

# Réalisation d'un client « riche »

**Rédacteurs** : Alexandre Baillif, Philippe Lacomme et Raksmei Phan

**Date** : septembre 2010

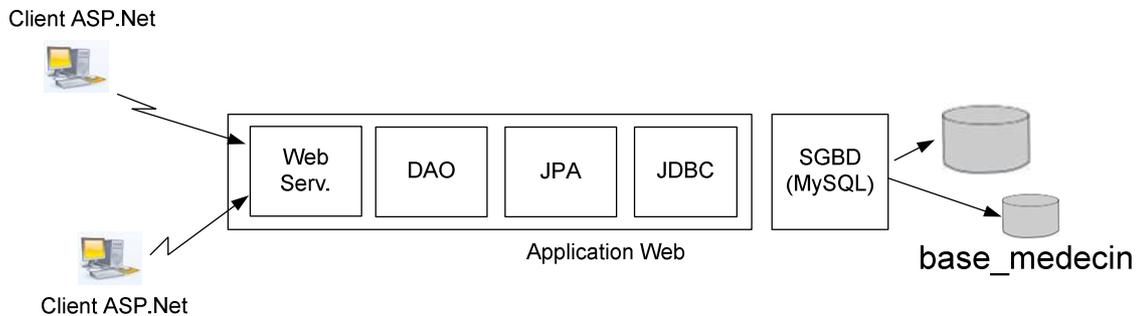
**Avertissement** : ce document est une reprise d'une partie d'un document écrit par Serge Tahé. En particulier la base de données utilisée.



Réalisation d'un client ASP.NET

Réalisation d'un client Flex

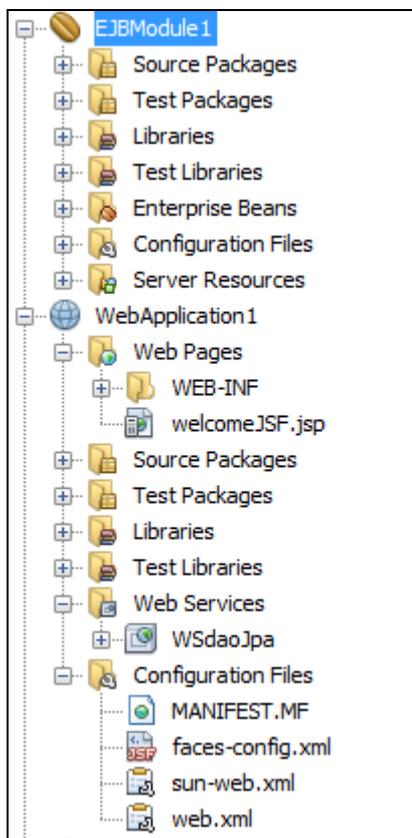
# Réalisation d'un client ASP.NET



## 1) Lien avec le tutorial 1.

Dans le tutorial numéro 1, nous avons déjà créé un web service...  
Petits rappels donc.

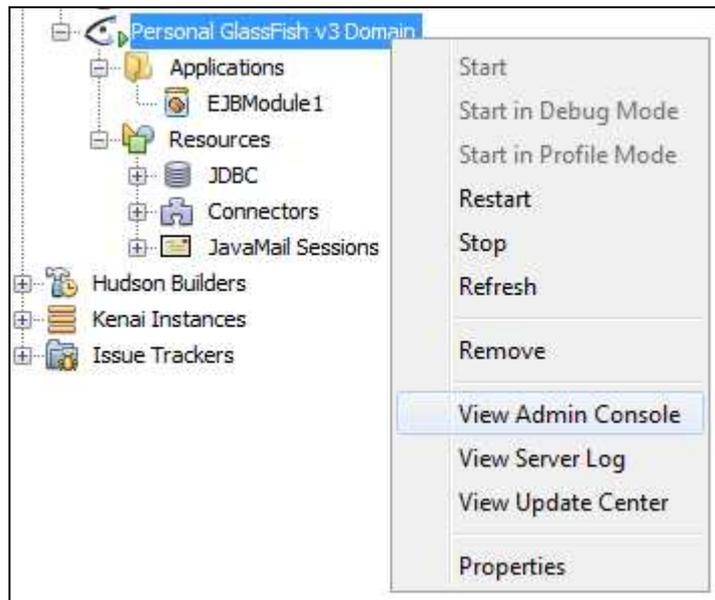
Ouvrez le code **EJBModule1** et **WebApplication1**.



