

Introduction ASP.NET

Sommaire

Introduction.....	1
1 Framework	2
1.1 Général (.NET)	2
1.2 ASP.NET	2
2 Prérequis	2
3 Présentation des bases d'un projet ASP.NET	3
3.1 Création d'un projet	3
3.2 Site Web : Les différentes méthodes d'accès.....	5
3.2.1 Fichier:	6
3.2.2 FTP:	6
3.2.3 HTTP:	6
3.3 <i>Les en-têtes / directives</i>	6
3.4 Behind.....	7
3.5 Using / Import	9
4 Comprendre le fonctionnement d'une requête.....	9
4.1 Les intervenants et leur dialogue	9
4.2 Rôle du navigateur (client)	10
4.3 Rôle du serveur.....	10
4.4 HyperText Transfer Protocol	10
5 La configuration.....	11
5.1 Accueil	11
5.2 Sécurité.....	12
5.2.1 Authentification.....	13
5.2.2 Rôles	Erreur ! Signet non défini.Erreur ! Signet non défini.
5.2.3 Règles de sécurités	Erreur ! Signet non défini.Erreur ! Signet non défini.
5.3 Application.....	Erreur ! Signet non défini.Erreur ! Signet non défini.
5.4 Fournisseur	Erreur ! Signet non défini.Erreur ! Signet non défini.

Vous allez suivre un cours sur le développement en ASP.NET. Ce premier document expliquera les bases du Framework. Le logiciel et les connaissances nécessaires pour suivre le cours sont considérés comme acquis.

1 Framework

1.1 Général (.NET)

Un Framework est, en général, un espace de travail modulaire où l'on trouve des bibliothèques et des conventions / règles de développement qui vont permettre un développement normalisé d'une application (dans notre cas application Web). Il permet ainsi un développement plus rapide, plus performant et facile à maintenir ceci grâce à une implémentation de fonctionnalités récurrentes (par exemple gestion de fenêtres, communication à une base de données ...).

Dans le cas du Framework .NET, le code est compilé en langage MSIL. Cela entraîne une amélioration de la rapidité d'exécution. Tout langage qui se compile en MSIL (le MSIL étant un langage intermédiaire avec le langage machine) peut être utilisé pour faire du .NET. Le premier langage développé exclusivement pour le .NET est le C#, le Visual Basic de Microsoft aillant été porté .NET un peu après. Plus tard ont été portés le java, le C++, le cobol, le python ...

1.2 ASP.NET

Le Framework .NET possède un ensemble de fonctionnalités dédiées à la création et à la gestion de sites Web. C'est le Framework qui interprète l'ASP.NET. L'ASP.NET permet de créer des sites Web dynamiques. Ainsi chaque page peut être unique et afficher des informations spécifiques à l'utilisateur. ASP.NET est à IIS ce que PHP est à apache c'est à dire un moyen de coder la partie logicielle au site.

L'ASP.NET permet la séparation des couches. La méthode des couches est très employée en entreprise pour permettre une meilleure organisation du code. Il est ainsi plus facile de le faire évoluer, de le maintenir et de le corriger. Cela consiste à isoler dans des dossiers / fichiers les couches, on appelle cela l'architecture n-tiers. L'organisation la plus utilisé est l'architecture 3-tiers : il y a une partie d'accès aux données, la partie métier applicative (des fonctions et des classes) et la partie interface.

2 Prérequis

Pour satisfaire vos besoins logiciels, nous vous suggérons l'utilisation de [Visual Web Developer Express Edition](http://www.microsoft.com/express/vwd/) (<http://www.microsoft.com/express/vwd/>).

Il y a d'autres IDE qui ne sont pas basés sur Visual Studio et qui permettent le développement en .NET...

Pour utiliser Visual Studio il faut avoir au moins 2 Go d'espace libre, un potentiel de 150Mo de mémoire vive utilisable et nous vous recommandons une résolution au moins égale à 1024*768, un Windows (2000, XP, Serveur 2003, Vista, Serveur 2008).

Même si parfois nous ré aborderons brièvement des bases si nous trouvons cela pertinent, nous admettons que vous connaissez le HTML et les concepts de la programmation orientée objet. Lorsque du code vous sera fourni, il le sera dans deux langages différents : en C# et en VB.NET. Pour que vous puissiez suivre ce cours si vous n'êtes pas au niveau, voici quelques liens qui vous aideront.

<http://jlambert.developpez.com/tutoriels/dotnet/motsclessharp/> => Les mots clef du C#

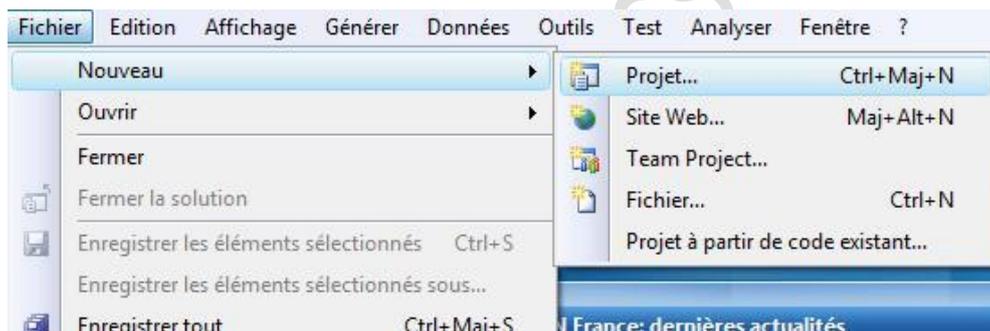
<http://plasserre.developpez.com/vsommair.htm> => Cours sur le VB.NET

