

Introduction rapide à ASP.NET avec Visual Studio 2010

serge.tahe at istia.univ-angers.fr, novembre 2011

1 Introduction

Nous nous proposons ici d'introduire, à l'aide de quelques exemples, les concepts importants d'ASP.NET. Cette introduction ne permet pas de comprendre les subtilités des échanges client / serveur d'une application web. Pour cela, on pourra lire d'anciens documents qui décrivent les échanges Http entre un navigateur et un serveur web :

- **Programmation ASP.NET** [<http://tahe.developpez.com/dotnet/aspnet/vol1>] et [<http://tahe.developpez.com/dotnet/aspnet/vol2>]

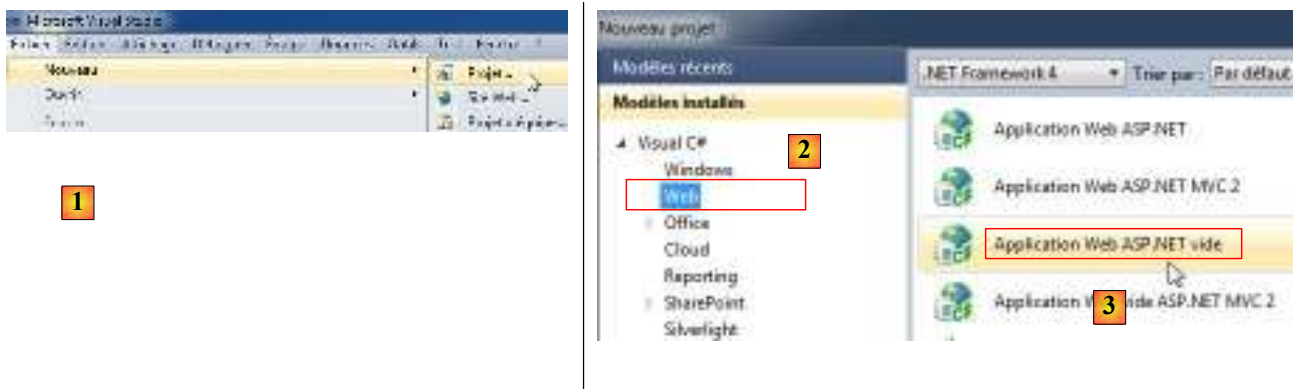
Cette introduction est faite pour ceux qui voudraient aller vite en acceptant, dans un premier temps, de laisser dans l'ombre des points qui peuvent être importants. Ce document peut être ultérieurement approfondi par une étude de cas :

- **Construction d'une application à trois couches avec ASP.NET, C#, Spring.net et Nhibernate** [<http://tahe.developpez.com/dotnet/pam-aspnet/>]

Les exemples ont été construits avec Visual Studio 2010 Professionnel. Des versions Express de Visual Studio 2010 sont disponibles à l'Url [<http://msdn.microsoft.com/fr-fr/express/aa975050>] (novembre 2011) et peuvent être téléchargées librement. On téléchargera ainsi Visual Web Developer Express 2010. Les copies d'écran de ce document sont celles de Visual Studio 2010 Professionnel . Elles pourront parfois différer des écrans de Visual Studio 2010 Express mais le lecteur devrait s'y retrouver néanmoins aisément.

