser un serveur Web entièrement fonctionnel tel que:

|  |  |
| --- | --- |
| • | Microsoft Internet Information Server. IIS est une technologie d'hébergement de sites Web avancée. Vous pouvez installer des serveurs IIS sur votre réseau local ou votre réseau de périmètre, ou utiliser des serveurs IIS hébergés par un fournisseur de services Internet (FAI). IIS peut héberger tous les sites Web ASP.NET, PHP ou Node.js. Vous pouvez faire évoluer IIS pour héberger des sites Web volumineux et occupés en configurant des batteries de serveurs qui contiennent plusieurs serveurs IIS, tous servant le même contenu. |
| • | Microsoft Azure. Microsoft Azure est une plate-forme cloud qui fournit des services à la demande pour créer, déployer, héberger et gérer des applications Web via des centres de données gérés par Microsoft. Lorsque vous utilisez les services Microsoft Azure, vous devez payer uniquement pour les données que votre site Web sert. De plus, vous n'avez pas à vous soucier de la création d'une infrastructure évolutive, car Microsoft Azure ajoute automatiquement des ressources à mesure que votre site Web se développe. |

La plupart des sites Web nécessitent une base de données pour gérer les données telles que les détails du produit, les informations utilisateur et les sujets de discussion. Vous pouvez choisir parmi les technologies Microsoft suivantes pour gérer vos données:

|  |  |
| --- | --- |
| • | Microsoft SQL Server. SQL Server est un serveur de base de données premium que vous pouvez utiliser pour héberger n'importe quelle base de données, de la plus simple à la plus complexe. SQL Server peut évoluer pour prendre en charge de très grandes bases de données et un très grand nombre d'utilisateurs. Vous pouvez créer de grands clusters SQL Server pour garantir la meilleure disponibilité et fiabilité. Bon nombre des plus grandes organisations du monde s'appuient sur SQL Server pour héberger les données. |
| • | Base de données Microsoft Azure SQL. SQL Database est une plate-forme de base de données cloud et fait partie de Microsoft Azure. À l'aide de SQL Database, vous pouvez déployer votre base de données et ne payer que pour les données que vous utilisez. Vous n'avez pas à vous soucier de la gestion de votre infrastructure de base de données, car votre base de données évolue automatiquement à mesure que votre site Web se développe. |

### ****Technologies d'exécution de code****

Le code que vous écrivez dans un outil de développement doit s'exécuter dans l'un des deux emplacements:

|  |  |
| --- | --- |
| • | Sur le serveur Web. Ce code a un accès complet à la puissance du serveur Web et à toutes les bases de données qui lui sont attachées. Il peut accéder rapidement à la base de données, envoyer des e-mails et afficher des pages Web. |
| • | Sur le navigateur Web de l'utilisateur. Ce code répond rapidement aux actions de l'utilisateur, telles que les clics de souris, mais il est plus limité dans ce qu'il peut accomplir sans interagir avec le serveur Web. |

Vous devez utiliser différentes technologies pour exécuter le code côté serveur et le code côté client.

#### Exécution côté serveur

Microsoft ASP.NET est un environnement Web côté serveur qui exécute le code .NET côté serveur que vous écrivez dans Visual Studio 2017. Le code accède à la base de données, rend les pages HTML et les renvoie au navigateur Web. Le modèle de programmation MVC fait partie d'ASP.NET. Les autres technologies côté serveur incluent PHP et Node.js.

#### Exécution côté client

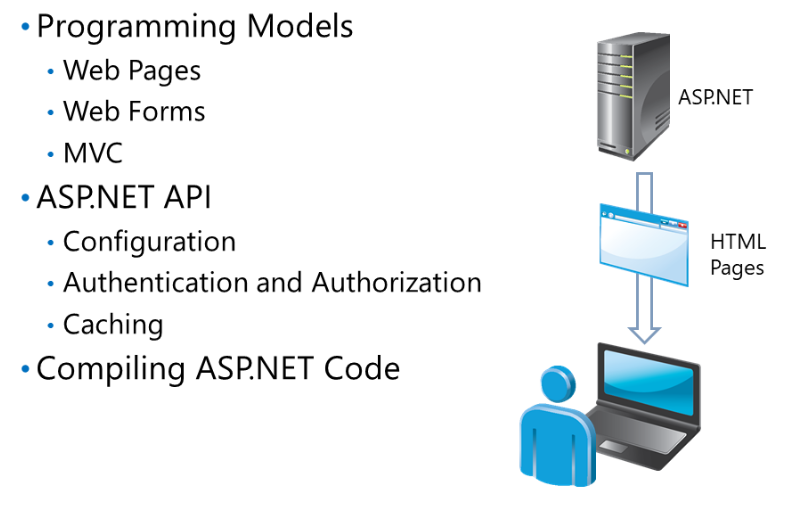
La plupart des navigateurs Web peuvent exécuter du code écrit dans le langage JavaScript. Ce code est envoyé au navigateur sous forme de texte dans une page HTML rendue ou dans un fichier .js séparé. Parce que JavaScript est local pour le navigateur, il peut répondre très rapidement aux actions de l'utilisateur telles que cliquer, pointer ou faire glisser.

De nombreuses bibliothèques JavaScript sont disponibles pour accélérer le développement de code client. Par exemple, la populaire bibliothèque jQuery simplifie l'accès aux éléments de la page et leur manipulation en modifiant leur style ou leur contenu. Lorsque vous utilisez de telles bibliothèques, vous pouvez écrire des fonctions en quelques lignes qui nécessiteraient autrement des centaines de lignes de votre propre code JavaScript.

Les applications ASP.NET peuvent également utiliser la technologie JavaScript et XML asynchrone (AJAX) sur l'ordinateur client. Vous pouvez utiliser AJAX pour mettre à jour une petite section d'une page HTML, au lieu de recharger la page entière à partir du serveur. Ces mises à jour partielles des pages contribuent à rendre les pages Web plus réactives et plus attrayantes.

**Question**: Si vous souhaitez animer un élément de page, par exemple en le fondant, écririez-vous du code côté serveur ou côté client?

## ****Vue d'ensemble d'ASP.NET****



Vous pouvez utiliser ASP.NET pour développer des sites Web dynamiques basés sur une base de données, hautement fonctionnels et évolutifs qui utilisent du code côté client et côté serveur. Vous pouvez créer de nombreux types de sites Web avec ASP.NET, par exemple des portails Web, des sites d'achat en ligne, des blogs et des wikis.

### ****Modèles de programmation****

Lorsque vous utilisez ASP.NET pour créer une application, vous n'êtes pas limité à un seul style de programmation; au lieu de cela, vous pouvez choisir parmi trois modèles de programmation différents. Chaque modèle de programmation a une structure typique dans l'environnement de développement et stocke le code à différents endroits de la hiérarchie Web:

|  |  |
| --- | --- |
| • | **Pages Web**: lorsque vous créez un site à l'aide de pages Web, vous pouvez écrire du code à l'aide du langage de programmation C # ou Visual Basic. Si vous écrivez du code C #, ces pages ont une extension de fichier .cshtml. Si vous écrivez du code Visual Basic, ces pages ont une extension de fichier .vbhtml. ASP.NET exécute le code de ces pages sur le serveur pour restituer les données d'une base de données, répondre à une publication de formulaire ou effectuer d'autres actions. Ce modèle de programmation est simple et facile à apprendre, et convient aux sites simples basés sur les données. ASP.NET comprend des pages Web. |
| • | **Web Forms**: lorsque vous créez un site à l'aide de Web Forms, vous utilisez un modèle de programmation avec de riches contrôles côté serveur et un cycle de vie de page qui n'est pas sans rappeler la création d'applications de bureau. Les contrôles intégrés incluent des boutons, des zones de texte et des vues de grille pour afficher les données tabulées. Vous pouvez également ajouter des contrôles tiers ou créer des contrôles personnalisés. Pour répondre aux actions de l'utilisateur, vous pouvez attacher des gestionnaires d'événements contenant du code aux contrôles côté serveur. Par exemple, pour répondre à un clic sur un bouton appelé firstButton, vous pouvez écrire du code dans le gestionnaire d'événements firstButton\_Click (). |
| • | **MVC:** lorsque vous créez un site à l'aide d'ASP.NET MVC, vous séparez le code côté serveur en trois parties:   |  |  | | --- | --- | | o | **Modèle**. Un modèle MVC définit un ensemble de classes qui représentent les types d'objets gérés par l'application Web. Par exemple, le modèle d'un site de commerce électronique peut inclure une classe de modèle de produit qui définit des propriétés telles que la description, le numéro de catalogue, le prix, etc. Les modèles incluent souvent une logique d'accès aux données qui lit les données d'une base de données et écrit des données dans cette base de données. | | o | **Contrôleurs**. Un contrôleur MVC est une classe qui gère l'interaction de l'utilisateur, crée et modifie des classes de modèle et sélectionne les vues appropriées. Par exemple, lorsqu'un utilisateur demande des détails complets sur un produit particulier, le contrôleur crée une nouvelle instance de la classe de modèle Product et la transmet à la vue Détails, qui l'affiche à l'utilisateur. | | o | **Vues**. Une vue MVC est un composant qui crée les pages Web qui composent l'interface utilisateur de l'application Web. Les contrôleurs transmettent souvent une instance d'une classe de modèle à une vue. La vue affiche les propriétés de la classe de modèle. Par exemple, si le contrôleur passe un objet Produit, la vue peut afficher le nom du produit, une image et le prix. | |

Cette séparation du modèle, de la vue et du code du contrôleur garantit que les applications MVC ont une structure logique, même pour les sites les plus complexes. Il améliore également la testabilité de l'application. En fin de compte, ASP.NET MVC permet plus de contrôle sur le HTML généré que les pages Web ou les formulaires Web.