3 Présentation des bases d'un projet ASP.NET

3.1 Création d'un projet

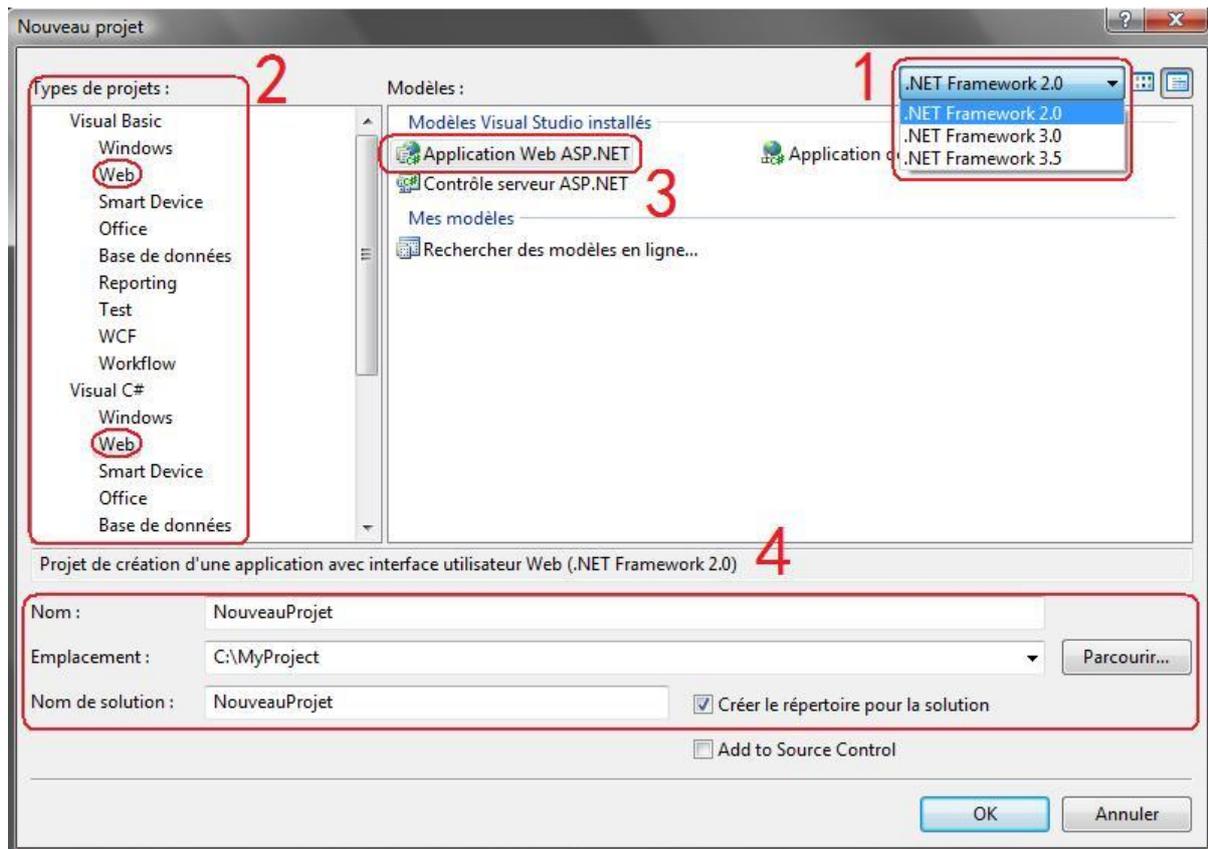
Nous allons expliquer comment ouvrir un nouveau projet dans Visual Studio 2008.

Une fois dans Visual Studio, faites *Fichier > Nouveau > Projet*.



Création d'un projet

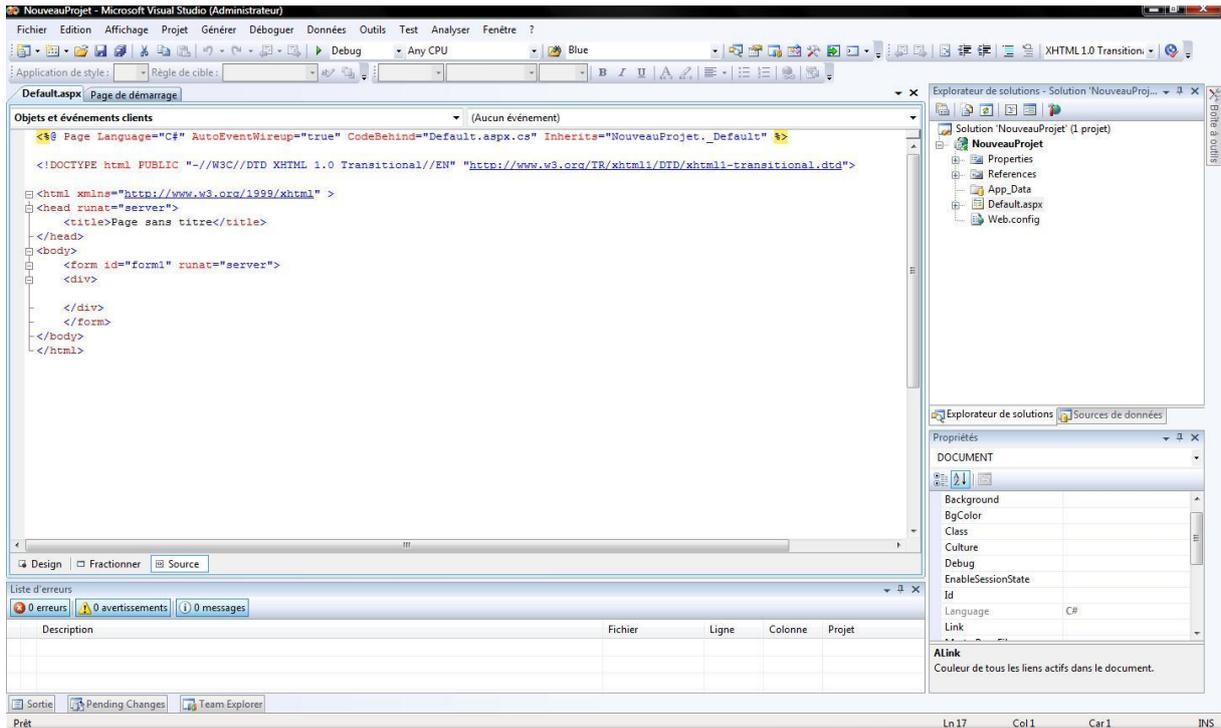
Vous arriverez alors sur une fenêtre comme celle là :