2 Un projet exemple

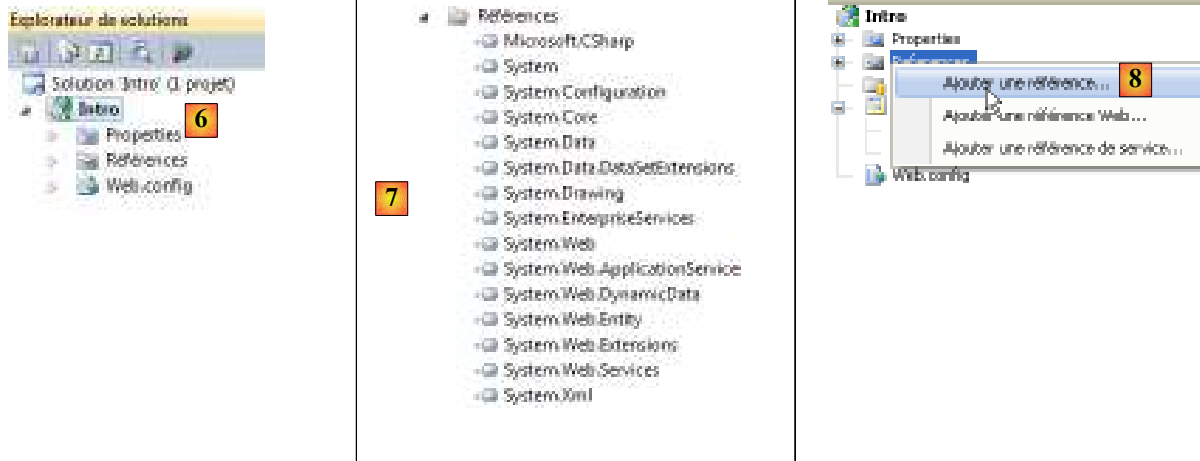
2.1 Création du projet



- en [1], on crée un nouveau projet avec Visual Studio
- en [2], on choisit un projet web en Visual C#
- en [3], on indique qu'on veut créer une application web ASP.NET vide



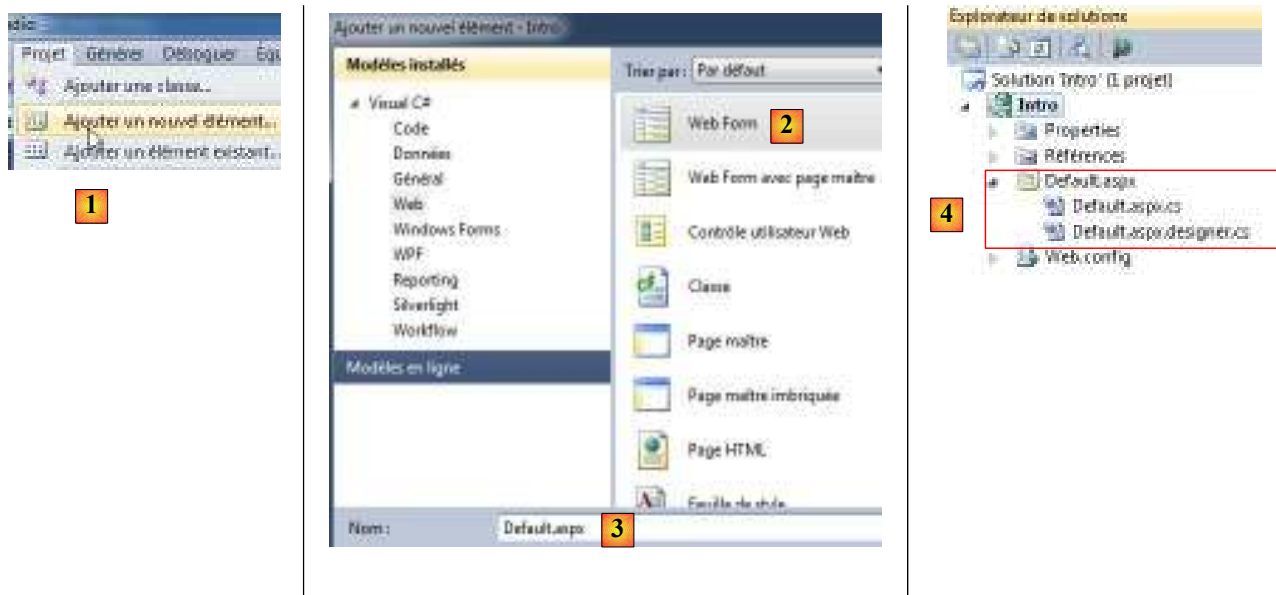
- en [4], on donne un nom à l'application. Un dossier sera créé pour le projet avec ce nom.
- en [5], on indique le dossier parent du dossier [4] du projet



- en [6], le projet créé
- [Web.config] est le fichier de configuration du projet ASP.NET.
- [References] est la liste des Dll utilisées par le projet web. Ces Dll sont des bibliothèques de classes que le projet est amené à utiliser. En [7] la liste des Dll mises par défaut dans les références du projet. La plupart sont inutiles. Si le projet doit utiliser une Dll non listée en [7], celle-ci peut être ajoutée par [8].

