



Dotnet France
Technologies Sharepoint, SQL Server & .NET

Association Dotnet France

ASP .NET MVC : présentation

Version 1.0



James RAVAILLE

<http://blogs.dotnet-france.com/jamesr>

MCours.com

Sommaire

1	Introduction.....	3
1.1	Présentation	3
1.2	Pré-requis	3
2	Le modèle MVC	4
2.1	Un peu d’histoire	4
2.2	Présentation générale	4
2.3	Présentation détaillée	5
2.4	MVC dans le Framework .NET	5
3	Pages Web VS Pages MVC	6
3.1	Caractéristiques des pages Web	6
3.2	Caractéristiques des pages MVC	6
3.3	Fin des pages Web ?	6
4	Conclusion	8

1 Introduction

1.1 Présentation

Dans ce cours, nous allons vous présenter les concepts de base d'ASP .NET MVC, qui constitue une alternative pour la conception et le développement de pages Web avec la technologie ASP .NET. D'un point de vue macroscopique, il permet de séparer la présentation des données, la définition du modèle de données et la gestion des demandes de l'utilisateur. De ce fait, les applications ASP .NET MVC sont structurées différemment par rapport à une application ASP.NET classique, tout comme les pages ASP .NET qu'elle contient.

Nous présenterons les éléments constituant le modèle MVC, ainsi que les différences entre ce modèle de développement et celui des Web Forms classiques que nous utilisons jusqu'à maintenant avec la technologie ASP .NET.

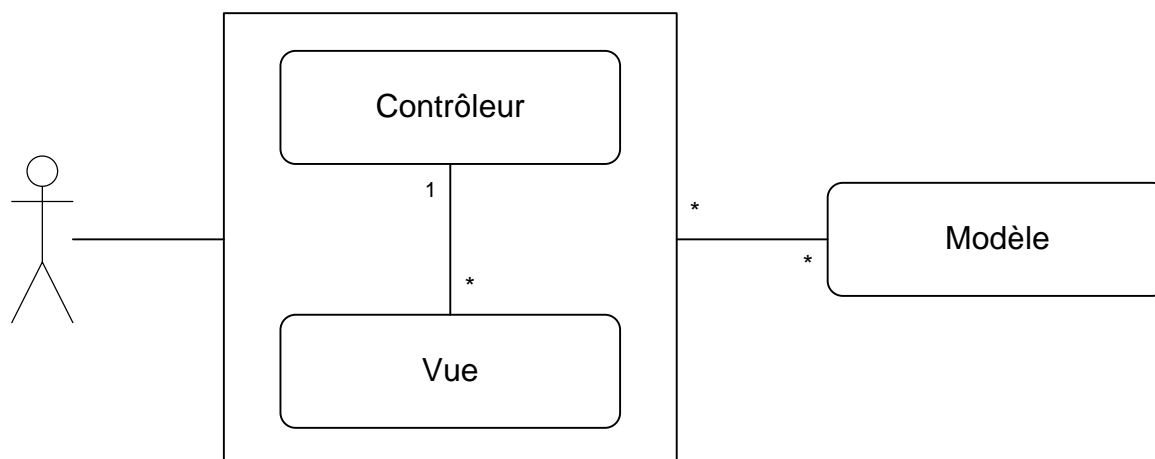
1.2 Pré-requis

Avant de lire ce cours, nous vous conseillons de maîtriser la conception et le développement d'applications Web avec la technologie ASP .NET.

2 Le modèle MVC

2.1 Un peu d'histoire

Le modèle MVC (Modèle Vue Contrôleur) a fait son apparition en 1978. Un an plus tard, Trygve Reenskaug le présente dans un projet Smalltalk en les termes suivants : « Le but principal est de combler l'écart entre la représentation humaine du modèle et le modèle digital dans l'ordinateur ». Il établit le diagramme suivant :



L'utilisateur ne voit qu'une vue, qui fait appel à un contrôleur (lui-même pouvant être utilisée par plusieurs vues), lui-même faisant appel au modèle pour réaliser des traitements de données. Le résultat de ces traitements sera récupéré par la vue qui les affichera.

