

ASP.NET Web Forms

Historiquement, la technologie ASP.NET Web Forms (tel que vue dans votre cours de développement multimédia de la deuxième session) fut développée pour faciliter la migration des développeurs d'applications Windows au développement d'applications Web. Pour parvenir à cela, un certain nombre de mécanismes tel que le ViewState, lePostBack, les contrôles serveurs et le cycle de vie complexe d'une page (pour ne nommer que ceux-ci!) ont été mis en place de façon à **cacher aux développeurs les mécanismes de communication entre le client et le serveur web.**

Malheureusement, cette technologie possède beaucoup d'inconvénients importants. En voici quelques un:

- Perte de contrôle du code HTML (rend difficile l'application de nouveaux standards);
- Volume important de données transférées sur le Web entre le client et le serveur à chaque clique, grâce au ViewState;
- Les URL sont difficilement accessibles, ce qui rend les fonctionnalités du site Web indissociables de ce dernier (réutilisation des composantes presque impossible) et complique l'accès des moteurs de recherche au contenu;
- La mise à jour de l'affichage des données (HTML) requiert le plus souvent l'interaction d'un programmeur.

Dans le cadre de ce cours, nous allons aller de l'avant avec l'utilisation de la technologie **ASP.NET MVC** qui palie à la plupart des inconvénients de la technologie ASP.NET Web Forms et qui, selon nous, va être appelée à la remplacer (et l'on a espoir que ce sera bientôt!).

Rappel de notions fondamentales sur le fonctionnement du WEB

Pour bien comprendre la technologie ASP .NET MVC, nous devons revenir à la base du fonctionnement du Web puisque cette dernière n'encapsule pas les mécanismes de communication entre le client et le serveur web.

Quelques rappels sur le Web:

- Tim Berners Lee invente, en 1990, le world wide web (www)
- But : Échanger des documents, du texte et des images
- W3C (World Wide Web Consortium) : (<http://www.w3.org/>) consortium qui règlemente le Web, établit les standards.

Fonctionnement : Série de connections et de déconnections : on accède à une page texte (html) (connexion au serveur, déconnexion), à partir de celle-ci on peut naviguer vers d'autres pages (connexion au serveur, déconnexion)

Protocole utilisé pour accéder aux sites Web est le **HTTP** (HyperText Transfer Protocol).

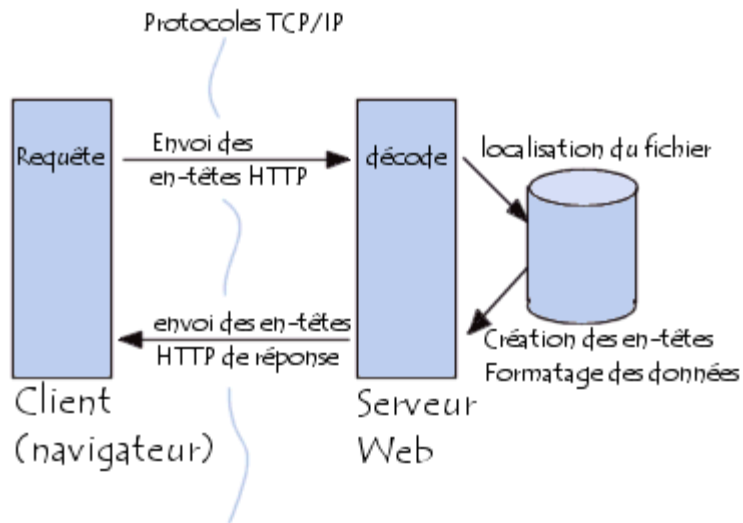
La version actuellement utilisée du protocole HTTP est 1.1 et cette version est considérée comme mature et stable, et répond parfaitement au besoin initial. Peu (voire pas) d'évolutions du protocole sont à attendre. D'ailleurs, le W3C a fermé l'activité http. Les efforts sont désormais consacrés à des protocoles de plus haut niveau (basés sur http) tel que les services Web.