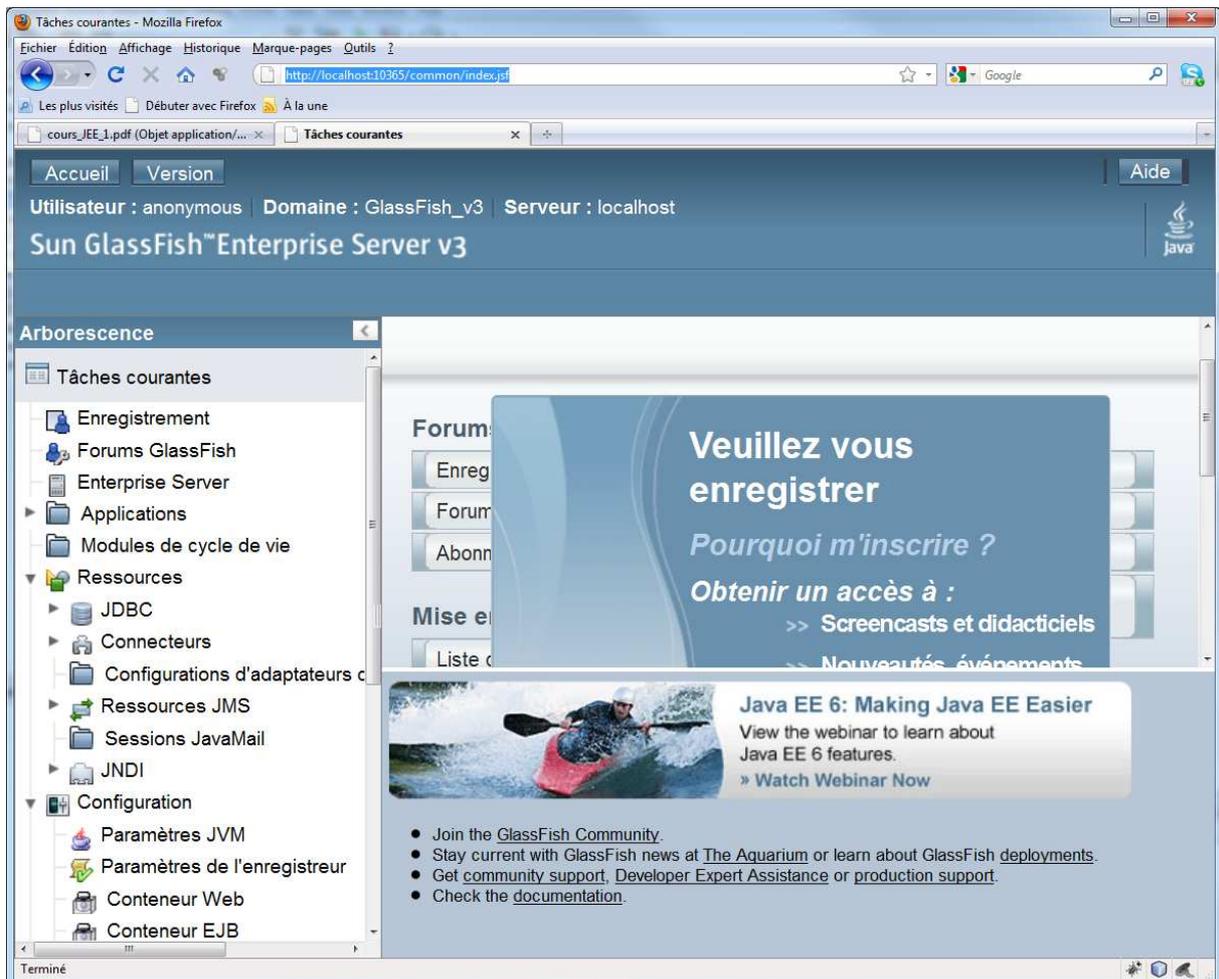
Faire un clic droit sur **EJBModule1** et choisir **Clean and Build** pour vérifier que tout fonctionne correctement. Prenez le temps de déployer l'EJB et de vérifier que celui-ci apparait dans la liste des applications



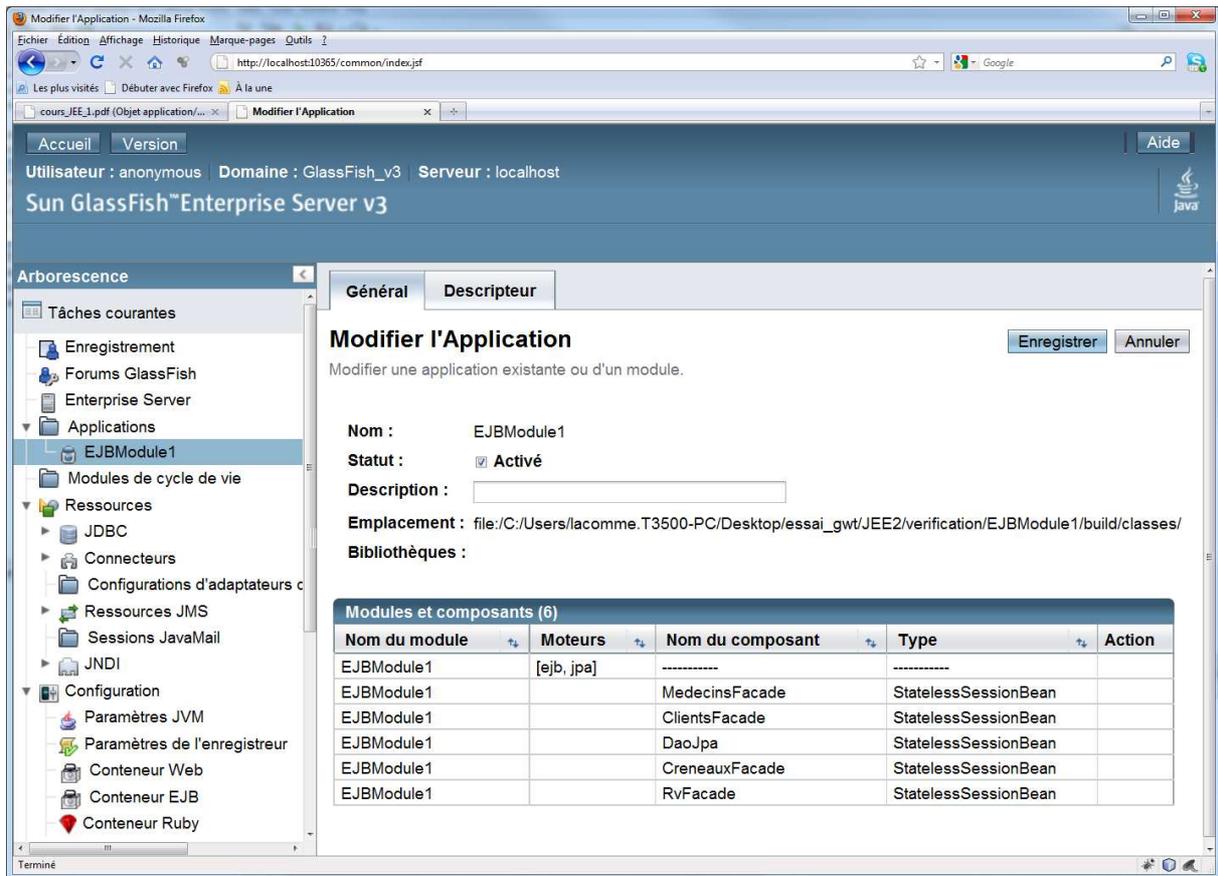
Faites un clic droit sur **GlassFish** et choisir **View Admin Console**.