2.2 Présentation générale

Le modèle MVC permet de créer une application permet de séparer les différentes parties constituant une application :

- La présentation des données et les interactions entre l'application et les utilisateurs.
- Le modèle métier, définissant les données manipulées par l'application, et les traitements métiers.
- L'interaction entre les éléments de présentation de données et le modèle métier.

Cette séparation permet de coupler faiblement chacune de ces parties entre elles. Elle permet :

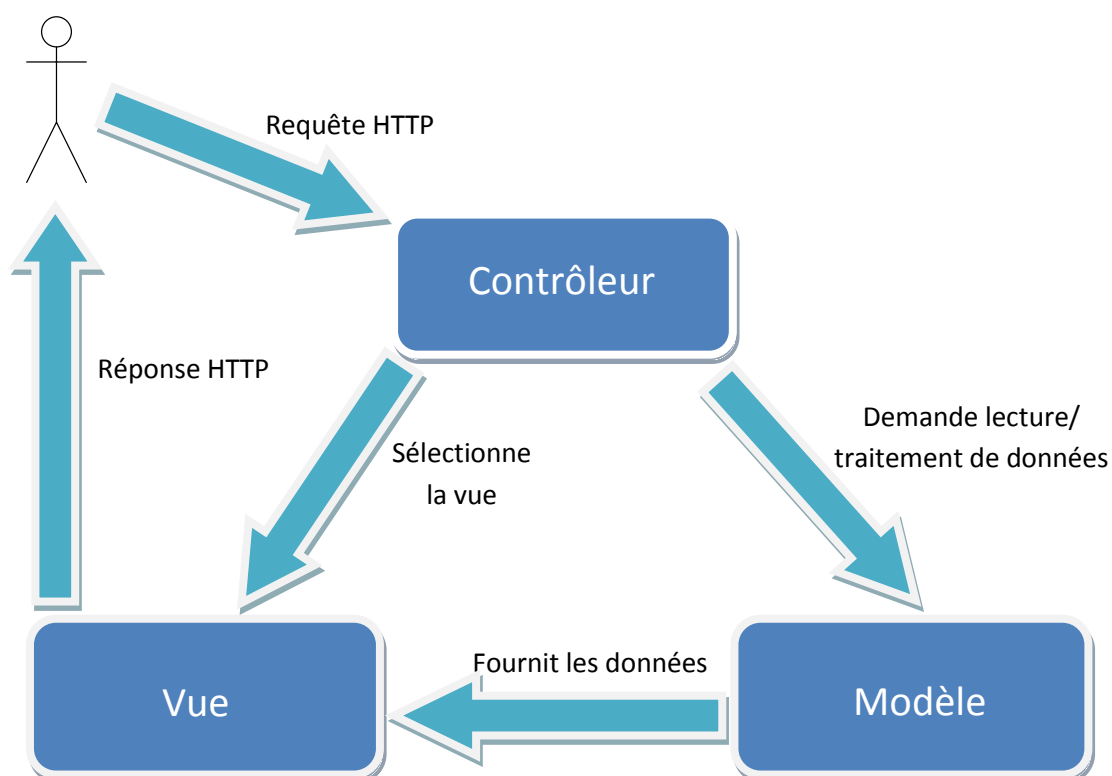
- De faciliter le développement de l'application, afin de répartir des tâches de conception et de développement de l'application entre les différentes personnes d'une équipe de développement.
- De bien structurer l'application, afin de faciliter son développement, ainsi que sa maintenance.
- De faciliter les tests de l'application, afin de mieux réaliser les tests unitaires, fonctionnels et de non-régression.

2.3 Présentation détaillée

Le modèle MVC est constitué des éléments suivants :

- Le **Modèle** : représente la couche métier d'une application, présentant des classes permettant de créer les objets contenant des données métier manipulées par l'application au travers de traitements, constituant les services métiers.
- La **Vue** : elle constitue les éléments d'interface utilisateurs : pages web, contrôles Web...
- Le **Contrôleur** : permettant de piloter l'application, il interprète les actions à réaliser et ordonne leur exécution (lecture, traitement de données et mises à jour).