Fenêtre pour la création d'un nouveau projet

- 1 : Choisir le Framework souhaité (dans notre cas 2.0).
- 2 : Choisir son langage (VB.NET ou C#) et cliquer sur Web.
- 3 : Vérifier que « Application Web ASP.NET » est bien sélectionné.
- 4 : Entrer un nom de projet, choisir son emplacement et le nom de la solution (généralement on met le même nom de solution que celui du projet).
- 5 : Cliquer sur « Ok » en bas à droite

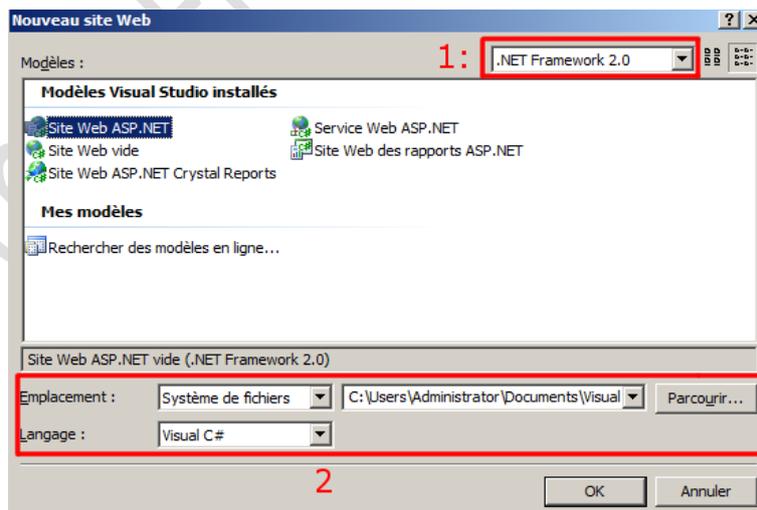
Et voilà vous venez de créer votre premier projet.



Page ASPX d'un projet par défaut

3.2 Site Web : Les différentes méthodes d'accès

Le début de la procédure étant commune pour tous les sites web quelle que soit sa méthode d'accès : Fichier > Nouveau > Site web, vous arrivez sur cette fenêtre (Les utilisateurs de Visual Studio 2005 n'auront pas le champ 1):



Création d'un site Web

Le champ 1 vous permet de choisir le Framework .NET cible, vous pouvez choisir le langage que vous utiliserait pour coder la partie dynamique (typiquement VB ou C#). Par la suite nous vous montrerons tous les cas possibles pour le cadre 2.

3.2.1 Fichier:

La création d'un site web expliquée auparavant (dans le 3.1) est la plus optimisée et la plus facile à partager, le code y est précompilé et moins étalé dans l'arborescence.

3.2.2 FTP:

Pour faire un site distant, on peut utiliser un accès ftp dans Visual Studio, ainsi le site est directement publié sur le serveur ce qui permet de faciliter les changements.

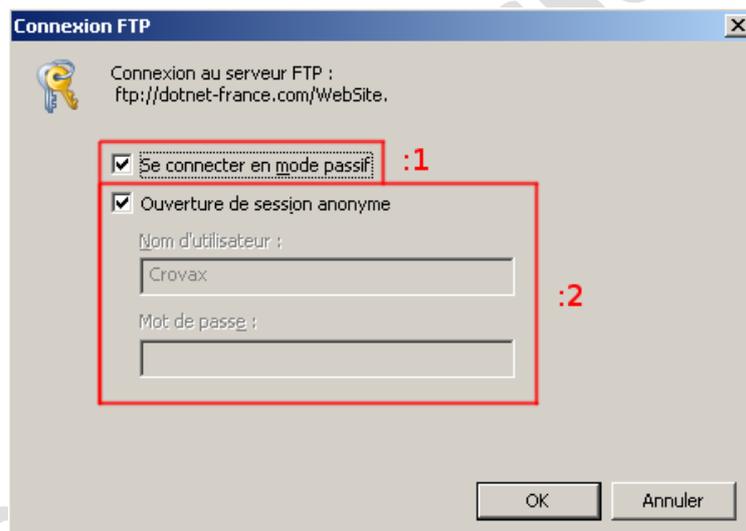
Il faut que le serveur FTP soit déjà existant et accessible par adresse IP ou nom de domaine. Pour y accéder il suffit de compléter la première fenêtre comme suit :



Configuration pour utiliser un FTP (à distance)

Renseignez le nom de domaine ou l'adresse IP de votre serveur FTP.

Lors de la première connexion, vous rencontrez cette fenêtre de connexion :



Fenêtre de connexion à un FTP

Cocher le mode passif (1) permet de n'utiliser que le port principal, utile dans le cas où le port en-dessous de celui utilisé pour faire passer la donnée est bloqué par un routeur ou un firewall.

Si vous décochez l'ouverture de session anonyme (2), vous pouvez insérer vos authentifiants pour bénéficier de vos droits sur le serveur.