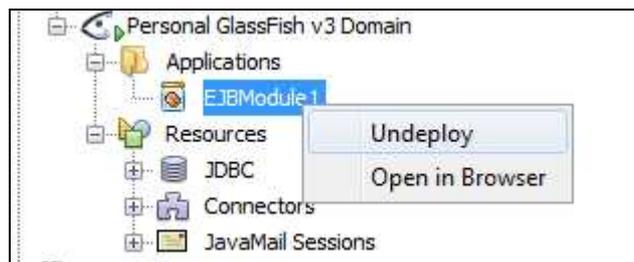
Ce qui, sur ma machine, lance le navigateur avec l'adresse suivante :  
<http://localhost:10365/common/index.jsf>



Dans la partie Applications, vous devriez retrouver EJBModule1.

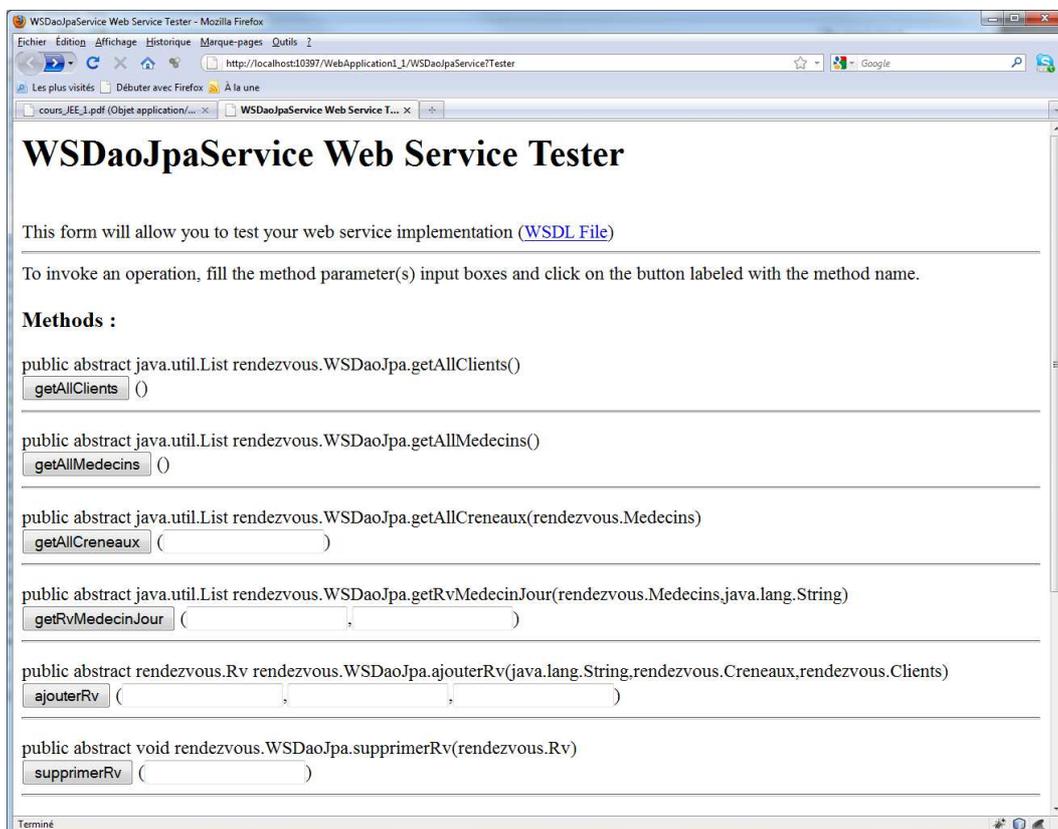
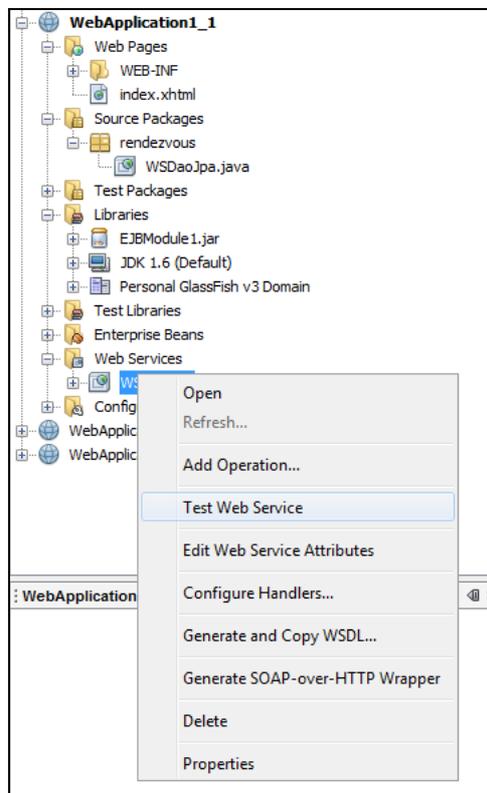