**Lecture supplémentaire:**Dans la leçon 2, Présentation d'ASP.NET, vous verrez comment choisir le modèle de programmation le plus approprié pour un projet de site Web donné.

**L'API ASP.NET**

Quel que soit le modèle de programmation que vous choisissez, vous avez accès aux classes à partir de l'API ASP.NET. Ces classes sont incluses dans le .NET Framework dans les espaces de noms de l'espace de noms System.Web et peuvent être utilisées pour implémenter rapidement des fonctionnalités de site Web courantes telles que:

|  |  |
| --- | --- |
| • | **Configuration**. À l'aide des fichiers Web.config, vous pouvez configurer votre application Web, quel que soit le modèle de programmation. Les fichiers Web.config sont des fichiers XML avec des balises et des attributs spécifiques acceptés par le runtime ASP.NET. Par exemple, vous pouvez configurer les connexions à la base de données et les pages d'erreur personnalisées dans le fichier Web.config. Dans le code, vous pouvez accéder à la configuration via l'espace de noms System.Web.Configuration. |
| • | **Authentification et autorisation**. De nombreux sites Web exigent que les utilisateurs se connectent en saisissant un nom d'utilisateur et un mot de passe ou en fournissant des informations supplémentaires. Vous pouvez utiliser les fournisseurs d'appartenance ASP.NET pour authentifier et autoriser les utilisateurs et restreindre l'accès au contenu. Vous pouvez également créer des pages qui permettent aux utilisateurs d'enregistrer un nouveau compte, de réinitialiser un mot de passe, de récupérer un mot de passe perdu ou d'effectuer d'autres tâches de gestion de compte. Les fournisseurs d'appartenance appartiennent à l'espace de noms System.Web.Security. |
| • | **Mise en cache**. ASP.NET peut prendre un certain temps pour afficher une page Web complexe qui peut nécessiter plusieurs requêtes de base de données ou des appels à des services Web externes. Vous pouvez utiliser la mise en cache pour atténuer ce délai. ASP.NET met en cache une page rendue en mémoire, afin qu'elle puisse renvoyer la même page aux demandes des utilisateurs suivants sans avoir à la restituer depuis le début. De la même manière, les objets .NET Framework peuvent également être mis en cache. Vous pouvez accéder aux pages mises en cache à l'aide de l'espace de noms System.Runtime.Caching et configurer les caches dans Web.config. |

### ****Compilation du code ASP.NET****

Étant donné que le code côté serveur ASP.NET utilise le .NET Framework, vous devez écrire du code dans un langage de programmation managé .NET tel que C # ou Visual Basic. Avant d'exécuter le code, il doit être compilé en code natif afin que le processeur du serveur puisse le traiter. Il s'agit d'un processus en deux étapes:

|  |  |
| --- | --- |
| 1. | **Compilation vers MSIL**. Lorsque vous créez un site Web dans Visual Studio, le compilateur ASP.NET crée des fichiers .dll avec tout le code compilé dans Microsoft Intermediate Language (MSIL). Ce code est à la fois indépendant du langage utilisé pour écrire l'application et de l'architecture CPU du serveur. |
| 2. | **Compilation en code natif.** Lorsqu'une page est demandée pour la première fois, le Common Language Runtime (CLR) compile MSIL en code natif pour le processeur du serveur. |

Ce processus de compilation en deux étapes permet aux composants écrits dans différents langages de fonctionner ensemble et permet de détecter de nombreuses erreurs au moment de la construction. Notez, cependant, que le rendu des pages peut prendre plus de temps la première fois qu'elles sont demandées après un redémarrage du serveur. Pour éviter ce retard, vous pouvez pré-compiler le site Web.

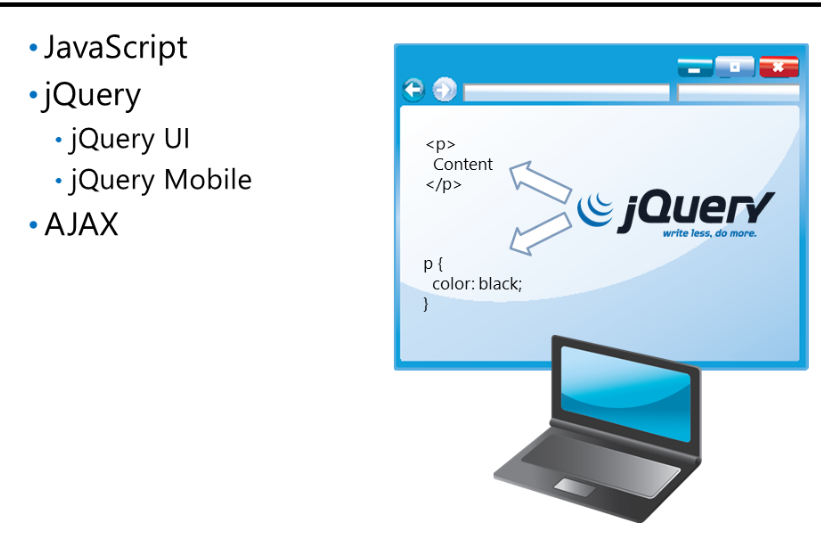
Lorsque vous utilisez le modèle de compilation par défaut, des retards peuvent survenir lorsque le premier utilisateur demande une page. En effet, ASP.NET doit compiler la page avant de la servir au navigateur. Pour éviter de tels retards et pour protéger le code source, utilisez la pré-compilation. Lorsque vous précompilez un site, tous les fichiers ASP.NET, y compris les contrôleurs, les vues et les modèles, sont compilés dans un seul fichier .dll.

**Lecture supplémentaire:**Pour plus d'informations sur la compilation ASP.NET, consultez: For more information about ASP.NET compilation, see:

<http://go.microsoft.com/fwlink/?LinkID=293680&clcid=0x409>

**Question**: Selon vous, lequel des trois modèles de programmation offre le plus de contrôle sur le code HTML et JavaScript envoyé au navigateur?

## ****Technologies Web côté client****



À l'origine, dans ASP.NET et des technologies similaires comme PHP, tout le code fonctionnait sur le serveur Web. Cette approche est souvent pratique car le serveur Web a généralement un accès immédiat à la base de données et plus de puissance de processeur et de mémoire qu'un ordinateur client classique. Cependant, dans une telle approche, chaque action de l'utilisateur nécessite un aller-retour entre le client et le serveur Web, et la plupart des actions nécessitent un rechargement complet de la page. Cela peut prendre beaucoup de temps. Pour répondre rapidement et offrir une meilleure expérience utilisateur, vous pouvez compléter le code côté serveur avec du code côté client qui s'exécute dans le navigateur Web.

### ****JavaScript****

JavaScript est un langage de script simple dont la syntaxe est similaire à C #, et il est pris en charge par la plupart des navigateurs Web. Un langage de script n'est pas compilé. Au lieu de cela, un moteur de script interprète le code au moment de l'exécution afin que le navigateur Web puisse exécuter le code.

**Noter:**Outre JavaScript, Internet Explorer prend en charge VBScript. Il existe également d'autres langages de script, mais JavaScript est pris en charge par pratiquement tous les navigateurs Web. Ce n'est vrai pour aucun autre langage de script. À moins que votre public cible ne soit très limité et que vous ayez le contrôle sur le navigateur utilisé par vos utilisateurs, vous devriez utiliser JavaScript en raison de sa prise en charge presque universelle.

