Module 9: Création de pages réactives dans les applications Web 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/20486C09.html#P1) |
| **Leçon 1:** | [Utilisation d'AJAX et des mises à jour partielles de page](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C09.html#P2) |
| **Leçon 2:** | [Mettre en œuvre une stratégie de mise en cache](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C09.html#P3) |
| **Laboratoire:** | [Création de pages réactives dans les applications Web 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/20486C09.html#P4) |
|  | [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/20486C09.html#P5) |

# **Aperçu du module**

De nombreuses applications Web doivent afficher une grande quantité d'informations et de graphiques. De grands volumes de données rendent les applications Web plus longues à charger. Au lieu de charger tous les éléments de la page plusieurs fois, vous pouvez implémenter la mise en cache dans vos applications Web pour réduire le besoin de charger à plusieurs reprises les mêmes éléments. Vous pouvez également utiliser des mises à jour partielles de la page pour réduire le besoin de charger la page Web entière, en permettant à l'application de n'actualiser que des sections spécifiques de la page Web.

### **Objectifs**

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

|  |  |
| --- | --- |
| • | Mettez en œuvre des mises à jour partielles des pages qui contribuent à réduire le besoin de recharger des pages entières. |
| • | Implémentez la mise en cache pour réduire le temps de chargement des différents éléments d'une page Web. |

# Leçon 1: Utilisation d'AJAX et des mises à jour partielles de page

Lors du développement d'applications Web, pour mettre à jour des sections individuelles de la page, vous devrez peut-être recharger la page Web entière. JavaScript et XML asynchrones (AJAX) dans ASP.NET MVC 5 permettent des mises à jour partielles de page, pour aider à mettre à jour des sections d'une page Web, sans recharger la page entière. L'assistant Ajax.ActionLink aide à implémenter des mises à jour partielles de page dans votre application Web.

## **Objectifs de la leçon**

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

|  |  |
| --- | --- |
| • | Décrivez les avantages de l'utilisation de mises à jour partielles de page. |
| • | Utilisez AJAX dans une application Web ASP.NET MVC 5. |
| • | Utilisez l'assistant Ajax.ActionLink. |

## **Pourquoi utiliser des mises à jour partielles de page?**

ASP.NET et MVC facilitent le traitement côté serveur. Le traitement côté serveur permet le rendu HTML du côté serveur et le serveur génère généralement le HTML. Lorsque vous mettez à jour une page Web, toutes les mises à jour ou actions effectuées sur la page nécessitent une demande aller-retour au serveur. Ces demandes de serveur constantes affectent les performances de l'application.

Le modèle de développement AJAX permet de réduire le besoin d'actualiser une page Web entière, chaque fois qu'une mise à jour du contenu de la page est requise. AJAX utilise JavaScript et XML pour obtenir des informations du système client. AJAX crée des pages Web basées sur les informations XML téléchargées à partir du serveur. Cependant, développer des applications Web à l'aide d'AJAX n'est pas facile, car il nécessite l'utilisation de technologies complexes, telles que JavaScript et XML. Microsoft inclut une fonctionnalité dans ASP.NET appelée mises à jour de page partielles qui fonctionne avec AJAX pour réduire le besoin d'actualiser une page Web entière, chaque fois qu'une mise à jour se produit.

Les mises à jour partielles de page utilisent les technologies AJAX pour aider à mettre à jour des sections individuelles d'une page Web, pendant la publication. Mises à jour partielles de la page:

|  |  |
| --- | --- |
| • | Nécessite moins de lignes de code. |
| • | Aide à réduire les données envoyées aux utilisateurs, chaque fois qu'une mise à jour de page Web se produit. |
| • | Augmentez la réactivité de l'application Web. |

**Question**: Comment les mises à jour partielles de page contribuent-elles à améliorer la réactivité d'une application Web?