2.2 La page [Default.aspx]

Créons notre première page web :

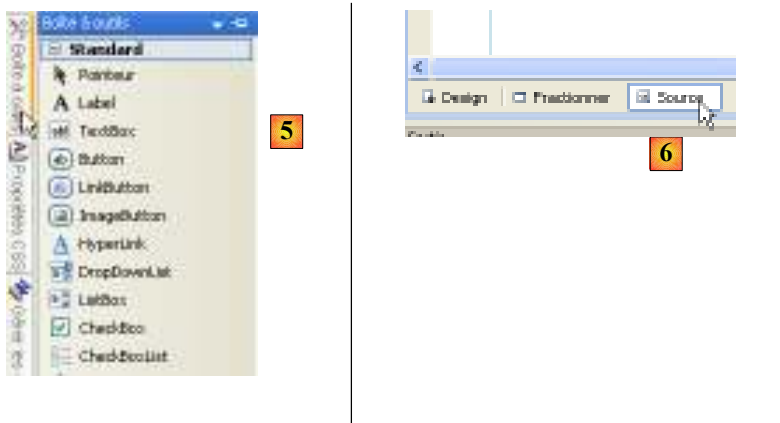


- en [1], nous ajoutons un formulaire web [2]
- en [3], nous le nommons [Default.aspx]. Cette page est particulière. C'est elle qui est servie lorsqu'un utilisateur fait une requête à l'application sans préciser de page. Elle contient des balises HTML et des balises ASP.NET
- en [4], les nouveaux fichiers du projet
- [Default.aspx.cs] contient le code de gestion des événements provoqués par l'utilisateur sur la page [Default.aspx] affichée dans son navigateur
- [Default.aspx.designer.cs] contient la liste des composants ASP.NET de la page [Default.aspx]. Chaque composant ASP.NET déposé sur la page [Default.aspx] donne naissance à la déclaration de ce composant dans [Default.aspx.designer.cs].

Si on exécute le projet par [Ctrl-F5], la page [Default.aspx] est affichée dans un navigateur :



- en [1], l'Url du projet web. Visual Studio a un serveur web intégré qui est lancé lorsqu'on demande l'exécution d'un projet web. Il écoute sur un port aléatoire, ici 57365. Le port d'écoute est habituellement le port 80. En [1], aucune page n'est demandée. Dans ce cas, c'est la page [Default.aspx] qui est affichée, d'où son nom de page par défaut.
- en [2], la page [Default.aspx] est vide.
- dans Visual Studio, la page [Default.aspx] [3] peut être construite visuellement (onglet [Design]) ou à l'aide de balises (onglet [Source])
- en [4], la page [Default.aspx] en mode [Design]. On la construit en y déposant des composants que l'on trouve dans la boîte à outils [5]. Si cette boîte n'est pas visible, on l'obtient par le menu [Affichage / Boîte à outils].



Le mode [Source] [6] donne accès au code source de la page :

```

1. <%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
   Inherits="Intro._Default" %>
2.
3. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4. <html xmlns="http://www.w3.org/1999/xhtml">
5. <head runat="server">
6.   <title></title>
7. </head>
8. <body>
9.   <form id="form1" runat="server">
10.    <div>
11.    </div>
12.  </form>
13. </body>
14. </html>

```

- la ligne 1 est une directive ASP.NET qui liste certaines propriétés de la page
 - la directive *Page* s'applique à une page web. Il y a d'autres directives telles que *Application*, *WebService*, ... qui s'applique à d'autres objets ASP.NET
 - l'attribut *CodeBehind* indique le fichier qui gère les événements de la page
 - l'attribut *Language* indique le langage .NET utilisé par le fichier *CodeBehind*
 - l'attribut *Inherits* indique le nom de la classe définie à l'intérieur du fichier *CodeBehind*
 - l'attribut *AutoEventWireUp="true"* indique que la liaison entre un événement dans [Default.aspx] et son gestionnaire dans [Default.aspx.cs] se fait par le nom de l'événement. Ainsi l'événement *Load* sur la page [Default.aspx] sera traité par la méthode *Page_Load* de la classe *Intro._Default* définie par l'attribut *Inherits*.
- les lignes 4-14 décrivent la page [Default.aspx] à l'aide de balises :
 - Html classiques telles que la balise *<body>* ou *<div>*

- ASP.NET. Ce sont les balises qui ont l'attribut *runat="server"*. Les balises ASP.NET sont traitées par le serveur web avant envoi de la page au client. Elles sont transformées en balises Html. Le navigateur client reçoit donc une page Html standard dans laquelle il n'existe plus de balises ASP.NET.