Vous pouvez inclure du code JavaScript dans vos pages ASP.NET, quel que soit le modèle de programmation que vous choisissez. JavaScript est un langage puissant, mais peut nécessiter de nombreuses lignes de code pour obtenir des effets visuels ou appeler des services externes. Les bibliothèques de scripts contiennent des fonctions JavaScript prédéfinies qui aident à implémenter des actions courantes que vous souhaiterez peut-être effectuer sur le code côté client. Vous pouvez utiliser une bibliothèque de scripts, au lieu de créer tout votre propre code JavaScript dès le début; l'utilisation d'une bibliothèque de scripts permet de réduire le temps et les efforts de développement.

Différents navigateurs interprètent JavaScript différemment. Lorsque vous développez un site Internet, vous ne savez pas quels navigateurs les visiteurs du site utilisent. Par conséquent, vous devez écrire du JavaScript qui fonctionne autour de la compatibilité du navigateur.

### ****jQuery****

jQuery est l'une des bibliothèques JavaScript les plus populaires. Il fournit des fonctions élégantes pour interagir avec les éléments HTML de votre page et avec les styles de feuille de style en cascade (CSS). Par exemple, vous pouvez localiser tous les éléments <div> sur une page Web et modifier leur couleur d'arrière-plan en utilisant une seule ligne de code. Pour obtenir le même résultat en utilisant uniquement JavaScript, vous devez écrire plusieurs lignes de code et une boucle de programmation. De plus, le code que vous écrivez peut être différent pour différents navigateurs. En utilisant jQuery, il est plus facile d'écrire du code pour répondre aux actions de l'utilisateur et de créer des animations simples. jQuery gère également les différences de navigateur. Vous pouvez utiliser jQuery pour appeler des services Web sur des ordinateurs distants et mettre à jour la page Web avec les résultats renvoyés.

Le projet jQuery comprend également deux autres bibliothèques JavaScript qui étendent la bibliothèque jQuery de base:

|  |  |
| --- | --- |
| • | **jQuery UI**. Cette bibliothèque comprend un ensemble de widgets que vous pouvez utiliser pour vous aider à créer une interface utilisateur. Il existe des sélecteurs de date, des flèches, des boîtes de dialogue, des zones de texte de saisie semi-automatique et d'autres widgets. Vous pouvez également appliquer des thèmes à vos widgets d'interface utilisateur jQuery pour intégrer leurs couleurs et styles à l'image de marque du site Web. |
| • | **jQuery Mobile**. Cette bibliothèque facilite la fourniture d'une interface utilisateur pour les appareils mobiles tels que les téléphones et les tablettes. Il rend le HTML en utilisant une amélioration progressive. Par exemple, en utilisant jQuery Mobile, vous pouvez afficher des commandes avancées et riches sur des appareils mobiles avancés tels que les smartphones avec de grands écrans, alors que vous pouvez afficher une interface utilisateur plus simple avec les mêmes fonctionnalités pour les appareils mobiles plus anciens avec des écrans plus petits. |

**Noter:**Il existe de nombreuses autres bibliothèques JavaScript telles que Prototype, Enyo, Ext et Dojo Toolkit. Si vous trouvez que l'un de ces éléments est mieux adapté à une tâche particulière, ou si vous avez de l'expérience dans le développement d'applications Web en les utilisant, vous pouvez les inclure dans vos pages ASP.NET au lieu de jQuery.

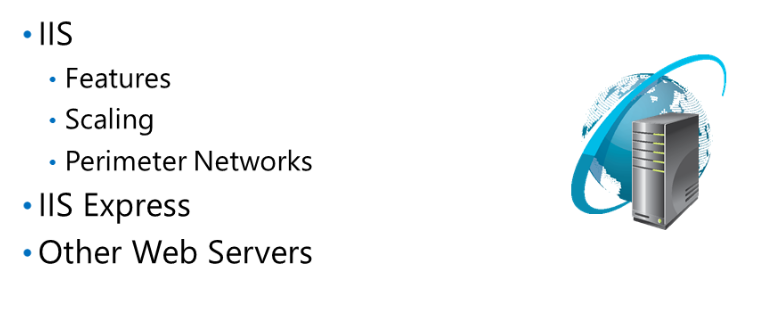
### ****AJAX****

AJAX est une technologie qui permet aux navigateurs de communiquer avec les serveurs Web de manière asynchrone en utilisant l'objet XmlHttpRequest sans actualiser complètement la page. Vous pouvez utiliser AJAX dans une page pour mettre à jour une partie de la page avec de nouvelles données, sans recharger la page entière. Par exemple, vous pouvez utiliser AJAX pour obtenir les derniers commentaires sur un produit, sans actualiser toute la page du produit.

AJAX est une abréviation de JavaScript asynchrone et XML. AJAX est entièrement implémenté en JavaScript et ASP.NET, par défaut, s'appuie sur la bibliothèque jQuery pour gérer les requêtes et réponses AJAX. Le code est exécuté de manière asynchrone, ce qui signifie que le navigateur Web ne se fige pas pendant qu'il attend une réponse AJAX du serveur. Au départ, les développeurs utilisaient XML pour formater les données renvoyées par le serveur. Plus récemment, cependant, les développeurs utilisent JavaScript Object Notation (JSON) comme un format alternatif à XML.

**Question**: Pouvez-vous penser à des problèmes qui pourraient survenir si vous incluez la bibliothèque jQuery avec chaque page de votre application?

## ****Internet Information Server****



Chaque site Web doit être hébergé par un serveur Web. Un serveur Web reçoit des demandes de contenu Web de la part des navigateurs, exécute tout code côté serveur et renvoie des pages Web, des images et d'autres contenus. HTTP est utilisé pour la communication entre le navigateur Web et le serveur Web.

Internet Information Server est un serveur Web qui peut passer d'un petit site Web exécuté sur un seul serveur Web à un grand site Web exécuté sur plusieurs batteries de serveurs Web. Internet Information Server est disponible avec Windows Server.

#### *Fonctionnalités d'Internet Information Server*

Internet Information Server est étroitement intégré à ASP.NET, Visual Studio 2017 et Windows Server. Il comprend les fonctionnalités suivantes:

|  |  |
| --- | --- |
| • | Protocoles de déploiement. Le protocole Web Deploy avancé, intégré à Visual Studio 2017, gère automatiquement le déploiement d'un site Web avec toutes ses dépendances. Vous pouvez également utiliser le protocole FTP (File Transfer Protocol) pour déployer du contenu. |
| • | Gestion centralisée de la ferme Web. Lorsque vous exécutez un grand site Web, vous pouvez configurer une batterie de serveurs à charge équilibrée de nombreux serveurs IIS pour évoluer vers de grandes tailles. Les outils de gestion IIS facilitent le déploiement de sites sur tous les serveurs de la batterie et la gestion des sites après le déploiement. |
| • | Caches hautes performances. Vous pouvez configurer ASP.NET pour utiliser de manière optimale les caches IIS afin d'accélérer les réponses aux demandes des utilisateurs. Lorsque IIS sert une page ou un autre contenu, il peut le mettre en cache en mémoire afin que les demandes identiques ultérieures puissent être traitées plus rapidement. |
| • | Authentification et sécurité. IIS prend en charge les normes d'authentification les plus courantes, notamment l'authentification par carte à puce et l'authentification Windows intégrée. Vous pouvez également utiliser Secure Sockets Layer (SSL) pour crypter les communications sensibles à la sécurité, telles que les pages de connexion et les pages contenant des numéros de carte de crédit. |
| • | Prise en charge d'ASP.NET. IIS est un serveur Web qui prend entièrement en charge ASP.NET. |
| • | Autres technologies côté serveur. Vous pouvez héberger des sites Web développés en PHP et Node.js sur IIS. |

#### Mise à l'échelle d'IIS

Un serveur Web unique a une évolutivité limitée car il est limité par sa vitesse de processeur, sa mémoire, sa vitesse de disque et d'autres facteurs. En outre, les serveurs Web uniques sont vulnérables aux pannes et aux pannes matérielles. Par exemple, lorsqu'un seul serveur Web est hors ligne, votre site Web n'est pas disponible pour les utilisateurs.

Vous pouvez améliorer l'évolutivité et la résilience de votre site Web en l'hébergeant sur une batterie de serveurs multiples. Dans ces batteries de serveurs, de nombreux serveurs partagent le même nom de serveur. Par conséquent, tous les serveurs peuvent répondre aux demandes du navigateur. Un système d'équilibrage de charge tel que l'équilibrage de la charge réseau Windows ou un système matériel tel que Riverbed Cascade, répartit les demandes de manière égale entre les serveurs de la batterie de serveurs. En cas de défaillance d'un serveur, d'autres serveurs sont disponibles pour répondre aux demandes et, par conséquent, la disponibilité du site Web n'est pas interrompue. Les serveurs IIS sont conçus pour fonctionner dans de telles batteries de serveurs et incluent des outils de gestion avancés pour le déploiement de sites et la gestion des serveurs membres.