## **Utilisation d'AJAX dans une application Web MVC 5**

Pour implémenter AJAX dans votre application MVC 5, vous devez créer des vues qui ne rendent que le contenu mis à jour, et non la page Web entière. Vous pouvez initialement développer votre application Web sans utiliser AJAX, puis vérifier l'application pour toute erreur de fonctionnalité. Cette pratique permet de réduire le temps nécessaire pour dépanner l'application. Cette pratique permet également de séparer les erreurs de fonctionnalité de l'application des erreurs qui se produisent lors de la mise en œuvre d'AJAX.

Pour implémenter des mises à jour partielles de page, vous devez créer une vue, appelée vue partielle, qui comprend uniquement la section que vous devez mettre à jour.

**Un exemple de vue**

@ {

Disposition = null;

}

<! DOCTYPE html>

<html>

<head>

<meta name = "viewport" content = "width = device-width" />

<title> Index </title>

</head>

<body>

<div>

<div id = "divMessage"> @ ViewBag.Message </div>

@ Html.ActionLink ("Actualiser", "HelloWorld")

</div>

</body>

</html>

Dans l'exemple de code précédent, ActionLink permet de diriger les utilisateurs vers une autre vue appelée HelloWorld.

**La vue Hello World**

@ {

Disposition = null;

}

<! DOCTYPE html>

<html>

<head>

<meta name = "viewport" content = "width = device-width" />

<title> HelloWorld </title>

</head>

<body>

<div>

@ ViewBag.Message

</div>

</body>

</html>

Pour implémenter AJAX dans l'exemple de code précédent, vous devez mettre à jour la vue HelloWorld pour restituer uniquement le contenu mis à jour.

**Un exemple de ViewBag**

@ ViewBag.Message

Avec AJAX, le Javascript ne récupère qu'une partie spécifique d'une page Web, que vous souhaitez mettre à jour, à partir du serveur. Dans la classe ViewController, vous devez mettre à jour la fonction View pour renvoyer la classe PartialView, au lieu de la classe View entière.

**Retour d'une vue partielle dans une action de contrôleur**

public class Default1Controller: Controller

{

//

// GET: / Default1 /

Index ActionResult public ()

{

ViewBag.Message = "Bonjour";

retour Vue ();

}

[HttpGet]

public PartialViewResult HelloWorld ()

{

ViewBag.Message = "Bonjour tout le monde";

return PartialView ();

}

}

Vous pouvez éventuellement ajouter les attributs HttpGet ou HttpPost avant la fonction View. Ces attributs aident à indiquer si la mise à jour partielle de la page doit être effectuée via la méthode HTTP POST ou HTTP GET.

**Question**: Quelle est l'action obligatoire que vous devez effectuer pour implémenter des mises à jour partielles de page dans votre application Web?

## **L'assistant Ajax.ActionLink**

Vous pouvez utiliser l'assistant Ajax.ActionLink pour déclencher des mises à jour partielles de page. L'assistant Ajax.ActionLink aide à lancer le Javascript, à obtenir les informations HTML mises à jour de la vue et à remplacer ou insérer les informations HTML mises à jour à un emplacement spécifique.

**Utilisation de l'assistant Ajax.ActionLink**

@ {

Disposition = null;

}

<! DOCTYPE html>

<html>

<head>

<meta name = "viewport" content = "width = device-width" />

<title> Index </title>

<script src = "http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.8.3.js" type = "text / javascript"> </script>

<script src = "http://ajax.aspnetcdn.com/ajax/mvc/3.0/jquery.unobtrusive-ajax.min.js"

type = "text / javascript"> </script>

</head>

<body>

<div>

<div id = "divMessage"> @ ViewBag.Message </div>

@ Ajax.ActionLink ("Refresh", "HelloWorld", new AjaxOptions {HttpMethod = "POST", UpdateTargetId = "divMessage", InsertionMode = InsertionMode.Replace})

</div>

</body>

</html>

Dans l'exemple précédent, des paramètres tels que HttpMethod et UpdateTargetId sont inclus avec l'assistant Ajax.ActionLink, pour:

|  |  |
| --- | --- |
| • | Obtenez les informations HTML de la vue HelloWorld à l'aide de la méthode HTTP POST. |
| • | Remplacez le contenu de l'élément HTML divMessage. |

Lors de l'utilisation de l'assistant Ajax.ActionLink, vous devez inclure les bibliothèques discrètes jQuery et jQuery dans la même page Web, car l'assistant Ajax.ActionLink utilise des scripts de ces deux bibliothèques.