La page [Default.aspx] peut être modifiée directement à partir de son code source. C'est parfois plus simple que de passer par le mode [Design]. Nous modifions le code source de la façon suivante :

```

1. <%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
   Inherits="Intro._Default" %>
2.
3. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4. <html xmlns="http://www.w3.org/1999/xhtml">
5. <head runat="server">
6.   <title>Introduction ASP.NET</title>
7. </head>
8. <body>
9.   <h3>Introduction à ASP.NET</h3>
10.  <form id="form1" runat="server">
11.    <div>
12.    </div>
13.  </form>
14. </body>
15. </html>

```

En ligne 6, nous donnons un titre à la page grâce à la balise Html <title>. En ligne 9, nous introduisons un texte dans le corps (<body>) de la page. Si nous exécutons le projet (Ctrl-F5), nous obtenons le résultat suivant dans le navigateur :



2.3 Les fichiers [Default.aspx.designer.cs] et [Default.aspx.cs]

Le fichier [Default.aspx.designer.cs] déclare les composants de la page [Default.aspx] :

```

1. //-----
2. // <auto-generated>
3. //   Ce code a été généré par un outil.
4. //   Version du runtime :2.0.50727.3603
5. //
6. //   Les modifications apportées à ce fichier peuvent provoquer un comportement incorrect et seront perdues si
7. //   le code est régénéré.
8. // </auto-generated>
9. //-----
10.
11. namespace Intro {
12.
13.
14.   public partial class _Default {
15.
16.       /// <summary>
17.       /// Contrôle form1.
18.       /// </summary>
19.       /// <remarks>
20.       /// Champ généré automatiquement.
21.       /// Pour modifier, déplacez la déclaration de champ du fichier de concepteur dans le fichier code-
   behind.
22.       /// </remarks>
23.       protected global::System.Web.UI.HtmlControls.HtmlForm form1;
24.   }
25. }

```

On trouve dans ce fichier la liste des composants ASP.NET de la page [Default.aspx] ayant un identifiant. Ils correspondent aux balises de [Default.aspx] ayant l'attribut *runat="server"* et l'attribut *id*. Ainsi le composant de la ligne 23 ci-dessus correspond à la balise

```
<form id="form1" runat="server">
```

de [Default.aspx].

Le développeur interagit peu avec le fichier [Default.aspx.designer.cs]. Néanmoins ce fichier est utile pour connaître la classe d'un composant particulier. Ainsi on voit ci-dessous que le composant *form1* est de type *HtmlForm*. Le développeur peut alors explorer cette classe pour en connaître les propriétés et méthodes. Les composants de la page [Default.aspx] sont utilisés par la classe du fichier [Default.aspx.cs] :

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Web;
5. using System.Web.UI;
6. using System.Web.UI.WebControls;
7.
8. namespace Intro
9. {
10.     public partial class _Default : System.Web.UI.Page
11.     {
12.         protected void Page_Load(object sender, EventArgs e)
13.         {
14.
15.         }
16.     }
17. }
```

On notera que la classe définie dans les fichiers [Default.aspx.cs] et [Default.aspx.designer.cs] est la même (ligne 10) : *Intro._Default*. C'est le mot clé *partial* qui rend possible d'étendre la déclaration d'une classe sur plusieurs fichiers, ici deux.

Ligne 10, ci-dessus, on voit que la classe [_Default] étend la classe [Page] et hérite de ses événements. L'un d'entre-eux est l'événement *Load* qui se produit lorsque la page est chargée par le serveur web. Ligne 12, la méthode *Page_Load* qui gère l'événement *Load* de la page. C'est généralement ici qu'on initialise la page avant son affichage dans le navigateur du client. Ici, la méthode *Page_Load* ne fait rien.

La classe associée à une page web, ici la classe *Intro._Default*, est créée au début de la requête du client et détruite lorsque la réponse au client a été envoyée. Elle ne peut donc servir à mémoriser des informations entre deux requêtes. Pour cela il faut utiliser la notion de *session utilisateur*.

2.4 Les événements d'une page web ASP.NET

Nous construisons la page [Default.aspx] suivante :

```
1. <%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
2.     Inherits="Intro._Default" %>
3.
4. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
5.     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
6. <html xmlns="http://www.w3.org/1999/xhtml">
7. <head runat="server">
8.     <title>Introduction ASP.NET</title>
9. </head>
10. <body>
11.     <h3>Introduction à ASP.NET</h3>
12.     <form id="form1" runat="server">
13.     <div>
14.         <table>
15.             <tr>
16.                 <td>
17.                     Nom</td>
18.                 <td>
19.                     <asp:TextBox ID="TextBoxNom" runat="server"></asp:TextBox>
20.                 </td>
21.             </tr>
22.             <tr>
23.                 <td>
24.                     &nbsp;   </td>
25.                 <td>
```

Ceci est un exemple, cliquez sur le lien de téléchargement pour obtenir le cours complet.