#### Réseaux de périmètre

Les serveurs Web, y compris IIS, sont souvent situés sur un réseau de périmètre. Un réseau de périmètre a un segment de réseau qui est protégé d'Internet via un pare-feu qui valide et autorise les requêtes HTTP entrantes. Un deuxième pare-feu, qui n'autorise que les requêtes provenant du serveur Web, sépare le réseau de périmètre du réseau organisationnel interne. Les serveurs de prise en charge, tels que les serveurs de base de données, sont généralement situés sur le réseau organisationnel interne.

Dans cette configuration, les utilisateurs Internet peuvent atteindre le serveur IIS pour demander des pages et le serveur IIS peut atteindre le serveur de base de données pour exécuter des requêtes. Cependant, les utilisateurs Internet ne peuvent pas accéder directement au serveur de base de données ou à tout autre ordinateur interne. Cela empêche les utilisateurs malveillants d'exécuter des attaques et garantit un niveau de sécurité élevé.

#### ****IIS Express****

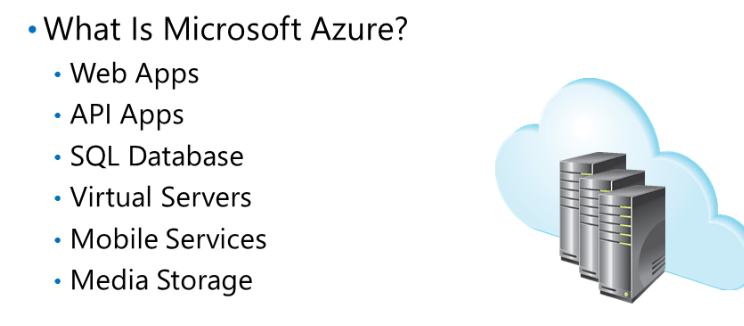
Internet Information Server Express ne fournit pas toutes les fonctionnalités d'Internet Information Server sur Windows Server. Par exemple, vous ne pouvez pas créer des batteries de serveurs à charge équilibrée à l'aide d'Internet Information Server Express. Cependant, il possède toutes les fonctionnalités nécessaires pour héberger des sites Web ASP.NET riches et d'autres sites Web sur un seul serveur.

### **Autres serveurs Web**

**Apache** est un serveur Web non-Microsoft populaire et il existe d'autres alternatives telles que **nginx**. Apache peut être installé sur des ordinateurs clients Windows Server ou Windows pour héberger des sites Web pendant le développement ou pour les déploiements de production.

**Question**: Si vous vouliez héberger un site ASP.NET que vous aviez écrit pour des tests simples par une petite équipe de développeurs, lequel des serveurs Web précédents utiliseriez-vous comme hôte?

## ****Microsoft Azure****



Microsoft Azure fait partie des services cloud de Microsoft pour l'hébergement de systèmes critiques pour l'entreprise. Lorsque vous exécutez du code sur Microsoft Azure, le code s'exécute sur des serveurs dans des centres de données gérés par Microsoft situés dans le monde entier. Vous avez deux avantages principaux lorsque vous utilisez Microsoft Azure pour héberger et déployer votre application Web:

|  |  |
| --- | --- |
| • | Évolutivité flexible: au fur et à mesure que les besoins de votre application Web et de votre base de données augmentent, des ressources serveur supplémentaires sont automatiquement allouées. Vous n'avez pas besoin de planifier des batteries de serveurs ou des systèmes d'équilibrage de charge, car ceux-ci sont intégrés à Microsoft Azure. |
| • | Tarification flexible: avec Microsoft Azure, vous pouvez choisir un modèle de tarification à l'utilisation, ce qui signifie que vous ne payez que pour les données que vous utilisez. Cela rend Microsoft Azure très rentable pour les petits sites Web. Cela rend également les coûts prévisibles pour les grands sites Web. |

Vous pouvez utiliser Microsoft Azure pour héberger les éléments suivants:

|  |  |
| --- | --- |
| • | Applications Web. Vous pouvez héberger un site Web entier sur Microsoft Azure. Microsoft Azure prend en charge les sites Web développés en ASP.NET, PHP ou Node.js. Vous pouvez également déployer des sites Web sur Microsoft Azure directement à partir de Visual Studio 2017. |
| • | Applications API. Une application API n'inclut pas d'interface utilisateur comme une application Web. Au lieu de cela, il consiste en un ensemble de méthodes que vous pouvez appeler à partir d'autres logiciels. Vous pouvez appeler des applications API à partir d'appareils mobiles, d'applications de bureau et à partir du code de site Web exécuté côté client ou côté serveur. |
| • | Bases de données. Lorsque vous utilisez Internet Information Server pour héberger un site Web, vous pouvez utiliser SQL Server pour stocker la base de données du site Web. Lorsque vous hébergez un site Web dans Microsoft Azure, vous pouvez utiliser SQL Database, qui est un service de base de données cloud basé sur SQL Server, pour héberger votre base de données. |
| • | Serveurs virtuels. Vous pouvez provisionner des serveurs virtuels entiers dans Microsoft Azure pour exécuter des logiciels de back-office critiques pour l'entreprise ou utiliser les serveurs virtuels comme environnements de test. Les serveurs virtuels dans Microsoft Azure peuvent exécuter Windows Server ou Linux. |
| • | Services mobiles. Si vous développez des applications pour des appareils mobiles tels que des téléphones et des tablettes, vous pouvez utiliser Microsoft Azure pour héberger les services qui les sous-tendent. Ces services peuvent fournir le stockage des préférences de l'utilisateur, le stockage des données et d'autres fonctions. |
| • | Services médias. Lorsque vous utilisez Microsoft Azure pour héberger des médias tels que l'audio et la vidéo, il est automatiquement disponible pour de nombreux types d'appareils tels que les ordinateurs, les téléphones mobiles et les tablettes, et il est codé dans divers formats, tels que les formats MP4 et Windows Media. |

**Question**: Comment les visiteurs du site sauraient-ils si votre site est hébergé dans Microsoft Azure ou IIS Server?

# Leçon 2: Vue d'ensemble d'ASP.NET

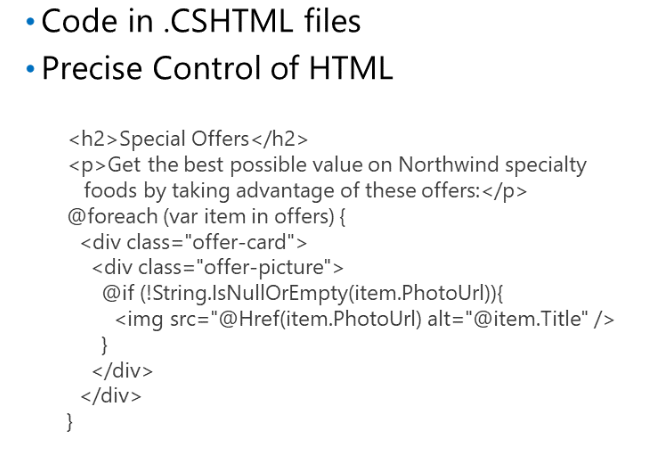
ASP.NET vous aide à créer des sites Web dynamiques qui utilisent du code côté client et côté serveur. De plus, avec ASP.NET, vous n'êtes pas limité à un seul style de programmation; au lieu de cela, vous pouvez choisir parmi trois modèles de programmation différents: pages Web, formulaires Web et MVC. Ces modèles de programmation diffèrent les uns des autres, et ils ont leurs propres avantages et inconvénients dans différents scénarios. ASP.NET fournit également de nombreuses fonctionnalités que vous pouvez utiliser, quel que soit le modèle de programmation que vous choisissez.

## **Objectifs de la leçon**

Après avoir terminé cette leçon, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Décrivez le modèle de programmation des pages Web. |
| • | Décrivez le modèle de programmation Web Forms. |
| • | Décrivez le modèle de programmation MVC. |
| • | Déterminez s'il faut créer des pages Web, des formulaires Web ou des applications Web MVC, en fonction des besoins du client. |
| • | Décrivez les fonctionnalités qui peuvent être utilisées dans toutes les applications ASP.NET, quel que soit le modèle de programmation choisi. |

## ****Applications de pages Web****



**Pages Web** est le modèle de programmation le plus simple que vous pouvez choisir pour créer une application Web dans ASP.NET. Vous pouvez utiliser les pages Web pour créer un site Web rapidement et avec peu de connaissances techniques. Un seul fichier est associé à chaque page Web que les utilisateurs peuvent demander. Pour chaque page, vous écrivez du code HTML, du code côté client et du code côté serveur dans le même fichier .cshtml. **Le symbole @ est utilisé pour distinguer le code côté serveur du HTML et du JavaScript**. Lorsque les utilisateurs demandent la page, le **runtime ASP.NET compile et exécute le code côté serveur** pour rendre le HTML et renvoie ce HTML au navigateur Web pour affichage.