3.2.3 HTTP:

Pour faire un site distant sur un serveur en passant par le protocole HTTP il est nécessaire que le serveur soit local ou qu'il ait les extensions "Front Page Server".

Il faut que le serveur http soit déjà existant et accessible par adresse IP ou nom de domaine. Pour y accéder, il suffit de compléter la première fenêtre comme suit:

Emplacement :	HTTP	http://dotnet-france.com	Parcourir...
Langage :	Visual C#		

Configuration pour utiliser le protocole http (un serveur à distance)

Entrez l'adresse DNS ou l'IP du serveur à la place de l'adresse http.

3.3 Les en-têtes / directives

Le Doctype est exactement le même qu'en HTML (que vous devez déjà connaître je le rappelle). Il ressemble à cela :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

C'est la ligne qui définit quelle version du HTML vous allez utiliser. Vous pouvez la changer exactement comme en HTML.

L'ASP.NET a aussi sa ligne qui définit certains paramètres et qui ressemble par défaut à ce qui suit (en C#) :

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="NouveauProjet._Default" %>
```

On appelle ceci une directive (plus précisément, il s'agit ici de la directive Page).

Il y est indiqué que nous allons utiliser le C# comme langage (`Language="C#"`), que le nom de la page du code behind, correspondant à cette page .aspx, s'appelle Default.aspx.cs (`CodeBehind="Default.aspx.cs"`).

Est aussi définit le namespace et le nom de la classe lié à la page (`Inherits="NouveauProjet._Default"`).

Nous verrons dans un autre chapitre à quoi sert `AutoEventWireup="true"`.

Remarque : En VB.NET la directive ressemble à ceci (exactement la même chose sauf le langage qui change) :

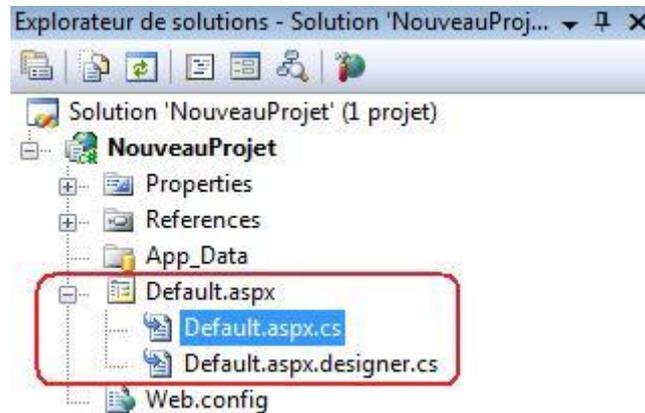
```
<%@ Page Language="vb" AutoEventWireup="true" CodeBehind="Default.aspx.vb"
Inherits=" NouveauProjet._Default " %>
```

3.4 Behind

Le code behind est le code dans lequel on écrira les instructions qui seront exécutées sur le serveur comme, par exemple, récupérer une valeur, changer un attribut dynamiquement, créer un objet

On remarque dans l'explorateur de solution qu'une page web par défaut a été créée (Default.aspx) et que si l'on déroule l'arborescence il y a deux autres pages qui y sont liées : le designer et une autre

page Default.aspx.cs (sinon faire un clic droit sur la page ASPX et faire "Afficher le code" pour accéder à Default.aspx.cs).

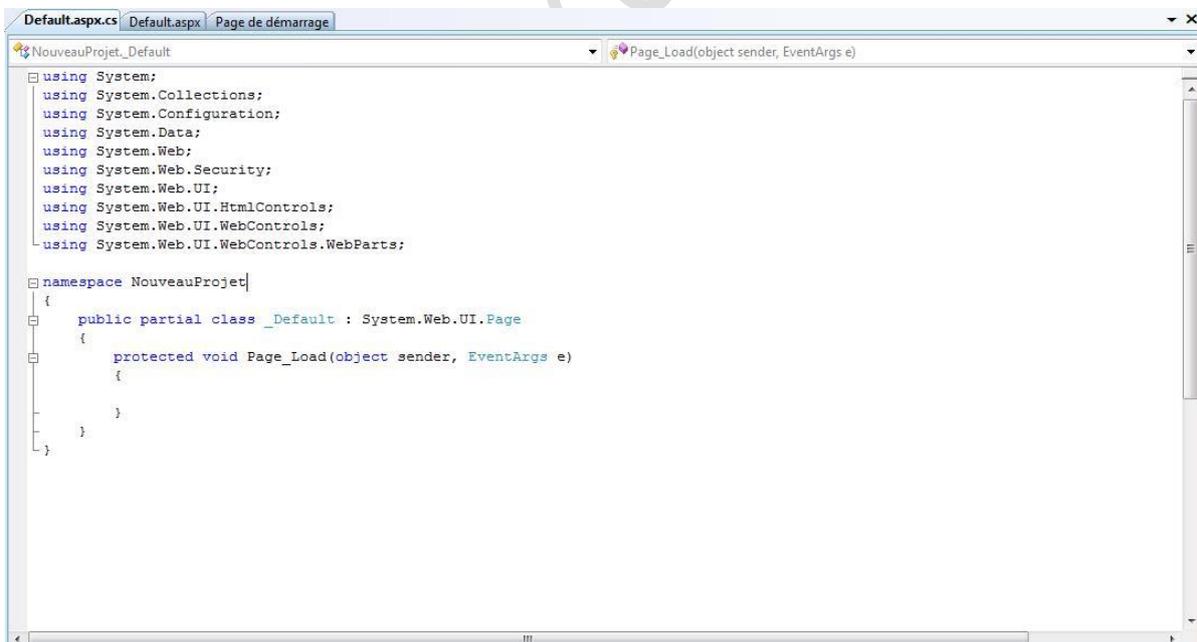


L'explorateur de solution

Default.aspx.designer.cs contient du code behind **génééré automatiquement**, pour l'instant nous n'en parlerons pas.

Default.aspx.cs contient le code behind que **vous** allez écrire. Voyons un peu à quoi il ressemble.

Pour cela double-cliquez dessus. Vous verrez ceci :



Le code behind

Nous voyons donc plusieurs lignes (en haut du fichier) avec des using que nous verrons dans la partie suivante de ce module. Le reste de la page est englobé dans le namespace NouveauProjet. Dans celui-ci se trouve la classe `_Default` héritant de `System.Web.UI.Page` qui correspond à la page aspx (rappelez-vous la page ASPX est lié au code behind) : `public partial class _Default : System.Web.UI.Page`

Voici ce que l'on retrouve dans la page ASPX :

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="NouveauProjet._Default" %>
```