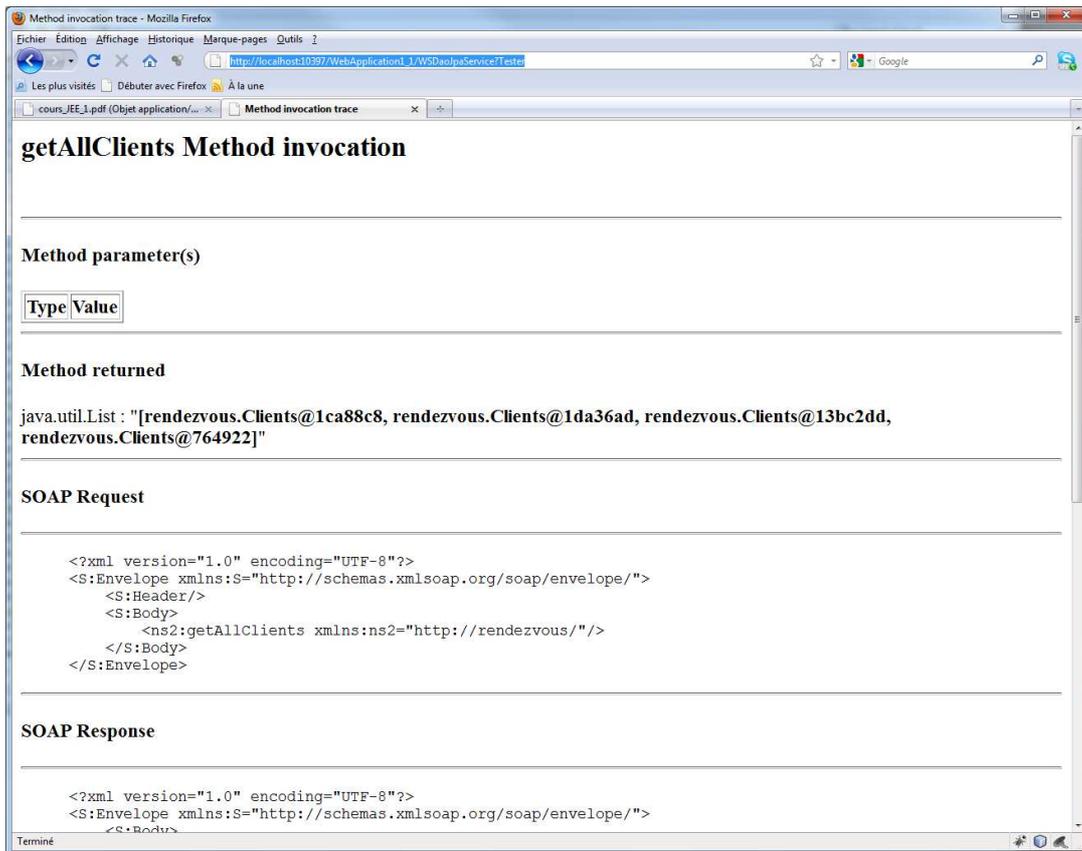
Supprimer l'EJBModule1 des applications déployées.



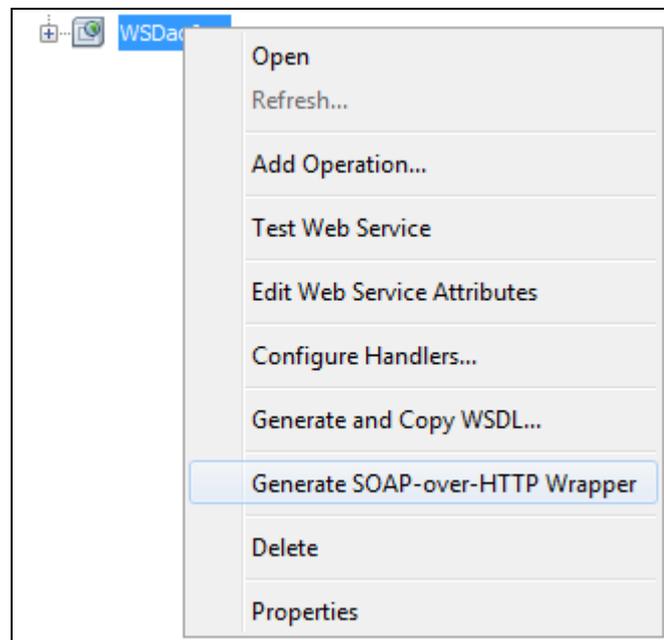
Reprenez le tutorial 1 et en suivant les indications du support, vous pouvez recréer en quelques minutes une WebApplication1\_1 que nous allons utiliser par la suite.

Prenez le temps de faire un test du web service :



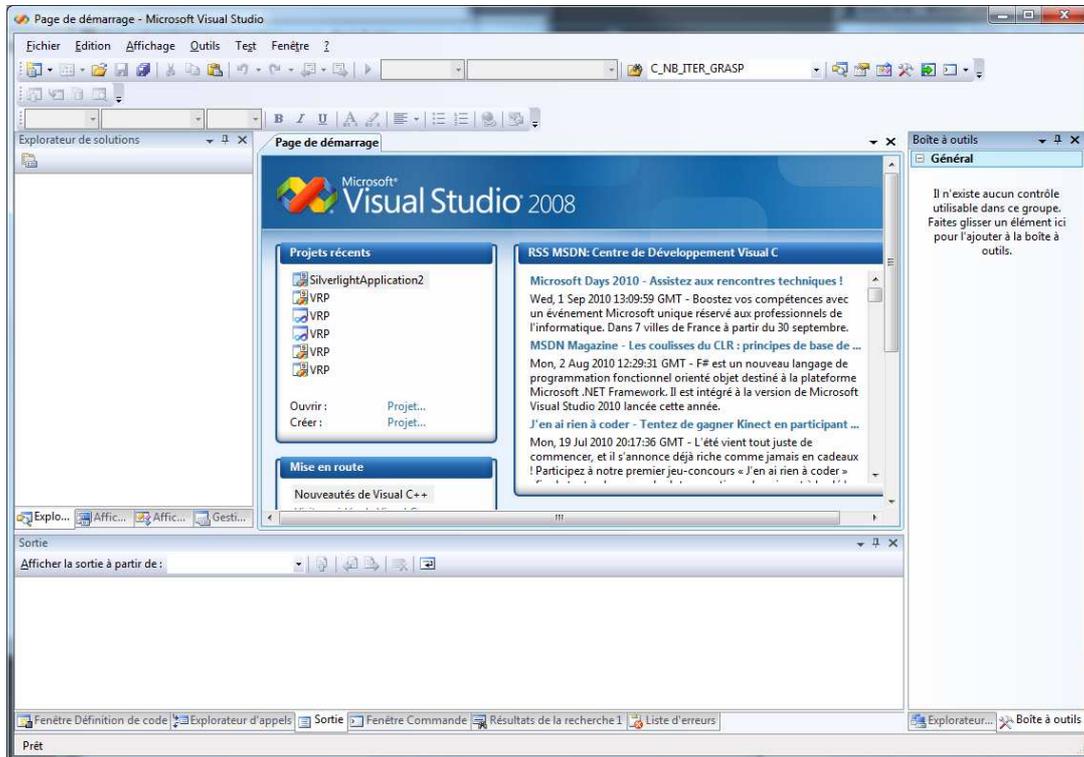


Pour terminer faire un clic droit sur **WSDaoJpa** et **Generate SOAP**.