**Noter:**Si vous souhaitez écrire du code côté serveur dans Visual Basic, vous utilisez des fichiers .vbhtml au lieu de fichiers .cshtml.

Vous pouvez utiliser Visual Studio 2017 pour créer des applications de pages Web.

**Un exemple de code de pages Web**

<h2> Offres spéciales </h2>

<p> Obtenez le meilleur rapport qualité-prix possible sur les spécialités Northwind en profitant de ces offres: </p>

@foreach (article var dans les offres) {

<div class = "offer-card">

<div class = "offer-picture">

@if (! String.IsNullOrEmpty (item.PhotoUrl)) {

<img src = "@ Href (item.PhotoUrl) alt =" @ item.Title "/>

}

</div>

</div>

}

Le modèle de programmation des pages Web présente les avantages suivants:

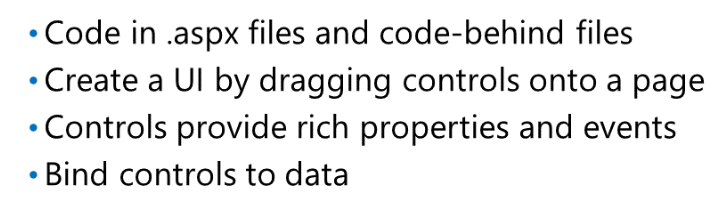
|  |  |
| --- | --- |
| • | C'est simple à apprendre. |
| • | Il fournit un contrôle précis sur le HTML rendu. |

L'utilisation d'un site Web Pages présente certains inconvénients:

|  |  |
| --- | --- |
| • | Il ne fournit aucun contrôle sur les URL qui apparaissent dans la barre d'adresses. |
| • | Les grands sites Web nécessitent un grand nombre de pages, dont chacune doit être codée individuellement. |
| • | Il n'y a pas de séparation entre la logique métier, la logique d'entrée et l'interface utilisateur. |

**Question**: Pourquoi les développeurs Web doivent-ils prendre le contrôle des URL qui apparaissent dans la barre d'adresses lorsqu'un visiteur navigue sur un site de pages Web?

## ****Applications de formulaires Web****



Web Forms est un autre modèle de programmation que vous pouvez choisir dans ASP.NET. Une application Web Forms est caractérisée par des contrôles que vous pouvez faire glisser de la boîte à outils Visual Studio sur chaque page Web. Cette méthode de création d'une interface utilisateur ressemble à la méthode utilisée dans les applications de bureau.

### ****Contrôles des formulaires Web****

ASP.NET fournit une grande variété de contrôles hautement fonctionnels que vous assemblez sur Web Forms. Après avoir ajouté un contrôle à une page, vous pouvez écrire du code pour répondre aux événements utilisateur. Par exemple, vous pouvez utiliser du code dans un événement de clic de bouton pour traiter l'entrée d'un utilisateur dans un formulaire. Les contrôles fournis comprennent:

|  |  |
| --- | --- |
| • | Contrôles d'entrée, tels que les zones de texte, les boutons d'option et les cases à cocher. |
| • | Afficher les commandes, telles que les zones d'image, les cartes d'image et les rotateurs d'annonces. |
| • | Contrôles d'affichage des données, tels que les vues de grille, les vues de formulaire et les graphiques. |
| • | Contrôles de validation, qui vérifient les données saisies par l'utilisateur. |
| • | Contrôles de navigation, tels que les menus et les arborescences. |

Vous pouvez également créer vos propres contrôles personnalisés pour encapsuler des fonctionnalités personnalisées.

### **Fichiers de code Web Forms**

Dans une application Web Forms, le balisage HTML et de contrôle est stocké dans des fichiers avec une extension .aspx. Le code C # côté serveur est généralement écrit dans un fichier .cs associé appelé fichier code-behind. Par exemple, une page appelée Default.aspx a généralement un fichier code-behind appelé Default.aspx.cs.

De même, lorsque vous écrivez des contrôles personnalisés, vous stockez le code HTML et le balisage de contrôle dans un fichier .ascx. Un contrôle appelé CustomControl.ascx a un fichier code-behind appelé CustomControl.ascx.cs.

Les applications Web Forms peuvent également contenir des fichiers de classe portant l'extension .cs.

**Noter:**Si vous écrivez du code côté serveur dans Visual Basic, les fichiers code-behind ont une extension .vb au lieu d'une extension .cs.

**Lier les contrôles aux données**

Dans les applications Web Forms, vous pouvez facilement afficher des données en liant des contrôles à des sources de données. Cette technique supprime la nécessité de parcourir les lignes de données et de créer des affichages ligne par ligne. Par exemple, pour lier un contrôle d'affichage de grille à une table de base de données SQL Server, vous faites glisser un contrôle de source de données SQL sur le formulaire Web et utilisez une boîte de dialogue pour lier l'affichage de grille à la source de données. Lorsque la page est demandée, ASP.NET exécute la requête sur la source de données et fusionne les lignes de données renvoyées avec la page Web Forms.

### **Avantages et inconvénients des formulaires Web**

Le modèle de programmation Web Forms présente les avantages suivants:

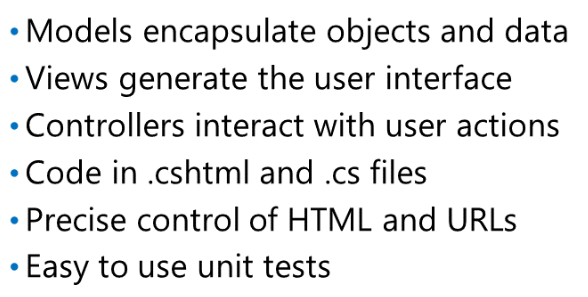
|  |  |
| --- | --- |
| • | Vous pouvez concevoir votre page visuellement à l'aide des contrôles serveur et du mode Création. |
| • | Vous pouvez utiliser une large gamme de contrôles hautement fonctionnels qui encapsulent de nombreuses fonctionnalités. |
| • | Vous pouvez afficher des données sans écrire de nombreuses lignes de code côté serveur. |
| • | L'interface utilisateur dans le fichier .aspx est séparée de l'entrée et de la logique métier dans les fichiers code-behind. |

L'utilisation d'un site Web Forms présente certains inconvénients:

|  |  |
| --- | --- |
| • | Le cycle de vie de la page ASP.NET Web Forms est une couche d'abstraction sur HTTP et peut se comporter de manière inattendue. Vous devez avoir une compréhension complète de ce cycle de vie pour écrire du code dans les gestionnaires d'événements appropriés. |
| • | Vous n'avez pas de contrôle précis sur le balisage généré par les contrôles côté serveur. |
| • | Les contrôles peuvent ajouter de grandes quantités d'informations de balisage et d'état à la page HTML rendue. Cela augmente le temps de chargement des pages. |

**Question**: Pourquoi les développeurs Web devraient-ils se préoccuper du balisage et des informations d'état que les contrôles ASP.NET Web Forms ajoutent à une page HTML rendue?

## ****Applications MVC****



MVC est un autre modèle de programmation disponible dans ASP.NET. Les applications MVC se caractérisent par une forte séparation de la logique métier, du code d'accès aux données et de l'interface utilisateur en modèles, contrôleurs et vues.

### ****Des modèles****

Chaque site Web présente des informations sur différents types d'objets aux visiteurs du site. Par exemple, le site Web d'un éditeur peut présenter des informations sur des livres et des auteurs. Un livre comprend des propriétés telles que le titre, un résumé et le nombre de pages. Un auteur peut avoir des propriétés telles qu'un prénom, un nom de famille et une courte biographie. Chaque livre est lié à un ou plusieurs auteurs.

Lorsque vous écrivez un site Web MVC pour un éditeur, vous créez un modèle avec une classe pour les livres et une classe pour les auteurs. Ces classes de modèles incluraient les propriétés décrites et peuvent inclure des méthodes telles que «acheter ce livre» ou «contacter cet auteur». Si les livres et les auteurs sont stockés dans une base de données, le modèle peut inclure un code d'accès aux données qui peut lire et modifier les enregistrements.

Les modèles sont des classes .NET personnalisées et stockent le code dans des fichiers .cs.