Ceci est exactement le même que celui vu dans la partie sur les en-têtes. Ce qui nous intéresse maintenant est le `Inherits` : il définit que la page hérite de la classe `_Default` qui se trouve dans le namespace `NouveauProjet`.

3.5 Using / Import

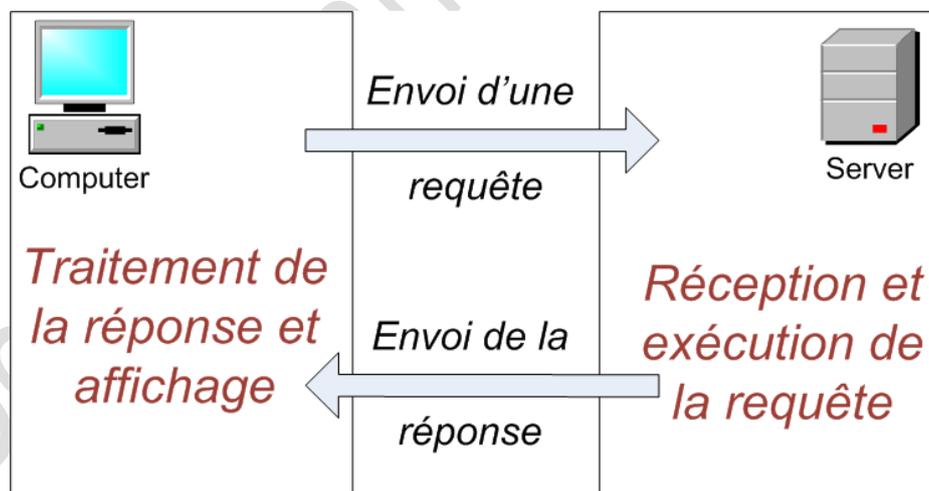
Les `using` / `import` permettent de définir un espace de noms (namespace) qui seront utilisé plus tard dans notre code pour faciliter l'appel d'une classe, méthode etc... ou encore permettre de créer des alias par rapport à un espace de noms. Nous passons rapidement sur cette partie car nous l'expliquerons bien plus en détails dans le chapitre 3.

4 Comprendre le fonctionnement d'une requête

Il est important de comprendre le rôle d'un serveur, d'un navigateur et du protocole HTTP avant de commencer le développement. Nous allons voir une « discussion » typique entre le serveur Web et le navigateur par le biais du protocole HTTP. Pour mieux comprendre nous décrirons plus en détail le rôle du serveur Web ainsi que celui du navigateur. Nous verrons aussi ce qu'il faut savoir sur le protocole HTTP.

4.1 Les intervenants et leur dialogue

Voyons rapidement comment fonctionne une requête. Tout cela sera détaillé dans les parties suivantes (les rôles et le HTTP).



Cycle d'une requête avec un serveur Web

- Le navigateur envoie une requête pour accéder aux ressources du serveur Web (récupère une page, envoi de données ...). Cette requête est envoyée par le biais de la méthode GET et dans le cas de données elles sont passées par méthode POST.
- Le serveur Web reçoit et exécute la requête.

- Il renvoie ensuite la réponse au navigateur.
- Le navigateur reçoit la réponse. Il la traite et l'affiche.
- L'utilisateur lit, donne des informations au navigateur, interagit avec la page (clique sur un lien, remplit un formulaire ...).

Nous venons de voir que les deux intervenants lors du dialogue sont la personne qui va demander une page au serveur (et plus exactement son navigateur), que nous nommerons « client », et le serveur qui va réceptionner et répondre à la requête. Une requête est une demande d'accès aux ressources d'un serveur que ce soit pour demander une page Web ou pour envoyer des informations. Chaque requête et réponse du serveur utilise le protocole HTTP.

Voyons maintenant plus en détail le rôle des deux intervenants et le protocole HTTP.

4.2 Rôle du navigateur (client)

Le rôle d'un navigateur est de récupérer et de traiter les données envoyées par le serveur. Il va exécuter le code qui doit être comme le Javascript et afficher le résultat à l'écran de l'utilisateur. Du fait qu'il existe une grande diversité de navigateurs, la gestion des codes à exécuter et l'affichage ne se feront pas de la même manière sur chacun d'entre eux (par exemple le Javascript ne sera pas supporté pareil, il en va de même pour les feuilles de style CSS).

Il va ensuite recueillir les données entrées par l'utilisateur (lorsqu'il remplit un formulaire entre autres) et envoyer ces données au serveur pour qu'il les traite.

4.3 Rôle du serveur

Le serveur reçoit et récupère les requêtes qui lui sont envoyées. Il va ensuite les traiter : la requête peut être la demande d'affichage d'une page ou l'envoi de données telles que celles d'un formulaire. Il va exécuter la page ASPX correspondante et retourner le résultat. Il est important de savoir que ce que retourne le serveur ne contient pas d'ASP.NET mais du HTML, CSS, Javascript ...