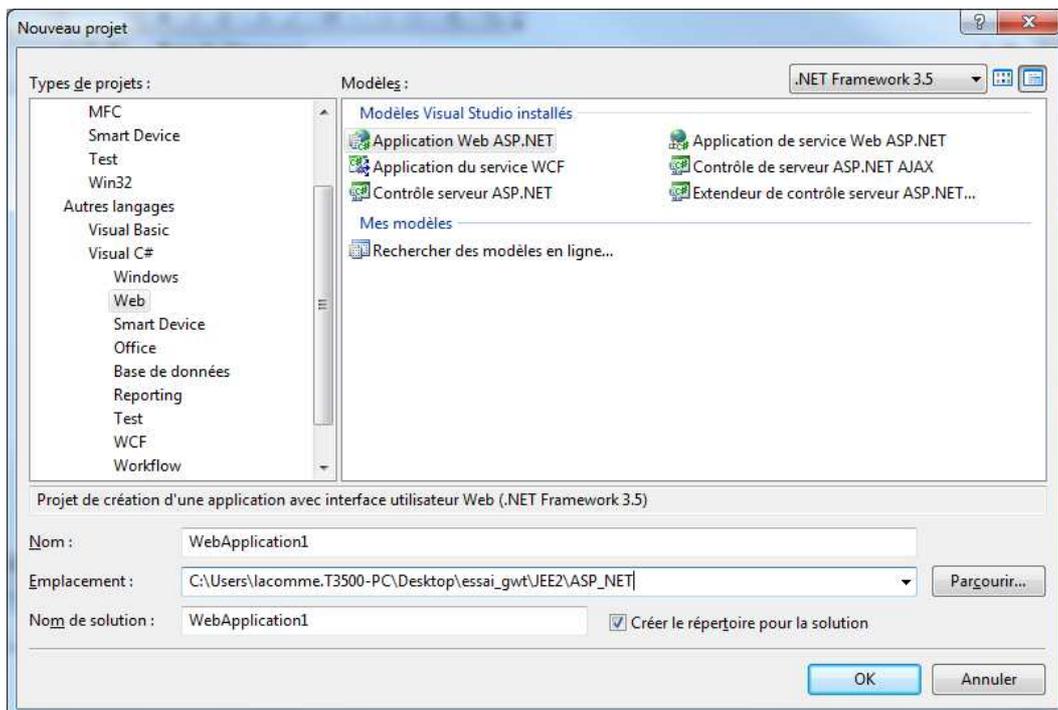


## 2) Création d'une application C# - Exemple numéro 1

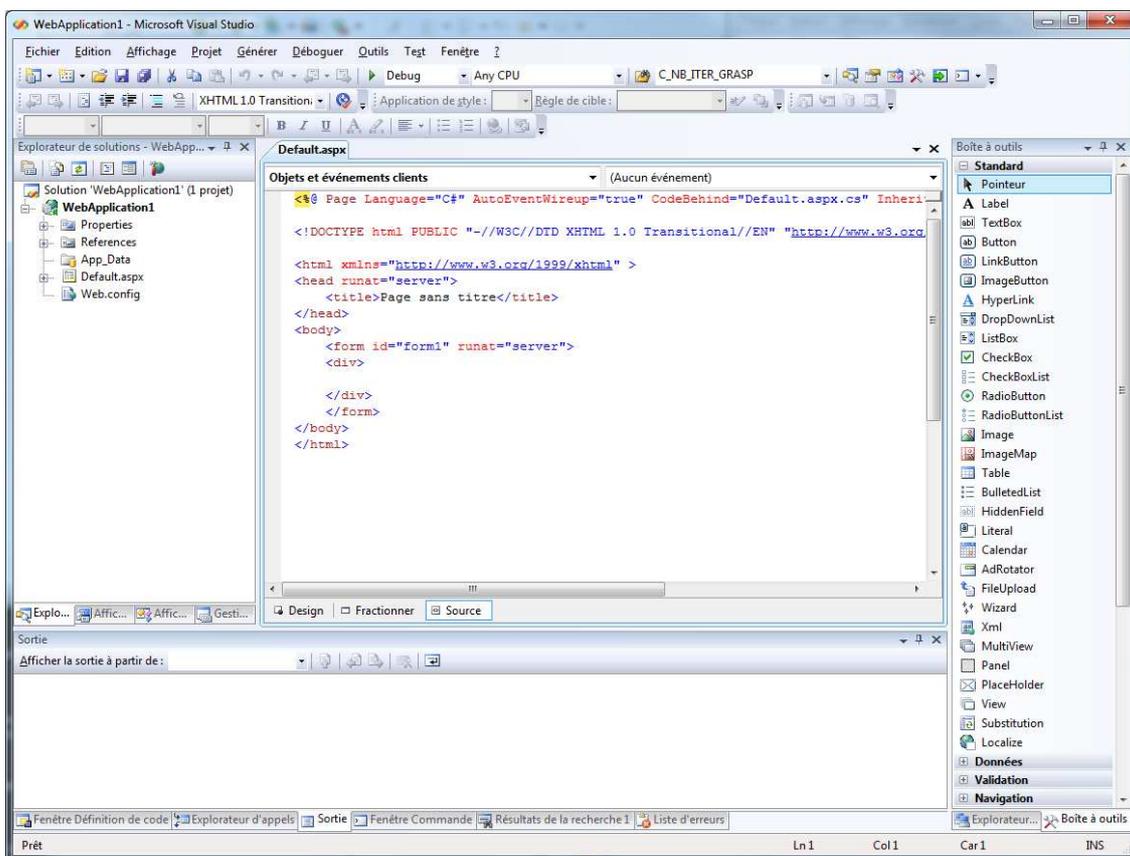
### Etape 2.1. Démarrer Microsoft Visual Studio 2008