### **Vues**

Chaque site Web doit afficher des pages HTML qu'un navigateur peut afficher. Ce rendu est complété par Views. Par exemple, dans le site de publication, une vue peut extraire des données du modèle de livre et les rendre sur une page Web afin que l'utilisateur puisse voir tous les détails. Dans les applications MVC, Views crée l'interface utilisateur.

Les vues sont des pages de balisage qui stockent à la fois du code HTML et C # dans des fichiers .cshtml. Cela signifie qu'ils ressemblent à des pages Web, mais qu'ils n'incluent que du code d'interface utilisateur. L'autre logique est séparée en modèles et contrôleurs.

### **Contrôleurs**

Chaque site Web doit interagir avec les utilisateurs lorsqu'ils cliquent sur des boutons et des liens. Les contrôleurs répondent aux actions de l'utilisateur, chargent des données à partir d'un modèle et les transmettent à une vue, de manière à afficher une page Web. Par exemple, sur le site de publication, lorsque l'utilisateur double-clique sur un livre, il s'attend à voir tous les détails de ce livre. Le contrôleur de livre reçoit la demande de l'utilisateur, charge le modèle de livre avec le bon ID de livre et le transmet à la vue Détails du livre, qui rend une page Web qui affiche le livre. Les contrôleurs implémentent la logique d'entrée et relient les modèles aux bonnes vues.

Les contrôleurs sont des classes .NET qui héritent de la classe System.Web.Mvc.Controller et stockent le code dans des fichiers .cs.

### ****Avantages et inconvénients de MVC****

Le modèle de programmation MVC présente les avantages suivants:

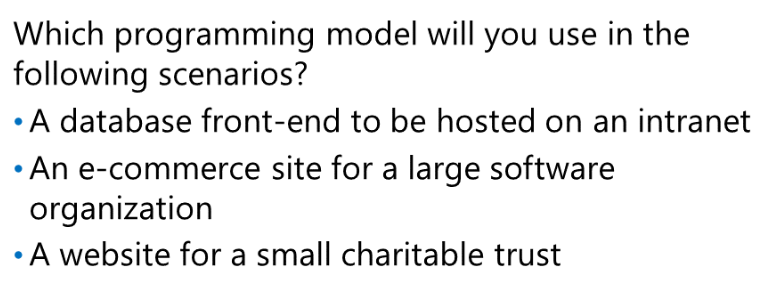
|  |  |
| --- | --- |
| • | Les vues permettent au développeur de contrôler précisément le code HTML rendu. |
| • | Vous pouvez utiliser le moteur de routage pour prendre un contrôle précis des URL. |
| • | La logique métier, la logique d'entrée et la logique d'interface utilisateur sont séparées en modèles, contrôleurs et vues. |
| • | Les techniques de test unitaire et le développement piloté par les tests (TDD) sont possibles. |

L'utilisation d'un site MVC présente certains inconvénients:

|  |  |
| --- | --- |
| • | MVC est potentiellement plus complexe à comprendre que les pages Web ou les formulaires Web. |
| • | MVC vous oblige à séparer vos préoccupations (modèles, vues et contrôleurs). Certains programmeurs peuvent trouver cela difficile. |
| • | Vous ne pouvez pas créer visuellement une interface utilisateur en faisant glisser des contrôles sur une page. |
| • | Vous devez avoir une compréhension complète de HTML, CSS et JavaScript pour développer des vues. |

**Question**: Lorsqu'un utilisateur fait une demande pour un produit particulier dans votre catalogue de produits, quel composant reçoit la demande en premier: un modèle, un contrôleur ou une vue?

## ****Discussion: Scénarios d'application ASP.NET****



Les scénarios suivants décrivent certaines exigences pour les sites Web. Dans chaque cas, discutez du modèle de programmation que vous choisiriez pour implémenter la fonctionnalité requise.

### ****Front-end de base de données****

Votre organisation dispose de son propre système de gestion de la relation client qui stocke les données dans une base de données SQL Server. Votre équipe de développeurs a écrit l'interface utilisateur dans Visual Studio 2017 en tant qu'application de bureau. Les directeurs exigent désormais que tous les ordinateurs puissent accéder à l'application même lorsque l'application client de bureau n'est pas installée. Étant donné que tous les ordinateurs ont un navigateur, vous avez décidé d'écrire une application Web dans ASP.NET pour l'activer.

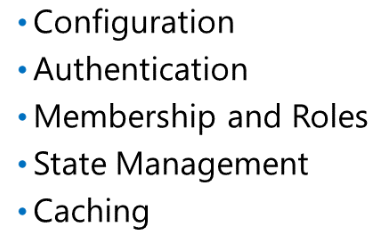
### ****Site de commerce électronique****

Vous êtes consultant pour une grande organisation logicielle. Vous avez été invité à concevoir un site Web de commerce électronique qui permettra aux clients de parcourir l'ensemble du catalogue de logiciels, de télécharger les packages et d'acheter des licences. La société dispose d'une grande équipe de développeurs familiarisés avec la programmation orientée objet .NET. La politique de l'entreprise est d'utiliser le développement piloté par les tests pour tous les logiciels.

### ****Site Web pour une petite fiducie caritative****

Votre ami travaille pour une organisation caritative et vous demande votre avis sur un site Web. Votre ami n'a pas de budget pour engager un consultant, mais a créé des sites Web à l'aide de Microsoft FrontPage. Votre ami souhaite inclure une base de données de produits que les visiteurs du site peuvent parcourir et acheter.

## ****Fonctionnalités ASP.NET partagées****



ASP.NET comprend également une gamme de fonctionnalités disponibles quel que soit le modèle de programmation que vous utilisez. Cela signifie que si vous êtes familiarisé avec ces fonctionnalités grâce à l'utilisation de pages Web ou de formulaires Web, vos connaissances peuvent également être utilisées dans les applications MVC.

### **Configuration**

Lorsque vous configurez un site ASP.NET, vous pouvez contrôler la façon dont les erreurs sont gérées, la façon dont le site se connecte aux bases de données, la façon dont l'entrée utilisateur est validée et de nombreux autres paramètres. Vous pouvez configurer des sites ASP.NET en créant et en modifiant des fichiers Web.config. Le fichier Web.config dans le dossier racine de votre site configure l'ensemble du site, mais vous pouvez remplacer cette configuration à des niveaux inférieurs en créant des fichiers Web.config dans des sous-dossiers.

**Un exemple de fichier Web.config**

<? xml version = "1.0" encoding = "utf-8"?>

<configuration>

<appSettings>

<add key = "aspnet: UseTaskFriendlySynchronizationContext" value = "true" />

<add key = "webpages: Version" value = "2.0.0.0" />

<add key = "webpages: Enabled" value = "false" />

<add key = "PreserveLoginUrl" value = "true" />

<add key = "ClientValidationEnabled" value = "true" />

<add key = "UnobtrusiveJavaScriptEnabled" value = "true" />

</appSettings>

<connectionStrings>

<add name = "ApplicationServices"

connectionString = "Source de données =

. \ SQLEXPRESS; Sécurité intégrée = SSPI;

AttachDBFilename = | DataDirectory | aspnetdb.mdf; Instance utilisateur = true "

providerName = "System.Data.SqlClient" />

</connectionStrings>

<system.web>

<customErrors mode = "RemoteOnly" defaultRedirect = "~ / Error.html" />

<authentication mode = "Forms">

<forms loginUrl = "~ / Account / Login" timeout = "2880">

</forms>

</authentication>

<pages>

<espaces de noms>

<add namespace = "System.Web.Helpers" />

<add namespace = "System.Web.Mvc" />

<add namespace = "System.Web.Mvc.Ajax" />

<add namespace = "System.Web.Mvc.Html" />

<add namespace = "System.Web.Routing" />

<add namespace = "System.Web.WebPages" />

<add namespace = "System.Web.Optimization" />

</namespaces>

</pages>

</system.web>

</configuration>

Si vous devez accéder aux valeurs de configuration au moment de l'exécution dans votre code .NET côté serveur, vous pouvez utiliser l'espace de noms System.Web.Configuration.

### **Authentification**

De nombreux sites Web identifient les utilisateurs par authentification. Cela se fait généralement en demandant et en vérifiant des informations d'identification telles qu'un nom d'utilisateur et un mot de passe, bien que l'authentification puisse être effectuée à l'aide de méthodes plus sophistiquées, telles que l'utilisation de cartes à puce. Certains sites exigent que tous les utilisateurs s'authentifient avant de pouvoir accéder à une page, mais il est courant d'activer l'accès anonyme à certaines pages et d'exiger une authentification uniquement pour le contenu sensible ou sur abonnement.