**Question**: Quelle est la fonction principale de l'assistant Ajax.ActionLink?

# Leçon 2: Mettre en œuvre une stratégie de mise en cache

Les applications Web affichent des informations sur une page Web en récupérant les informations à partir d'une base de données. Si les informations qui doivent être extraites de la base de données sont volumineuses, l'application peut mettre plus de temps à afficher les informations sur une page Web. ASP.NET MVC 5 prend en charge certaines techniques de mise en cache pour aider à réduire le temps requis pour traiter une demande utilisateur.

Avant d'implémenter la mise en cache, vous devez d'abord analyser si la mise en cache est pertinente pour votre application, car la mise en cache n'est pas pertinente pour les pages Web dont le contenu change fréquemment. Pour implémenter avec succès la mise en cache dans votre application Web, vous devez vous familiariser avec les différents types de caches, tels que le cache de sortie, le cache de données et le cache HTTP.

## **Objectifs de la leçon**

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

|  |  |
| --- | --- |
| • | Décrivez les avantages de la mise en cache. |
| • | Décrivez le cache de sortie. |
| • | Décrivez le cache de données. |
| • | Décrivez le cache HTTP. |
| • | Décrivez comment empêcher la mise en cache du contenu de page Web qui change fréquemment. |
| • | Configurez la mise en cache. |

## **Pourquoi utiliser la mise en cache?**

La mise en cache consiste à stocker les informations obtenues à partir d'une base de données dans la mémoire d'un serveur Web. Si le contenu rendu par une page Web est de nature statique, le contenu peut être stocké dans des caches ou des serveurs proxy. Lorsqu'un utilisateur demande du contenu à une application Web, la mise en cache garantit que l'utilisateur reçoit le contenu du cache, éliminant ainsi le besoin d'un traitement répété en temps réel.

Mise en cache:

|  |  |
| --- | --- |
| • | Réduit le besoin de récupérer à plusieurs reprises les mêmes informations de la base de données. |
| • | Réduit le besoin de retraiter les données, si un utilisateur place une demande plusieurs fois. |
| • | Aide à améliorer les performances d'une application Web, en réduisant la charge sur les serveurs. |
| • | Aide à augmenter le nombre d'utilisateurs pouvant accéder à la batterie de serveurs. |

Cependant, la mise en cache n'aide pas les applications Web qui incluent des changements de contenu fréquents. En effet, le contenu rendu à partir d'un cache peut être obsolète, par rapport aux informations actuelles. Par conséquent, vous devez évaluer le contenu de votre application Web et analyser l'impact du rendu du contenu obsolète, avant d'implémenter la mise en cache.

**Question**: Comment la mise en cache aide-t-elle à augmenter l'évolutivité d'une application Web?

## **Le cache de sortie**

Le cache de sortie permet aux moteurs ASP.NET de stocker le contenu rendu d'une page Web dans la mémoire du serveur Web. Par conséquent, lorsqu'un utilisateur demande une page spécifique plusieurs fois, le contenu est récupéré du cache, évitant ainsi l'exécution de code de programmation sur le serveur.

Le cache de sortie est un bon complément aux mises à jour partielles des pages AJAX. Le cache de sortie et les mises à jour partielles des pages réduisent la charge de travail sur le serveur et augmentent le nombre de requêtes utilisateur qu'un serveur peut traiter.

Dans ASP.NET MVC 5, vous pouvez implémenter la mise en cache de sortie en ajoutant l'attribut OutputCache au contrôleur.