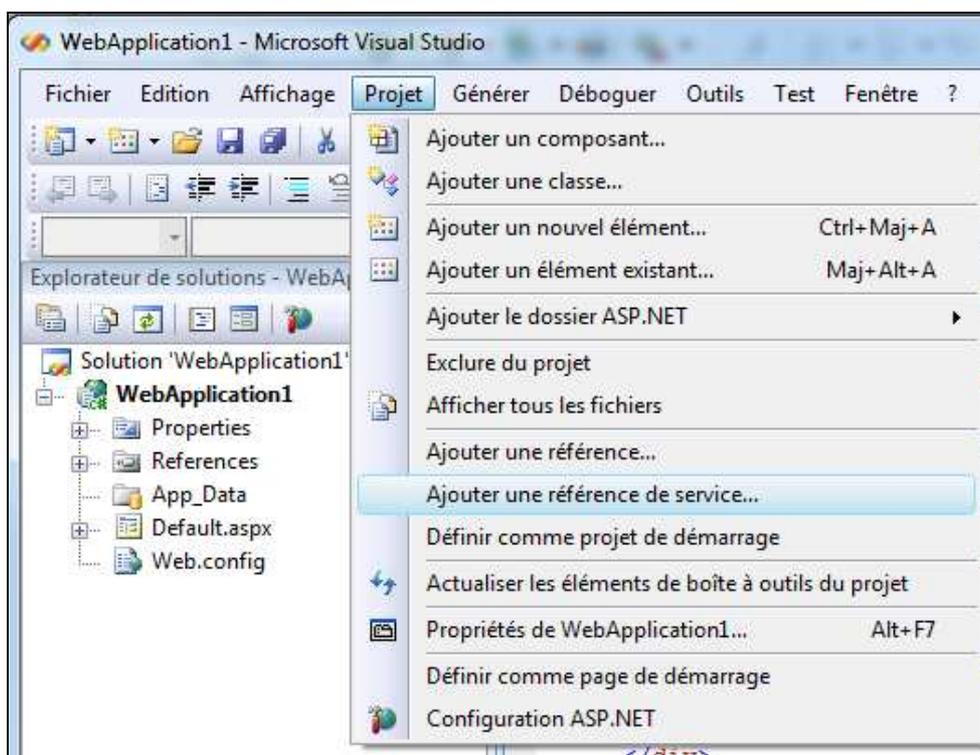
### Choisir Application Web ASP.NET



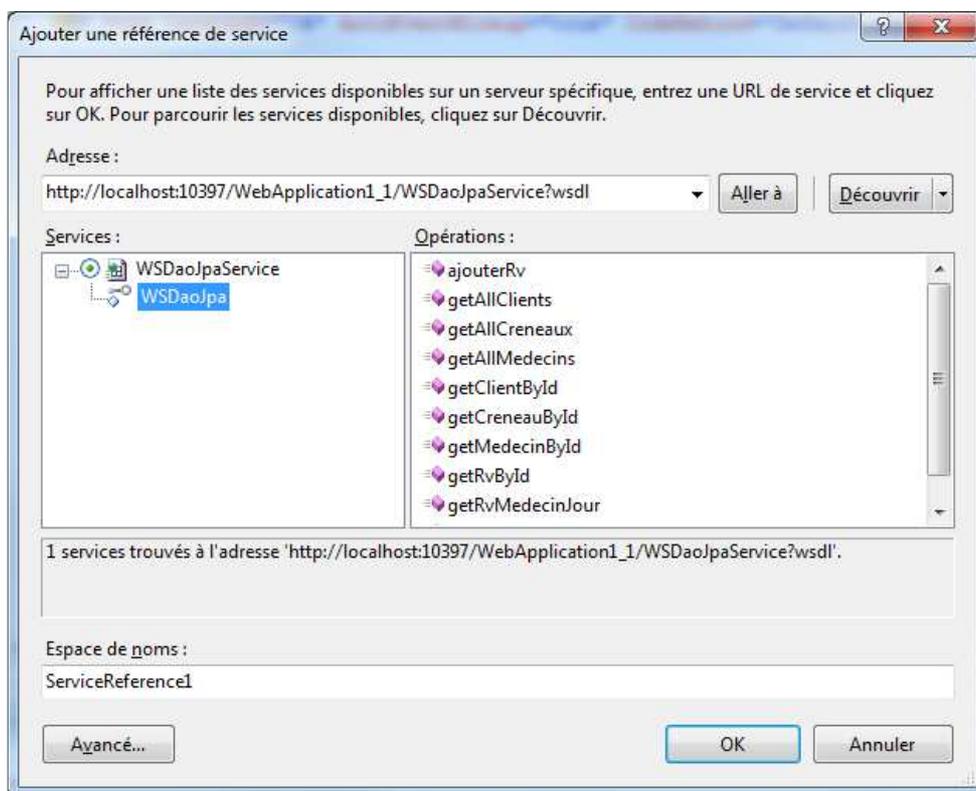
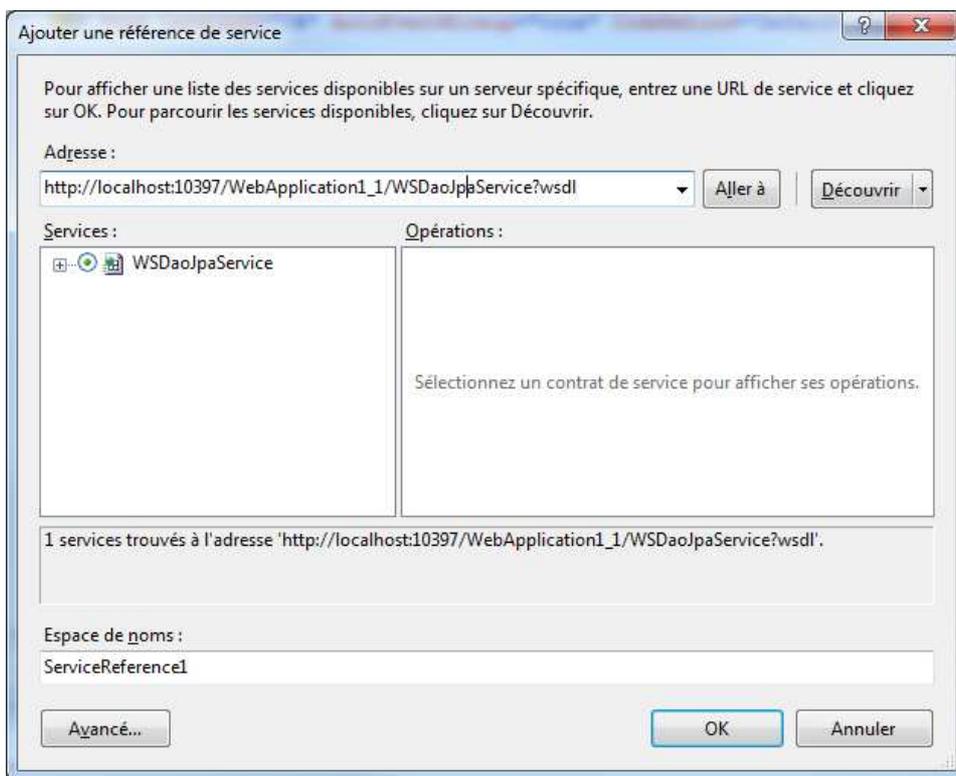
L'environnement de développement se présente comme ceci :