Voici les différentes méthodes offertes par le protocole http, avec la version de HTTP dans laquelle elles sont apparues :

requête Version/HTTP		description
GET	HTTP/0.9	obtient un document
HEAD	HTTP/1.0	obtient l'en-tête de la réponse
POST	HTTP/1.0	envoie du contenu au serveur

Port de communication : 80

Illustration du fonctionnement du protocole http :



- Le modèle requête-réponse : il faut bien comprendre les rôles du client et du serveur HTTP et ce que chacun peut faire (i.e. quelles technologies chacun peut comprendre et utiliser). Il faut se rappeler que c'est toujours le client qui initie les échanges avec le serveur. Deux conséquences: si on veut que le serveur envoie quelque chose au client, il faut d'abord que ce dernier contacte le serveur et, sans mécanisme spécifique particulier, le serveur n'exécutera rien sans sollicitation de la part d'un client.
- Pas de connexion permanente: cela implique une sorte de fonctionnement en mode déconnecté d'un point de vue applicatif : d'une requête à l'autre, le serveur ne reconnaît pas le client. Il est donc plus difficile dans ces conditions de faire une application transactionnelle avec HTTP qu'avec un système client-serveur traditionnel...

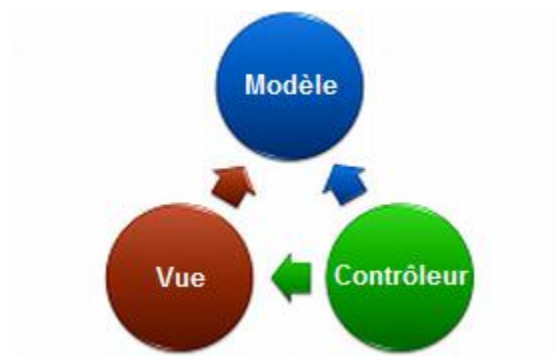
Patron de conception MVC

Le patron de conception **Modèle-Vue-Contrôleur** (en abrégé **MVC**) est un patron de conception architectural qui découpe une application en trois types de composants:

- les modèles (contiennent aussi bien des données que des opérations) ;
- les vues (responsables de la présentation aux utilisateurs) ;
- des contrôleurs (gèrent les événements et la synchronisation entre les vue et les modèles).

MVC a été mis au point en 1979 par Trygve Reenskaug qui travaillait alors sur SmallTalk.

Voici un schéma représentant l'interaction possible entre ces trois composants (modèle, vue et contrôleur) :



ASP.NET MVC

ASP.NET MVC est un Framework de développement d'application Web, basé sur le patron de conception MVC.

Voici les trois composants de MVC tel qu'implémentés par ASP.NET MVC :

Le Contrôleur est une classe exposant un certain nombre de fonctions (ou actions) renvoyant un objet de type *ActionResult*. Une Action sera toujours le point d'entrée dans ASP.NET MVC, et pourra renvoyer, au choix, une page HTML, des données ou rien du tout.

Les contrôleurs sont stockés dans le répertoire *Controllers* de l'application Web, et utilisent comme convention de nommage <Objet>Controller.

La Vue est une page Web qui affiche les données fournies par un modèle.

Les vues sont stockées dans le répertoire *Views* de l'application Web. Toutes les vues qui seront renvoyées par un contrôleur nommé <Objet>Controller doivent être stockées dans le répertoire *Views/<Objet>*.

Le modèle est un ensemble de classes dont le but est de gérer un ensemble de données (accès, ajout, validation, mise à jour, suppression) ou d'opérations afin de les rendre disponibles pour les contrôleurs ainsi que pour les vues. Les modèles sont stockées dans le répertoire *Models*.

Explications des autres dossiers :

- le dossier *App_Data* possède le même rôle que pour les sites ASP.NET Web Forms et permet de stocker physiquement des données;
- le dossier *Content* contient en général les fichiers statiques. Il est recommandé d'y ajouter les fichiers tels que les feuilles de style CSS et les images;
- le dossier *Scripts* pour stocker les fichiers de scripts utilisés par l'application Web;