**Configuration du cache de sortie**

[OutputCache (Durée = 60)]

public PartialViewResult HelloWorld ()

{

ViewBag.Message = "Bonjour tout le monde";

return PartialView ();

}

L'attribut OutputCache permet de diriger le moteur de rendu vers le cache qui contient les résultats du processus de rendu précédent. Le paramètre Duration de l'attribut OutputCache permet de contrôler la période en secondes pendant laquelle les données doivent être stockées dans le cache.

Par défaut, le cache de sortie stocke une seule copie du contenu rendu, pour chaque vue. Considérez une vue avec le paramètre d'entrée QueryString qui permet au contenu de changer en fonction de la variable collectée à partir de la base de données ou d'une demande antérieure. Dans ce cas, vous pouvez ajouter la propriété VaryByParam à l'attribut OutputCache, pour stocker une copie de chaque combinaison unique de paramètres dans le cache.

**Mise en cache par valeur de paramètre**

[OutputCache (Duration = 60, VaryByParam = "ID")]

public PartialViewResult HelloWorld ()

{

ViewBag.Message = "Bonjour tout le monde";

return PartialView ();

}

Dans l'exemple précédent, observez que la propriété VaryByParam fait référence à QueryString, au lieu d'autres paramètres MVC. Vous pouvez également utiliser la propriété VaryByCustom.

**Utilisation de VaryByCustom**

[OutputCache (Duration = 60, VaryByCustom = "browser")]

public PartialViewResult HelloWorld ()

{

ViewBag.Message = "Bonjour tout le monde";

return PartialView ();

}

Vous pouvez ajouter un navigateur comme paramètre d'entrée à la propriété VaryByCustom. Le paramètre de navigateur permet de stocker une copie du contenu rendu correspondant à chaque navigateur sur lequel l'application est exécutée. Si vous souhaitez contrôler et implémenter votre propre logique pour déterminer quand une nouvelle copie doit être stockée, vous devez remplacer la fonction GetVaryByCustomString dans le fichier Global.asax de votre projet.

**Question**: En quoi le fonctionnement d'une application Web qui implémente le cache de sortie diffère-t-il d'une application qui n'implémente pas de cache de sortie?

## **Le cache de données**

Les applications Web dépendent généralement du contenu d'une base de données, pour rendre le contenu sur une page Web. Les bases de données rencontrent parfois des problèmes de performances dus à des requêtes mal rédigées, ce qui peut ralentir les performances des requêtes de base de données et entraîner de mauvaises performances de la page Web. Vous pouvez implémenter le cache de données dans votre application Web pour éviter de charger des données à partir d'une base de données chaque fois qu'un utilisateur place une demande. La classe MemoryCache vous permet d'implémenter le cache de données dans votre application Web. La mise en œuvre du cache de données implique les actions suivantes:

|  |  |
| --- | --- |
| 1. | Chargement des informations de la base de données |
| 2. | Stockage du contenu dans l'objet MemoryCache |
| 3. | Récupération des données de l'objet MemoryCache |
| 4. | S'assurer que le contenu est disponible dans l'objet MemoryCache; sinon, recharger le contenu |

**Utilisation du cache mémoire**

System.Data.DataTable dtCustomer = System.Runtime.Caching.MemoryCache.Default.

AddOrGetExisting ("CustomerData", this.GetCustomerData (),

System.DateTime.Now.AddHours (1));

Dans l'exemple précédent, les paramètres suivants sont spécifiés:

|  |  |
| --- | --- |
| • | Clé. L'identifiant unique de l'objet qui doit être stocké dans le cache mémoire. |
| • | Valeur. L'objet qui doit être stocké dans le cache mémoire. |
| • | AbsoluteExpiration. L'heure à laquelle le cache doit expirer. |

Vous pouvez utiliser la fonction AddOrGetExisting, au lieu de la fonction Ajouter, pour permettre à l'application d'actualiser et de récupérer des données dans une ligne de code. Si le cache contient les données pertinentes, la fonction AddOrGetExisting récupère les données du cache. Si le cache ne contient pas les données pertinentes, la fonction AddOrGetExisting permet d'ajouter les données au cache, puis de rendre les mêmes données sur la page Web.