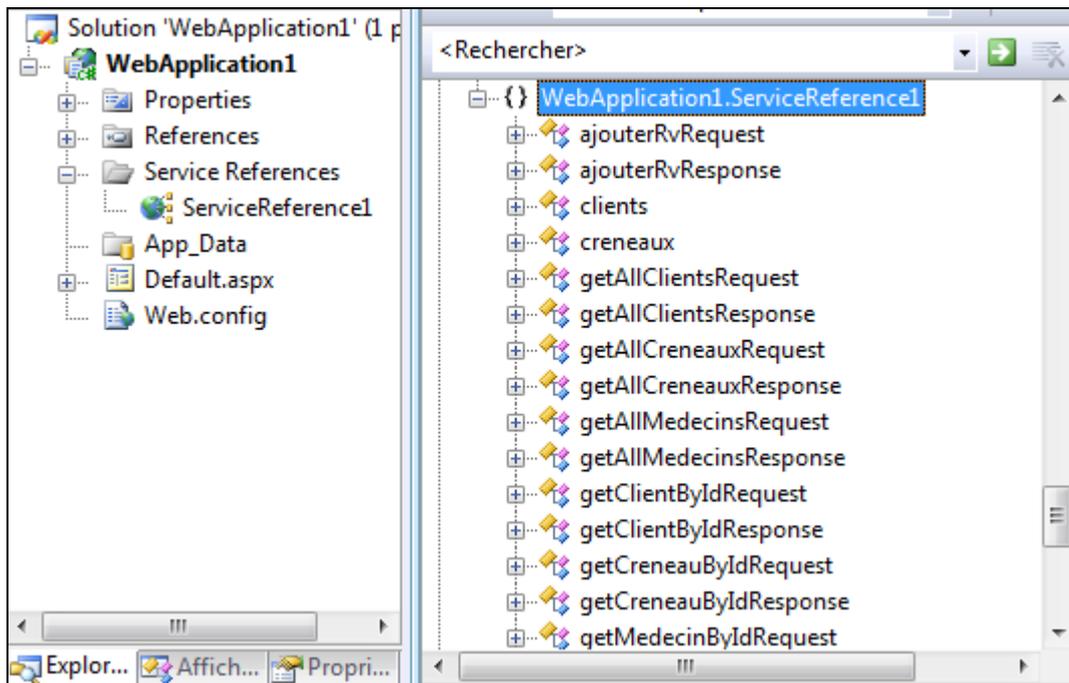
Faire **Projet** / Ajouter une référence de service



Choisir comme adresse : [http://localhost:10397/WebApplication1\\_1/WSDaoJpaService?wsdl](http://localhost:10397/WebApplication1_1/WSDaoJpaService?wsdl)



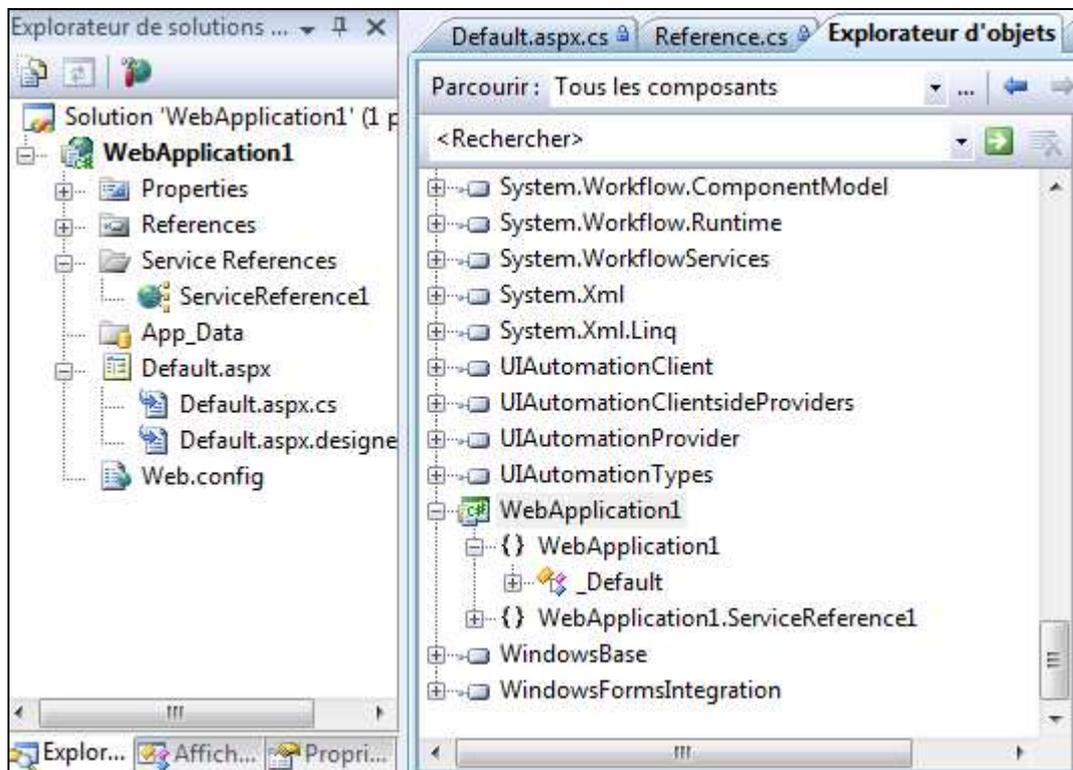
Faire un double clic sur ServiceReference1 qui est l'espace de nom choisi précédemment.