Les relations entre ces trois éléments sont les suivantes :



2.4 MVC dans le Framework .NET

Pour mettre en œuvre le modèle MVC, Microsoft a ajouté un nouvel espace de noms dans le Framework .NET, nommé System.Web.Mvc. Il contient toutes les classes et interfaces permettant de mettre en œuvre le modèle MVC : classes de base pour les contrôleurs, classes pour les vues, classes désignant les actions, permettant de créer des liaisons de données entre la vue et le modèle...

On trouvera aussi l'espace de noms System.Web.Mvc.Ajax, permettant de mettre en œuvre des mécanismes Ajax dans les pages ASP .NET MVC.

3 Pages Web VS Pages MVC

La structure des applications ASP .NET « classiques » et les applications ASP .NET MVC sont radicalement différentes, tout en ayant le même but : construire des applications Web.

3.1 Caractéristiques des pages Web

Les caractéristiques des pages Web sont les suivantes :

- Elles proposent un modèle évènementiel, avec prise en charge de l'état d'affichage des contrôles entre l'exécution de deux requêtes HTTP via le *ViewState* (champ caché et crypté permettant de reporter dans la réponse HTTP les données provenant de la requête). Lors de l'exécution du cycle de vie de la page (où un évènement correspond à une étape), le rendu de la page est obtenu et renvoyé au client Web ayant demandé son exécution.
- Elles utilisent le modèle *Page Controller*, permettant la séparation entre la présentation (XHTML / CSS / JavaScript) de la page et sa logique d'exécution (langage .NET). En effet, à toute page ASPX est associée une classe dite d'arrière plan (*code-behind*) contenant ce qui pourrait s'apparenter de manière grossière au contrôleur de la page.
- Permettent de développer des applications avec un faible effort de développement, grâce à un « développement orienté composant ». En effet, des tâches peuvent être réalisées très simplement en ajoutant des composants dans des pages, puis en les paramétrant au travers de la fenêtre de propriétés ou des assistants.

3.2 Caractéristiques des pages MVC

Les caractéristiques des pages Web sont les suivantes :

- Elles permettent de structurer davantage l'application.
- Elles n'utilisent pas le *ViewState*.
- Elles utilisent un modèle nommé *Front Controller*, qui permet de traiter les requêtes de l'application Web par l'intermédiaire du routage et d'un contrôleur.
- Elles offrent un meilleur support pour le développement dirigé par les tests (Test Driven Development). Lors de la création d'un projet de type SP .NET MVC, Microsoft propose de créer un projet de test, qui permettra de tester au fur et à mesure du développement de l'application ASP .NET MVC.

3.3 Fin des pages Web ?

Le modèle MVC est un modèle de développement qui a fait ses preuves dans d'autres technologies de développement d'applications Web telles que la technologie J2EE. Quand aux Web Forms ASP .NET telles que nous les concevons depuis nos premiers pas avec la technologies ASP .NET en 2002, la vue (page ASPX) manque de souplesse dans le sens où elle doit souvent contenir trop de traitements ne facilitant pas le découpage en couche de l'application. Le modèle MVC permet d'agir sur la conception et le développement de l'application en assurant un découpage en couche. Mais sa contrepartie réside dans un développement plus coûteux en temps.

Le choix entre les pages Web et les pages MVC se fera en fonction de l'évaluation des critères suivants : nombre de pages Web dans le projet, exigence en matière d'architecture (la présence d'un

architecte étant souhaitée), connaissances des développeurs en matière d'architecture, temps alloué pour développer le projet ...

4 Conclusion

Ce cours vous a présenté le modèle MVC et ses caractéristiques, ainsi que les différences entre les pages ASP .NET « classiques » et les pages ASP .NET MVC. Avant d'aller plus loin dans la mise en œuvre du modèle MVC via les cours publiés sur Dotnet-France, il est essentiel de bien comprendre le rôle des trois acteurs qui le composent : le modèle, la vue, le contrôleur. C'est pourquoi, nous vous conseillons de relire ce cours dans le cas où ces éléments et leurs interactions ne sont pas pleinement compris.