**Question**: Quels sont les avantages de l'implémentation de la mise en cache des données dans les applications MVC?

## **Le cache HTTP**

Vous pouvez implémenter la mise en cache HTTP dans le cache du navigateur et le cache du proxy.

**Le cache du navigateur**

La plupart des navigateurs Web stockent le contenu téléchargé à partir des serveurs Web dans leur cache local. Le stockage des données dans le cache local permet de supprimer le besoin de télécharger à plusieurs reprises du contenu à partir du serveur. Les navigateurs Web vérifient fréquemment le contenu pour les mises à jour. Si le contenu est mis à jour sur le serveur, les navigateurs Web téléchargent le contenu du serveur pour répondre aux demandes des utilisateurs. Sinon, les navigateurs Web affichent le contenu à partir du cache local.

**Le cache proxy**

La fonctionnalité du cache proxy est similaire à la fonctionnalité du cache du navigateur. Cependant, le cache est stocké sur un serveur centralisé. Les utilisateurs peuvent se connecter à Internet ou à des serveurs Web en utilisant ce serveur proxy. Les serveurs proxy stockent une copie d'une application Web d'une manière similaire à un navigateur Web stockant une copie d'une application sur les lecteurs locaux. De nombreux utilisateurs peuvent accéder au cache dans un serveur proxy, tandis qu'un seul utilisateur peut accéder au cache du navigateur à la fois.

**Question**: Quelle est la différence entre le cache HTTP et le cache de sortie?

## **Empêcher la mise en cache**

La mise en cache peut parfois créer des problèmes dans les applications Web, car si une application implique des mises à jour de contenu fréquentes, la mise en cache empêche les utilisateurs d'afficher ces mises à jour de contenu. Pour résoudre les problèmes de mise en cache, vous pouvez implémenter un en-tête HTTP appelé Cache-Control. L'en-tête Cache-Control indique au navigateur Web comment gérer le cache local. Tous les clients HTTP, tels que les navigateurs et les serveurs proxy, répondent aux instructions fournies dans l'en-tête Cache-Control pour déterminer comment gérer le cache local d'une application Web.

**Lecture supplémentaire:**Pour plus d'informations sur la définition de cet en-tête, accédez à:<http://go.microsoft.com/fwlink/?LinkID=288972&clcid=0x409>

Vous pouvez utiliser la méthode HttpCachePolicy.SetCacheability pour spécifier la valeur de l'en-tête Cache-Control. La méthode HttpCachePolicy.SetCacheability permet de contrôler les performances de la mise en cache.

**Utilisation de SetCacheability**

Response.Cache.SetCacheability (HttpCacheability.Private);

Dans l'exemple précédent, la méthode HttpCachePolicy.SetCacheability prend la valeur d'énumération Private.

Pour empêcher la mise en cache dans votre application Web, vous devez définir la valeur d'en-tête Cache-Control sur NoCache.

**Empêcher la mise en cache**

Response.Cache.SetCacheability (HttpCacheability.NoCache);

**Question**: Quels scénarios vous obligeraient à empêcher la mise en cache pour une application Web?

## **Démonstration: comment configurer la mise en cache**

Dans cette démonstration, vous verrez:

|  |  |
| --- | --- |
| • | Comment configurer le cache de sortie pour une action de contrôleur MVC. |
| • | Mesurez la différence qu'apporte le cache configuré à la livraison de la page. |

### **Étapes de démonstration**

|  |  |
| --- | --- |
| • | Vous trouverez les étapes dans la section «Leçon 2: Mettre en œuvre une stratégie de mise en cache» sur la page suivante:<https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD09_DEMO.md>. |

# **Atelier: Création de pages réactives dans les applications Web ASP.NET MVC 5**

### **Scénario**

Votre responsable vous a demandé d'inclure des commentaires sur les photos dans l'application de partage de photos. Votre responsable a également souligné que les performances de certaines pages de l'application sont trop lentes pour un site de production.