4.4 HyperText Transfer Protocol

HTTP est un protocole de communication textuel utilisé entre les navigateurs et les serveurs. Il utilise le port 80. Il existe aussi un protocole plus sécurisé et chiffré : l'HTTPS qui lui utilise le port 443.

Nous allons voir les méthodes que ce protocole utilise pour communiquer les données mais aussi les différents statuts de la requête et le MIME.

4.4.1 Les méthodes HTTP

Les méthodes HTTP sont les méthodes utilisées pour passer les données entre le serveur et le navigateur.

HTTP méthode	Description ou quand l'utiliser
Head	Il contient les informations comme le navigateur utilisé, la page de provenance, le système d'exploitation utilisé

Get	Permet d'envoyer des données par l'adresse url (visible dans l'adresse en clair)
Post	Permet d'envoyer des données par les http Headers

4.4.2 Les codes de statuts des requêtes

On va voir les codes de statuts de deux manières différentes : la première en groupe, la seconde avec des codes plus précis.

Codes de statuts (groupe)	Description
1XX	Requête reçue, processus en cours.
2XX	Succès : l'action a bien été reçue, comprise et acceptée.
4XX	Erreur client : La requête comporte une erreur de syntaxe ou le serveur n'arrive pas à satisfaire la requête.
5XX	Erreur serveur : Le serveur n'arrive pas à satisfaire une requête qui semble être valide.

Les X représentent un chiffre entre 0 et 9.

Codes de statuts	Description
100	Continue
200	Ok
201	Créé
400	Mauvaise requête
401	Non autorisé
403	Interdit
404	Non trouvé
408	Délais de requête dépassé
501	Non implémenté

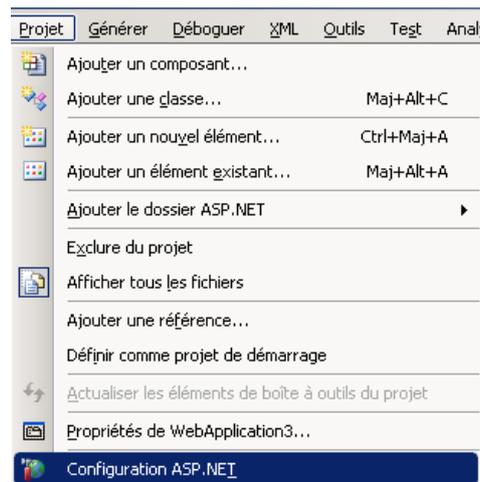
4.4.3 Le MIME

Le MIME permet de définir le type du fichier, on trouve 4 grands types de données: texte, image, son et application.

- Texte: Il peut être affiché par le navigateur : XML, texte simple, page HTML il peut contenir différentes données.
- Image: Les plus communs sont jpg, png, gif, les navigateurs peuvent désactiver les images ou être en mode texte.
- Son: Nécessite une sortie son active.
- Application: Fichier binaire à interpréter par un logiciel tierce ou exécutable.

5 La configuration

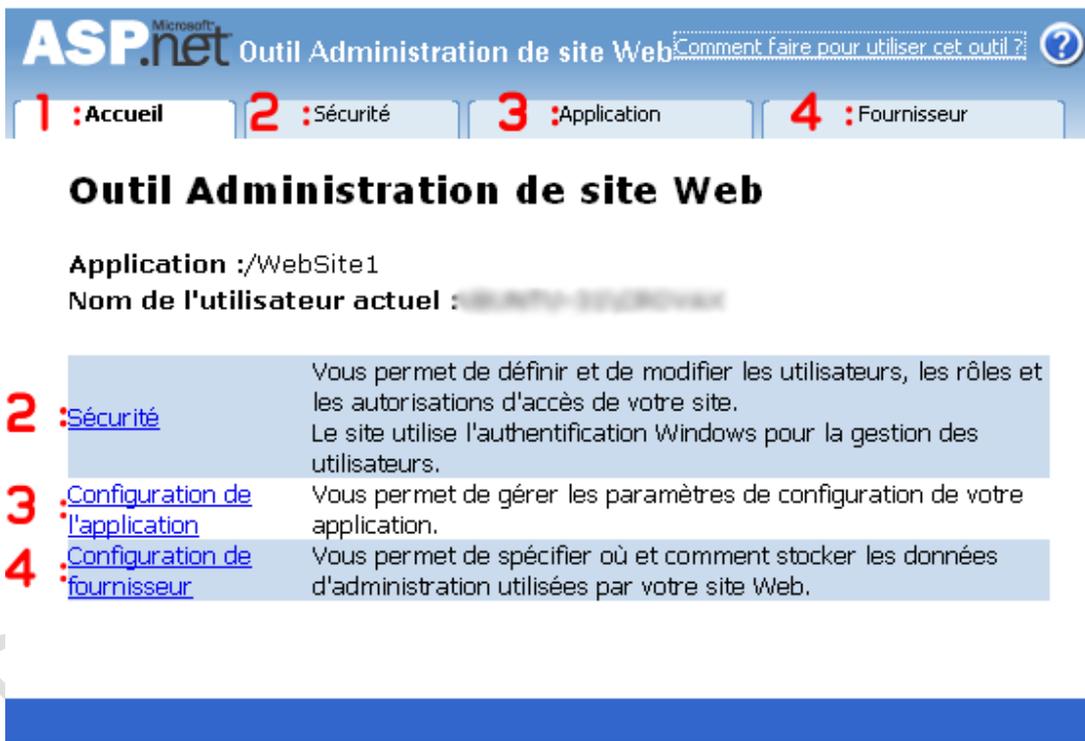
La configuration se fait dans un fichier XML appelé Web .config, les attributs et l'organisation de la structure n'étant pas innée, nous avons à notre disposition des interfaces permettant de modifier la configuration du site.