ASP.NET prend en charge plusieurs mécanismes d'authentification. Par exemple, si vous utilisez Internet Explorer sur un ordinateur Windows, ASP.NET peut être en mesure **d'utiliser l'authentification Windows intégrée**. Dans ce mécanisme, **votre compte utilisateur Windows est utilisé pour vous identifier.** Lorsque vous créez des sites Internet, vous ne pouvez pas être sûr que les utilisateurs disposent de Windows, d'un navigateur compatible ou d'un compte accessible. L'authentification par formulaire est donc souvent utilisée. L'authentification par formulaire est prise en charge par de nombreux navigateurs et peut être configurée pour vérifier les informations d'identification par rapport à une base de données, un service d'annuaire ou d'autres magasins de comptes d'utilisateurs.

**Adhésion et rôles**

Dans de nombreux sites Internet, par exemple Facebook et Twitter, les utilisateurs peuvent créer leurs propres comptes et définir des informations d'identification. De cette manière, votre site peut prendre en charge un grand nombre de membres sans nécessiter un effort administratif énorme car les administrateurs ne créent pas de comptes.

Dans ASP.NET, un fournisseur d'appartenance est un composant qui implémente les fonctionnalités de gestion des comptes d'utilisateurs. Plusieurs fournisseurs d'appartenance sont pris en charge par ASP.NET, tels que le fournisseur d'appartenance SQL, qui utilise une base de données SQL pour stocker les comptes d'utilisateurs. Vous pouvez également créer un fournisseur d'appartenance personnalisé, héritant de l'un des fournisseurs par défaut, si vous avez des exigences uniques.

Lorsque vous avez plus de quelques utilisateurs, vous souhaiterez peut-être les regrouper en rôles avec différents niveaux d'accès. Par exemple, vous pouvez créer un rôle «Membres Gold» contenant des comptes utilisateurs ayant accès aux meilleures offres spéciales. Les fournisseurs de rôles ASP.NET vous permettent de créer et de remplir des rôles avec le minimum de code personnalisé.

Vous pouvez activer l'accès aux pages de votre site Web pour des comptes d'utilisateurs individuels ou pour tous les membres d'un rôle. Ce processus est appelé autorisation.

**Gestion de l'état**

Les serveurs Web et les navigateurs Web communiquent via HTTP. Il s'agit d'un protocole sans état dans lequel chaque requête est distincte des requêtes antérieures et postérieures. Les valeurs des demandes précédentes ne sont pas automatiquement mémorisées.

Cependant, lorsque vous créez une application Web, vous devez fréquemment conserver les valeurs sur plusieurs demandes de page. Par exemple, si un utilisateur place un produit dans un panier, puis clique sur «Payer», il s'agit d'une demande Web distincte, mais le serveur doit conserver le contenu de ce panier; sinon, le panier sera vidé et le client n'achètera rien. ASP.NET fournit plusieurs emplacements où vous pouvez stocker ces valeurs ou informations d'état sur plusieurs demandes.

**Mise en cache**

Une page ASP.NET est créée dynamiquement par le runtime ASP.NET sur le serveur Web. Par exemple, dans une application Pages Web, le moteur d'exécution doit exécuter le code C # dans la page pour rendre le HTML afin de le renvoyer au navigateur. Ce code C # peut effectuer des opérations complexes et chronophages. Il peut exécuter plusieurs requêtes sur une base de données ou appeler des services sur des serveurs distants. Vous pouvez atténuer ces délais en utilisant des caches ASP.NET.

Par exemple, vous pouvez utiliser le cache de page ASP.NET pour stocker la version rendue d'une page fréquemment demandée dans la mémoire du serveur Web. La première page de votre catalogue de produits peut être demandée des centaines ou des milliers de fois par jour par de nombreux utilisateurs. Si vous mettez la page en mémoire cache la première fois qu'elle est rendue, le serveur Web peut la servir à la plupart des utilisateurs très rapidement, sans interroger le serveur de base de données et créer la page à partir de zéro.

**Question**: Pouvez-vous penser à d'autres fonctionnalités dont toutes les applications ASP.NET pourraient avoir besoin, quel que soit le modèle de programmation qu'elles utilisent?

# Leçon 3: Introduction à ASP.NET MVC 5

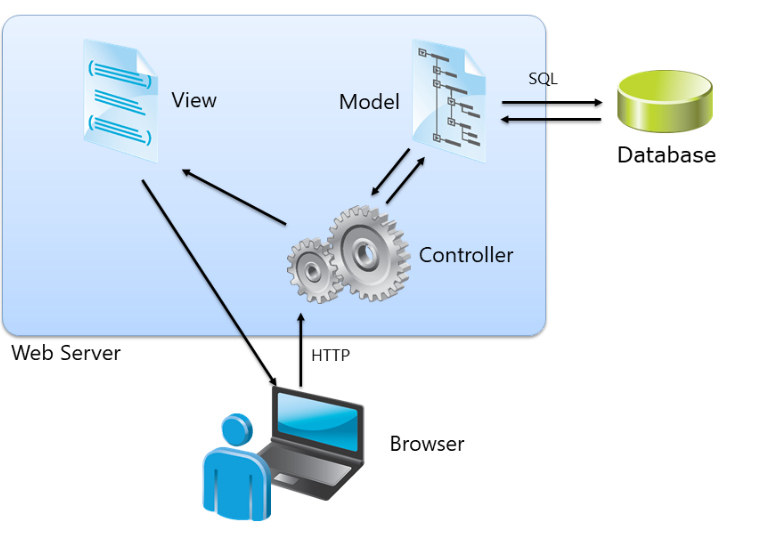
Vous devez savoir comment les modèles, les vues et les contrôleurs fonctionnent ensemble pour rendre le HTML et comment la structure des applications MVC détermine l'affichage des informations dans un projet Visual Studio 2017. Vous pouvez examiner les nouvelles fonctionnalités de MVC 5, incluses dans ASP.NET, pour créer des applications riches et attrayantes.

## **Objectifs de la leçon**

Après avoir terminé cette leçon, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Décrivez les modèles, les vues et les contrôleurs. |
| • | Explorez les principales fonctionnalités d'une application Web ASP.NET MVC 5 dans Visual Studio. |

## ****Modèles, vues et contrôleurs****



Les modèles représentent les données et la logique métier qui l'accompagne, les contrôleurs interagissent avec les demandes des utilisateurs et implémentent la logique d'entrée, et les vues créent l'interface utilisateur. En examinant la façon dont une demande utilisateur est traitée par ASP.NET MVC 5, vous pouvez comprendre comment les données circulent à travers les modèles, les vues et les contrôleurs avant d'être renvoyées au navigateur.

### **Modèles et données**

Un modèle est un ensemble de classes .NET qui représente des objets gérés par votre site Web. Par exemple, le modèle d'une application de commerce électronique peut inclure une classe de modèle de produit avec des propriétés telles que l'ID de produit, le numéro de pièce, le numéro de catalogue, le nom, la description et le prix. Comme toute classe .NET, les classes de modèle peuvent inclure un constructeur, qui est une procédure qui s'exécute lorsqu'une nouvelle instance de cette classe est créée. Vous pouvez également inclure d'autres procédures, si nécessaire. Ces procédures encapsulent la logique métier. Par exemple, vous pouvez écrire une procédure de publication qui marque le produit comme prêt à vendre.

La plupart des sites Web stockent des informations dans une base de données. Dans une application MVC, le modèle comprend du code qui lit et écrit des enregistrements de base de données. ASP.NET MVC fonctionne avec de nombreux frameworks d'accès aux données. Cependant, le framework le plus couramment utilisé est l'Entity Framework.

### ****Contrôleurs et actions****

Un contrôleur est une classe .NET qui répond aux requêtes du navigateur Web dans une application MVC. Il existe généralement une classe de contrôleur pour chaque classe de modèle. Les contrôleurs incluent des actions, qui sont des méthodes qui s'exécutent en réponse à une demande utilisateur. Par exemple, le contrôleur de produit peut inclure une action d'achat qui s'exécute lorsque l'utilisateur clique sur le bouton Ajouter au panier dans votre application Web.

Les contrôleurs héritent de la classe de base System.Web.Mvc.Controller. Les actions renvoient généralement un objet System.Web.Mvc.ActionResult.