Vous voulez vous assurer que les commentaires des photos prennent un temps de chargement minimal, pour lequel vous décidez d'utiliser des mises à jour partielles de la page. Vous souhaitez également renvoyer les pages rapidement, pendant que des informations mises à jour sont affichées, pour lesquelles vous décidez de configurer la mise en cache dans votre application.

### **Objectifs**

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

|  |  |
| --- | --- |
| • | Ecrivez des actions de contrôleur qui peuvent être appelées de manière asynchrone et renvoyer des vues partielles. |
| • | Utilisez les aides AJAX courantes pour appeler des actions de contrôleur asynchrones et insérez les résultats dans les vues Razor. |
| • | Configurez les caches ASP.NET pour servir les pages rapidement. |

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

Durée estimée: 70 minutes

Vous trouverez les étapes de haut niveau sur la page suivante:<https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD09_LAB_MANUAL.md>.

Vous trouverez les étapes détaillées sur la page suivante:<https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD09_LAK.md>.

### **Exercice 1: Utilisation des mises à jour partielles de page**

##### **Scénario**

Vous avez été invité à inclure une fonctionnalité de commentaire sur la vue d'affichage des photos de l'application de partage de photos. Vous souhaitez garantir des performances élevées en utilisant les mises à jour partielles des pages AJAX.

Dans cet exercice, vous allez:

|  |  |
| --- | --- |
| • | Importez un contrôleur partiellement complet pour ajouter des commentaires et une vue pour supprimer des commentaires. |
| • | Ajoutez du code au contrôleur pour une mise à jour partielle de la page. |

### **Exercice 2: configuration facultative des caches ASP.NET**

##### **Scénario**

Il vous a été demandé de configurer les caches ASP.NET dans l'application de partage de photos pour garantir des performances optimales. Les développeurs expérimentés craignent particulièrement que la galerie Toutes les photos puisse être rendue lentement car elle récupère et affiche de nombreuses photos de la base de données à la fois.

Dans cet exercice, vous allez:

|  |  |
| --- | --- |
| • | Configurez le cache de sortie pour stocker la vue de l'index des photos. |
| • | Utilisez les outils de développement d'Internet Explorer pour examiner la vitesse à laquelle les fichiers image et les pages sont rendus avec et sans mise en cache. |
| • | Configurez le cache de sortie pour stocker les résultats de l'action GetImage afin que les fichiers image puissent être renvoyés à partir du cache. |

Effectuez cet exercice si le temps le permet.

### **Question (s) de révision**

**Vérifiez vos connaissances**

**Découverte**

**Dans l'exercice 2, pourquoi le délai de demande pour / Photo n'a-t-il pas été réduit pour la première demande lorsque vous avez configuré le cache de sortie pour l'action d'index?**

Afficher la solution Réinitialiser

**Vérifiez vos connaissances**

**Découverte**

**Dans l'exercice 2, lorsque vous avez configuré le cache de sortie pour l'action GetImage (), pourquoi était-il nécessaire de définir VaryByParam = "id"?**

Afficher la solution Réinitialiser

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

Dans ce module, vous avez utilisé AJAX et des mises à jour partielles de page dans les applications MVC. AJAX et les mises à jour partielles de la page aident à réduire le besoin de recharger la page entière, lorsqu'un utilisateur place une demande. Les mises à jour partielles de page réduisent également le besoin d'écrire plusieurs lignes de code, pour mettre à jour des parties spécifiques d'une page Web. Vous avez également utilisé la mise en cache pour augmenter les performances d'une application Web.

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

Les applications Web exécutent généralement plusieurs requêtes pour récupérer des informations à partir d'une base de données et afficher le contenu sur les pages Web. Les utilisateurs se plaignent parfois du fait que les pages Web prennent plus de temps à se charger. Par conséquent, les développeurs implémentent la mise en cache dans l'application Web, pour réduire le besoin de charger des données à partir d'une base de données, chaque fois qu'un utilisateur place une demande. La mise en cache permet aux pages Web de se charger plus rapidement, augmentant ainsi les performances de l'application.

### **Question (s) de révision**