Configuration d'un projet

Cela ouvre un serveur Web avec un site qui permet la configuration, voici l'interface et les options qu'elle propose.

5.1 Accueil



ASP.net Outil Administration de site Web [Comment faire pour utiliser cet outil ?](#) 

1 : Accueil **2** : Sécurité **3** : Application **4** : Fournisseur

Outil Administration de site Web

Application : /WebSite1
Nom de l'utilisateur actuel : 

2 : Sécurité	Vous permet de définir et de modifier les utilisateurs, les rôles et les autorisations d'accès de votre site. Le site utilise l'authentification Windows pour la gestion des utilisateurs.
3 : Configuration de l'application	Vous permet de gérer les paramètres de configuration de votre application.
4 : Configuration de fournisseur	Vous permet de spécifier où et comment stocker les données d'administration utilisées par votre site Web.

Page d'accueil du site d'administration pour la configuration

Les liens des onglets sont en redondance avec la page d'accueil, cette page mentionne le nom de l'application, le nom de l'utilisateur actuel et permet donc un accès direct à chaque page.

- 1 : Vous permet de revenir à l'accueil.
- 2 : Vous mène à la page de gestion des utilisateurs, rôles et règles de sécurité.
- 3 : Vous permet de configurer le serveur mail, les paramètres d'application et le debug.
- 4 : Vous propose de configurer des accès à des bases de données.

5.2 Sécurité

La page de gestion de la sécurité est une interface simple qui permet de gérer l'authentification, les rôles et les règles d'accès.



Vous pouvez utiliser l'outil Administration de site Web pour gérer tous les paramètres de sécurité de votre application. Vous pouvez définir des utilisateurs et des mots de passe (authentification), créer des rôles (groupes d'utilisateurs) et créer des autorisations (règles de contrôle d'accès à différentes parties de votre application).

Par défaut, les informations utilisateur sont stockées dans une base de données Microsoft SQL Server Express dans le dossier Data de votre site Web. Pour stocker les informations utilisateur dans une base de données différente, utilisez l'onglet Fournisseur pour sélectionner un autre fournisseur.

[Utilisez l'Assistant Installation de sécurité pour configurer la sécurité étape par étape](#) : 4

Cliquez sur les liens dans le tableau pour gérer les paramètres de votre application.

Utilisateurs	Rôles	Règles d'accès
Le type d'authentification actuel est Windows . La gestion des utilisateurs est donc désactivée pour cet outil. Sélectionnez le type d'authentification : 1	Les rôles ne sont pas activés Activer les rôles : 2	Créer des règles d'accès Gérer les règles d'accès

Configuration de la sécurité

5.2.1 Types d'authentification

Il y a deux types d'authentification, celle par le réseau Windows (utile pour les solutions d'entreprises) qui permet de ne pas avoir à se loguer sur chaque site de l'entreprise et de ne pas avoir à implémenter la gestion de l'authentification nous-même. En revanche il faut nécessairement être sur un réseau local ou VPN avec des postes clients sous Windows. L'autre type est l'authentification internet où les utilisateurs sont dans la base de données.

Pour changer le type d'authentification vous pouvez cliquer sur le lien 1 et sélectionner le type d'authentification puis valider avec le bouton « Terminer » en bas à droite.

5.2.2 Rôles

Les rôles servent à gérer les droits par groupe. Si vous souhaitez gérer des droits il est recommandé d'utiliser des rôles pour faciliter la maintenance et les améliorations, cela permet de faire des règles sur des groupes plutôt que sur chaque personne.

Commencez par activer les rôles (lien 2) :

Rôles

Rôles existants : 0

[Désactiver des rôles](#)

[Créer ou gérer des rôles](#)

Gestion des rôles

Puis allez sur l'interface suivante (Créer ou gérer des rôles) :

Vous pouvez ajouter des rôles ou des groupes qui vous permettent d'autoriser ou d'empêcher des groupes d'utilisateurs d'accéder à des dossiers spécifiques de votre site Web. Par exemple, vous pouvez créer des rôles, tels que "gestion", "ventes" ou "membres", chacun avec un accès à des dossiers spécifiques différents.

Créer un nouveau rôle

Nouveau nom de rôle :

Nom du rôle	Ajouter /supprimer des utilisateurs	
bidule	Gérer	Supprimer

Création d'un nouveau rôle

Dans la TextBox vous pouvez mettre le nom du rôle à créer puis cliquer sur le lien, ensuite vous pouvez gérer les rôles pour y ajouter des utilisateurs qui répondront aux règles de sécurité du groupe.