### ****Vues et rasoir****

Une vue est, par défaut, un fichier .cshtml ou .vbhtml qui comprend à la fois le balisage HTML et le code de programmation. Un moteur de vue interprète les fichiers de vue, exécute le code côté serveur et rend le HTML au navigateur Web. **Razor est le moteur d'affichage par défaut dans ASP.NET MVC 5.**

**Partie d'une vue rasoir**

<h2> Détails </h2>

<fieldset>

<legend> Commentaire </legend>

<div class = "display-label">

@ Html.DisplayNameFor (modèle => modèle.Subject)

</div>

<div class = "display-field">

@ Html.DisplayFor (modèle => modèle.Subject)

</div>

<div class = "display-label">

@ Html.DisplayNameFor (modèle => model.Body)

</div>

<div class = "display-field">

@ Html.DisplayFor (modèle => model.Body)

</div>

</fieldset>

Souvent, la vue affiche les propriétés d'une classe de modèle. Dans l'exemple de code précédent, la propriété Subject et la propriété Body sont incorporées dans la page.

### ****Demander le cycle de vie****

Le cycle de vie de la demande comprend une série d'événements qui se produisent lorsqu'une demande Web est traitée. Les étapes suivantes illustrent le processus que les applications MVC suivent pour répondre à une demande utilisateur typique. La demande concerne les détails d'un produit avec l'ID «1»:

|  |  |
| --- | --- |
| 1. | L'utilisateur demande l'adresse Web: http://www.adventure-works.com/product/display/1 |
| 2. | Le moteur de routage MVC examine la demande et détermine qu'il doit la transmettre au contrôleur de produit et à l'action Afficher. |
| 3. | L'action Afficher dans le contrôleur de produit crée une nouvelle instance de la classe de modèle de produit. |
| 4. | La classe de modèle de produit interroge la base de données pour obtenir des informations sur le produit avec l'ID «1». |
| 5. | L'action Afficher crée également une nouvelle instance de la vue d'affichage du produit et lui transmet le modèle de produit. |
| 6. | Le moteur de vue Razor exécute le code côté serveur dans la vue d'affichage du produit pour rendre le HTML. Dans ce cas, le code côté serveur insère des propriétés telles que le titre, la description, le numéro de catalogue et le prix dans le HTML. |
| 7. | La page HTML terminée est renvoyée au navigateur pour affichage. |

**Question**: Si vous vouliez écrire du code qui restitue les données de votre catalogue de produits dans un tableau HTML, placeriez-vous ce code dans un modèle, une vue, un contrôleur ou une fonction JavaScript?

## **Démonstration: Comment explorer une application MVC**

Dans cette démonstration, vous verrez comment utiliser un exemple d'application de partage de photos pour explorer la structure d'un site Web MVC.

### **Étapes de démonstration**

|  |  |
| --- | --- |
| • | Vous trouverez les étapes de la section «Leçon 3: Introduction à ASP.NET MVC 5» sur la page suivante:<https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD01_DEMO.md>. |

# **Atelier: Exploration d'ASP.NET MVC 5**

### **Scénario**

Vous travaillez en tant que développeur junior chez Adventure Works. Un développeur senior vous a demandé d'étudier la possibilité de créer une application Web de partage de photos pour les clients de votre organisation, similaire à celle que le développeur senior a vue sur Internet. Une telle application favorisera une communauté de cyclistes qui utilisent des équipements Adventure Works, et les membres de la communauté pourront partager leurs expériences. Cette initiative vise à accroître la popularité d'Adventure Works Cycles et, par conséquent, à augmenter les ventes. On vous a demandé de commencer la planification de l'application en examinant une application de partage de photos existante et en évaluant ses fonctionnalités. Vous avez également été invité à examiner les modèles de programmation disponibles pour les développeurs ASP.NET. Pour faire ça, vous devez créer des applications Web de base écrites avec trois modèles différents: pages Web, formulaires Web et MVC. Votre responsable vous a demandé de répondre aux questions spécifiques suivantes pour chaque modèle de programmation:

|  |  |
| --- | --- |
| • | Comment le développeur définit-il une chaîne de connexion et un fournisseur de données? |
| • | Comment le développeur impose-t-il une mise en page cohérente, avec la marque et les menus Adventure Works, sur toutes les pages de l'application Web? |
| • | Comment le développeur définit-il une feuille de style en cascade avec un ensemble cohérent de couleurs, de polices, de bordures et d'autres styles? |
| • | Comment le développeur ajoute-t-il une nouvelle page à l'application et y applique-t-il la mise en page et les styles? |

### **Objectifs**

Après avoir terminé cet atelier, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Décrivez et comparez les trois modèles de programmation disponibles dans ASP.NET. |
| • | Décrivez la structure de chaque application Web développée dans les trois modèles de programmation: pages Web, formulaires Web et MVC. |
| • | Sélectionnez un modèle de programmation approprié pour un ensemble donné d'exigences d'application Web. |

##### **Configuration du laboratoire**

Durée estimée: 80 minutes

Vous trouverez les étapes de haut niveau sur la page suivante:<https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD01_LAB_MANUAL.md>.

Vous trouverez les étapes détaillées sur la page suivante:<https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD01_LAK.md>.

### **Exercice 1: Explorer une application de partage de photos**

##### **Scénario**

Dans cet exercice, vous commencerez par examiner l'application de partage de photos.

### **Exercice 2: Exploration d'une application de pages Web**

##### **Scénario**

Dans cet exercice, vous allez créer une application de pages Web simple et explorer sa structure.

### **Exercice 3: Exploration d'une application Web Forms**

##### **Scénario**

Dans cet exercice, vous allez créer une application Web Forms simple et explorer sa structure.

### **Exercice 4: Explorer une application MVC**

##### **Scénario**

Dans cet exercice, vous allez créer une application MVC simple et explorer sa structure.

### **Question (s) de révision**

**Vérifiez vos connaissances**

**Découverte**

**Lequel des trois modèles de programmation a la méthode la plus simple pour appliquer une seule mise en page sur plusieurs pages?**

Afficher la solution Réinitialiser

**Vérifiez vos connaissances**

**Découverte**

**Lequel des trois modèles de programmation a la méthode la plus simple pour créer une interface utilisateur?**

Afficher la solution Réinitialiser

**Vérifiez vos connaissances**

**Découverte**

**Lequel des modèles de programmation d'application recommanderez-vous pour l'application de partage de photos: pages Web, formulaires Web ou MVC?**

Afficher la solution Réinitialiser

# **Examen du module et points à retenir**

Dans ce module, vous avez vu les outils, technologies et serveurs Web disponibles dans la pile Web Microsoft, que vous pouvez utiliser pour créer et héberger des applications Web pour les intranets et sur Internet. Vous devez également être en mesure de distinguer les applications écrites dans les trois modèles de programmation ASP.NET: pages Web, formulaires Web et MVC. Vous devriez pouvoir utiliser les applications MVC pour rendre des pages Web à l'aide de modèles, de vues et de contrôleurs.

**Meilleur entrainement:**Utilisez les pages Web lorsque vous avez des exigences simples ou que vous avez des développeurs avec peu d'expérience d'ASP.NET.

**Meilleur entrainement:**Utilisez Web Forms lorsque vous souhaitez créer une interface utilisateur en faisant glisser des contrôles d'une boîte à outils sur chaque page Web ou lorsque vos développeurs ont une expérience des Web Forms ou Windows Forms.

**Meilleur entrainement:**Utilisez MVC lorsque vous voulez le contrôle le plus précis sur HTML et les URL, lorsque vous souhaitez séparer proprement la logique métier, le code d'interface utilisateur et la logique d'entrée, ou lorsque vous souhaitez effectuer un développement piloté par les tests.

**Problèmes courants et conseils de dépannage**

| **Problème commun** | **Astuce de dépannage** |
| --- | --- |
| Vous ajoutez une nouvelle vue à une application MVC, mais lorsque vous essayez d'accéder à la page, vous recevez une erreur HTTP 404. | Veuillez consulter le contenu du compagnon étudiant pour ce cours. |

**Lecture supplémentaire:**Pour plus d'informations sur ASP.NET, y compris les forums, les blogs et les outils tiers, visitez le site ASP.NET officiel:<http://go.microsoft.com/fwlink/?LinkID=293681&clcid=0x409>

### **Question (s) de révision**

**Vérifiez vos connaissances**

**Découverte**

**Laquelle des fonctionnalités partagées d'ASP.NET pouvez-vous utiliser dans les pages Web, les formulaires Web et les applications MVC pour augmenter la vitesse à laquelle les pages fréquemment demandées sont renvoyées au navigateur?**

Afficher la solution Réinitialiser

### **Problèmes et scénarios du monde réel**

Vous avez écrit une application Web pour un client qui vend des chapeaux. Les visiteurs du site pourront s'inscrire, échanger des bons d'offre et acheter des chapeaux. Vous vous attendez à ce que le trafic du site soit stable pendant la majeure partie de l'année, mais qu'il atteigne son maximum juste avant Noël. Devez-vous recommander IIS ou Microsoft Azure pour l'hébergement du site?