ASP.NET Core MVC est une infrastructure riche pour la création d’applications web et d'APIs à l’aide du modèle de conception MODEL-VIEW-CONTROLLER.

Le MVC Tire son nom de la part d’un DESIGN PATTERN très connu qui existe depuis 1970.

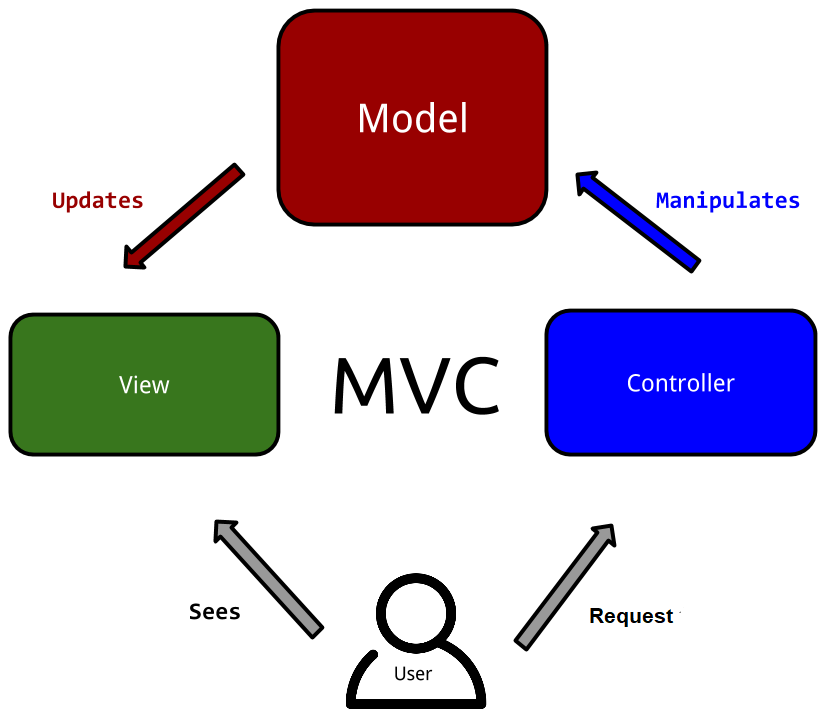
**Quel est le modèle de conception MVC ?**

Le modèle d’architecture MODEL-VIEW-CONTROLLER (MVC) sépare une application en trois groupes de composants principaux :

* M : MODEL : Le MODEL dans une application MVC représente l’état de l’application et de n'importe quelles logiques ou opérations qui doivent être effectuées par celui-ci, autrement dit c’est la manipulation et la gestion de données.
* V : VIEW : La VIEW est chargée de présenter du contenu via l’interface utilisateur. Elle utilise un [moteur d’affichage](https://docs.microsoft.com/fr-fr/aspnet/core/mvc/overview#razor-view-engine) pour incorporer le code .NET dans le balisage HTML. Donc c’est une sorte de présentation ou bien un IU pour l’utilisateur.
* C : CONTROLLER : Le CONTROLLER c’est le composant qui gère les interactions avec l’utilisateur, utilise le MODEL et finalement sélectionne une vue à restituer. Dans une application MVC, la VIEW affiche uniquement les informations ; le CONTROLLER gère et répond à la saisie de l’utilisateur et à l’interaction. Dans le modèle MVC, le CONTROLLER est le point d’entrée initial, il est chargé de sélectionner les types de MODEL avec lesquels travailler et la VIEW à restituer (d'où son nom - il contrôle la manière dont l’application répond à une requête donnée).

**Comment ça fonctionne le MVC ?**

Voilà un diagramme qui explique comment le MVC fonctionne et aussi la liaison entre ses composent



Dans le diagramme on peu voir que le MVC reçoit une requête de la part d’un utilisateur à partir du CONTROLLER et la dirigé vers le MODEL pour que ce dernier puisse manipuler les donnés et faire une mise a jour a la VIEW qui va montrer les données traitées pour l’utilisateur souvent comme une page HTML.

**ASP .NET Core MVC**

L’infrastructure ASP.NET MVC Core est un Framework de présentation léger, open source, facilement testable et optimisé pour une utilisation avec ASP.NET Core.

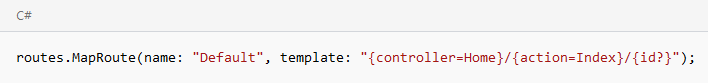
ASP.NET Core MVC offre un fonctionnement basé sur des modèles pour créer des sites Web dynamiques qui permettent une séparation claire des préoccupations. Il vous donne un contrôle total sur le balisage, prend en charge les développements TDD et utilise les standards web les plus récents.

## Fonctionnalités

ASP.NET Core MVC inclut les éléments suivants :

* **Routage :**

ASP.NET Core MVC est construit sur [le routage d'ASP.NET Core](https://docs.microsoft.com/fr-fr/aspnet/core/fundamentals/routing), un composant de mappage d’URL puissant qui vous permet de créer des applications ayant des URLs compréhensibles et découvrables. Cela vous permet de définir les modèles de noms d’URL de votre application qui fonctionnent bien pour l’optimisation des moteurs de recherche et pour la génération de lien, sans prendre en compte la façon dont les fichiers sont organisés sur votre serveur web. Vous pouvez définir vos routes à l’aide d’une syntaxe pratique de modèle de routes (Route Template) qui prend en charge les contraintes de valeur, les valeurs par défaut et les valeurs facultatives de routes.



Cette Template montre à l’application comment accès à une url par trouver le Controller et l’action, cette dernière est simplement une méthode publique dans le Controller, pour le paramétrer id ? sa montre qu’id là c’est optionnel dans l’url même si on n’ajoute pas dans l’url on va toujours y accéder.

Il existe aussi le Routage par attribut (Attribute Base Routing) qui utilise C# attribues sur les CONTROLLERS classes et méthodes qui permet à ASP .Net MVC quand d’appeler une méthode spécifique.