5.2.3 Règles de sécurités

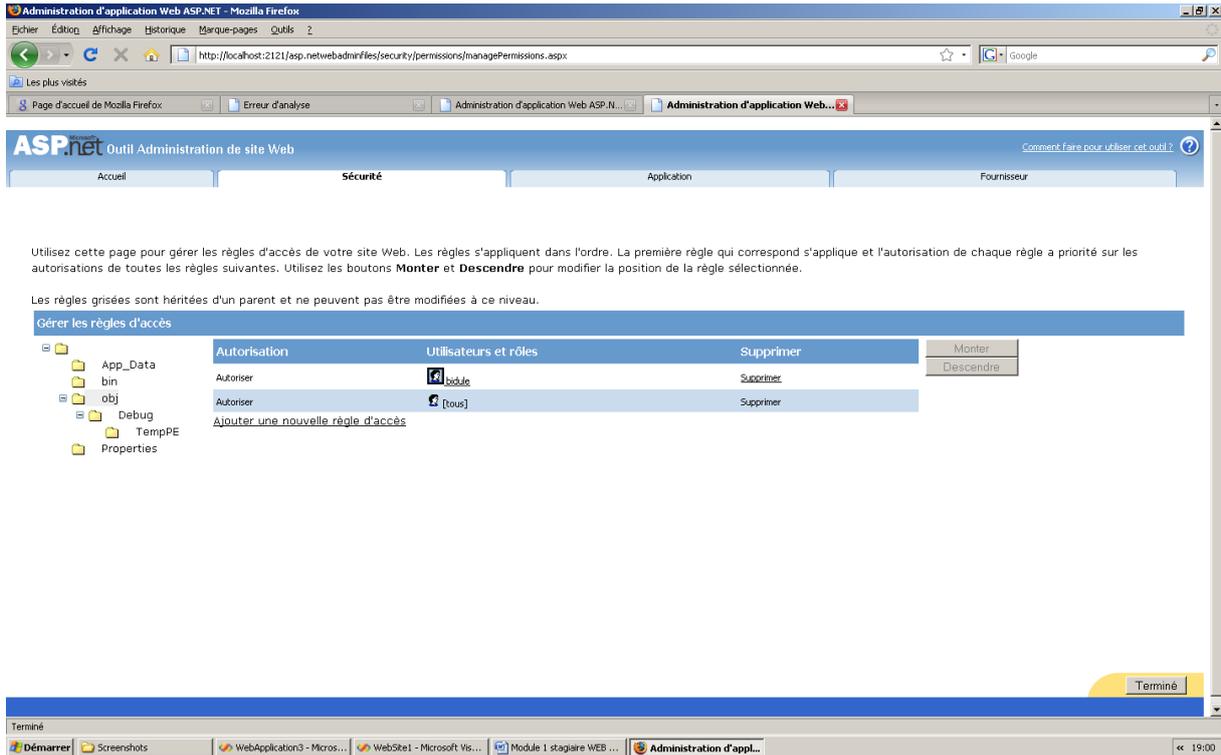
Les règles de sécurité sont une manière de restreindre des parties de l'arborescence à des utilisateurs, des rôles (ou groupes), à l'ensemble des personnes ou à l'ensemble des authentifiés.

Ajouter une nouvelle règle d'accès

<p>Sélectionnez un répertoire pour cette règle :</p> <ul style="list-style-type: none"> [-] App_Data [-] bin [-] obj <ul style="list-style-type: none"> [-] Debug <ul style="list-style-type: none"> [-] TempPE [-] Properties 	<p>La règle s'applique à :</p> <p><input checked="" type="radio"/> Rôle <input type="text" value="bidule"/></p> <p><input type="radio"/> utilisateur <input type="text"/></p> <p><input type="radio"/> Tous les utilisateurs</p> <p><input type="radio"/> Utilisateurs anonymes</p>	<p>Autorisation :</p> <p><input type="radio"/> Autoriser</p> <p><input checked="" type="radio"/> Refuser</p>
---	--	---

Gestion des règles de sécurité

Pour utiliser cette interface vous devez sélectionner les droits, l'utilisateur, le dossier puis un bouton apparaîtra pour pouvoir valider.



Fenêtre de gestion de la sécurité

Une fois les règles créées vous pouvez les organiser (grâce aux boutons « Monter » et « Descendre »), la règle la plus haute dans cette liste est prioritaire en cas de conflit.

5.3 Application



Utilisez cette page pour configurer votre application avec des valeurs que vous ne voulez pas coder en dur dans vos pages, activer votre application pour envoyer des messages, configurer le débogage, définir une page d'erreurs par défaut et arrêter ou redémarrer votre application.

<p>Paramètres de l'application</p> <p>Paramètres d'application existants : 0</p> <p>Créer des paramètres d'application : 1</p> <p>Gérer les paramètres d'application</p>	<p>Paramètres SMTP</p> <p>2 : Configurer les paramètres de messagerie SMTP</p>	<p>État de l'application</p> <p>L'application est : en ligne</p> <p>3 : Mettre l'application hors connexion</p>
		<p>Débogage et traçage</p> <p>4 : Configurer le débogage et le traçage</p> <p>5 : Définir la page d'erreurs par défaut</p>

Gestion des paramètres de l'application

Les paramètres de l'application peuvent être paramétrés par le lien 1, ils entrent dans un dictionnaire global qui va chercher ses données et ses clefs dans le web.config. Le dictionnaire est une variable associant à chaque clef une donnée. Nous reparlerons plus en détails des dictionnaires plus tard. Pour accéder à un paramètre d'application nommé "param" dans le code, il faut l'appeler comme suit : `ConfigurationManager.AppSettings("param")`

Le lien 2 permet de configurer un serveur mail (SMTP) accessible depuis un client mail quelconque.

Le lien 3 permet de diffuser ou non le site sur le réseau.

Le lien 4 sera détaillé dans la partie sur le debug et le traçage.

Le lien 5 définit une page de redirection par défaut en cas d'erreur au lieu de laisser aux visiteurs la possibilité de lire des extraits de sources, des messages d'erreurs explicites ou pas forcément rassurant pour les novices.

5.4 Fournisseur

Ce menu vous permet d'ajouter et de configurer des bases de données locales ou distantes.



Utilisez cette page pour configurer le stockage des données de gestion de site Web, telles que l'appartenance. Vous pouvez utiliser un fournisseur unique pour toutes les données de gestion de votre site ou spécifier un fournisseur différent pour chaque fonctionnalité.

Votre application est actuellement configurée pour utiliser le fournisseur :
AspNetSqlProvider

[Sélectionnez un fournisseur unique pour toutes les données de gestion de site](#)

[Sélectionnez un fournisseur différent pour chaque fonctionnalité \(avancée\)](#)

Gestion des fournisseurs

Nous approfondirons l'ajout des bases de données lorsque nous aborderons ADO.NET.