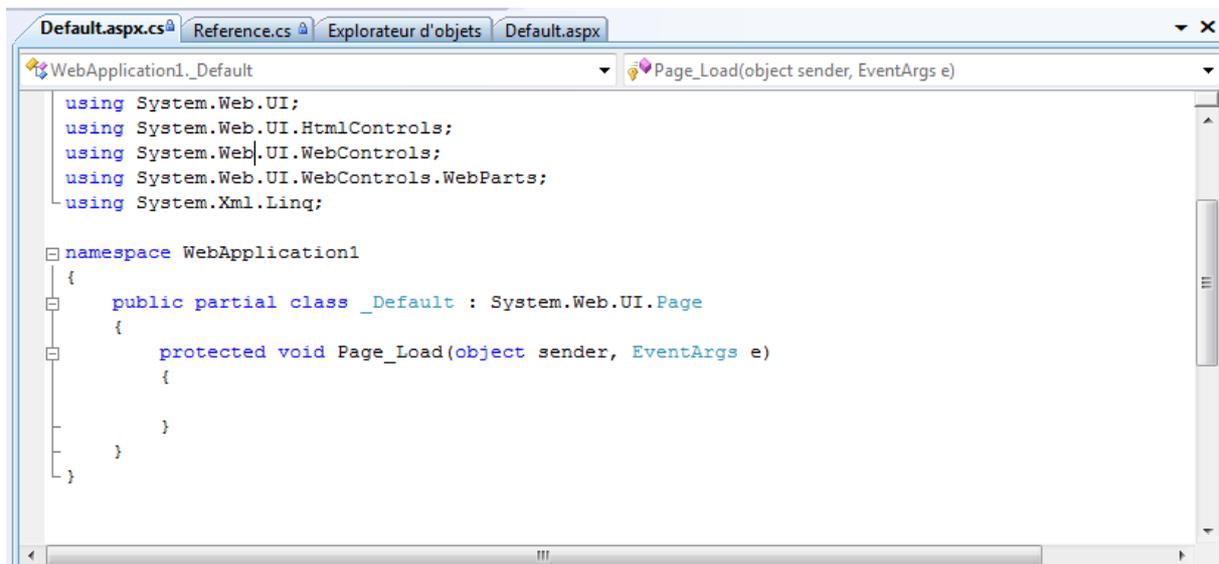
En choisissant une méthode et en cliquant dessus, on obtient le code correspondant par exemple **getAllClientRequest()** pour notre exemple.



Remontez d'un niveau pour faire apparaître la WebApplication1.



Ouvrez ensuite le fichier Default.aspx.cs



Le code doit ressembler à ce qui suit :

```
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

namespace WebApplication1
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
    }
}
```

On peut alors facilement accéder au web service. Le code suivant (qui n'a aucun intérêt ! ) fonctionne sans problème par exemple :

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

using WebApplication1.ServiceReference1;
using Client = WebApplication1.ServiceReference1.clients;

namespace WebApplication1
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            WSDaoJpaClient dao = new WSDaoJpaClient();
            foreach (Client client in dao.getAllClients()) {
                Console.WriteLine(String.Format("{0} {1} {2}",
                    client.titre, client.prenom, client.nom));
            }
        }
    }
}
```

## Etape 2.2. Principes d'une page dynamique

Ouvrir le fichier **Default.aspx** et modifier le code comme suit :

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="WebApplication1._Default" %>

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

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>Page sans titre</title>
</head>
<body>
  <form id="form1" runat="server">
  <div>

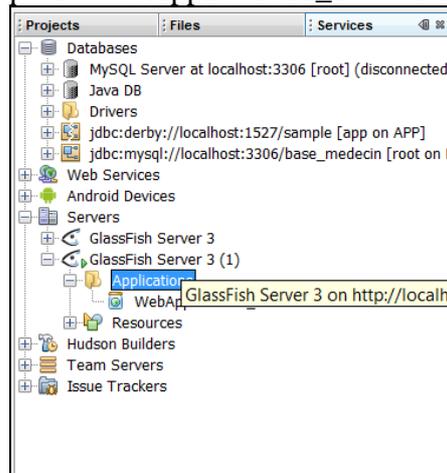
    <table>
      <tr>
        <td> Name: </td>
        <td> <asp:textbox id="MyTextBox" runat="server" /> </td>
      </tr>
      <tr>
        <td></td>
        <td><asp:button id="MyButton" text="Click Here"
          onclick="SubmitBtn_Click" runat="server" /></td>
      </tr>
      <tr>
        <td></td>
        <td><span id="MySpan" runat="server" /></td>
      </tr>
    </table>

  </div>
</form>
</body>
</html>
```

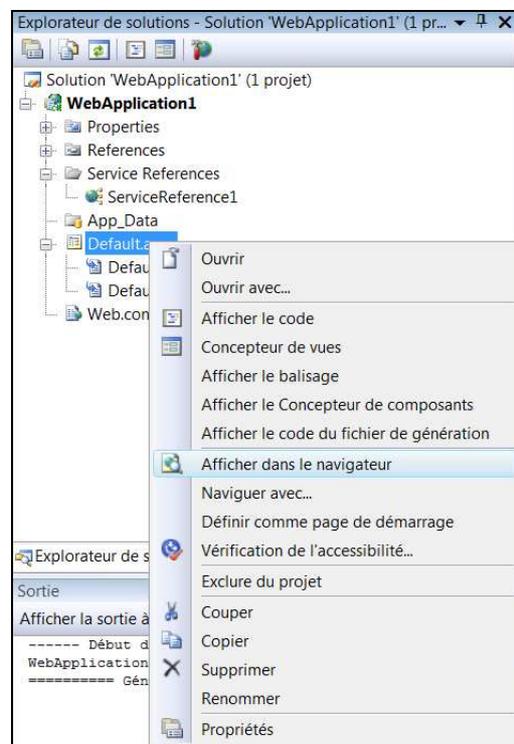
Ouvrir le fichier Default.aspx.cs et ajouter la procedure qui suit :

```
protected void SubmitBtn_Click(object sender, EventArgs e)
{
    MySpan.InnerHtml = "Hello, " + MyTextBox.Text + ".";
}
```

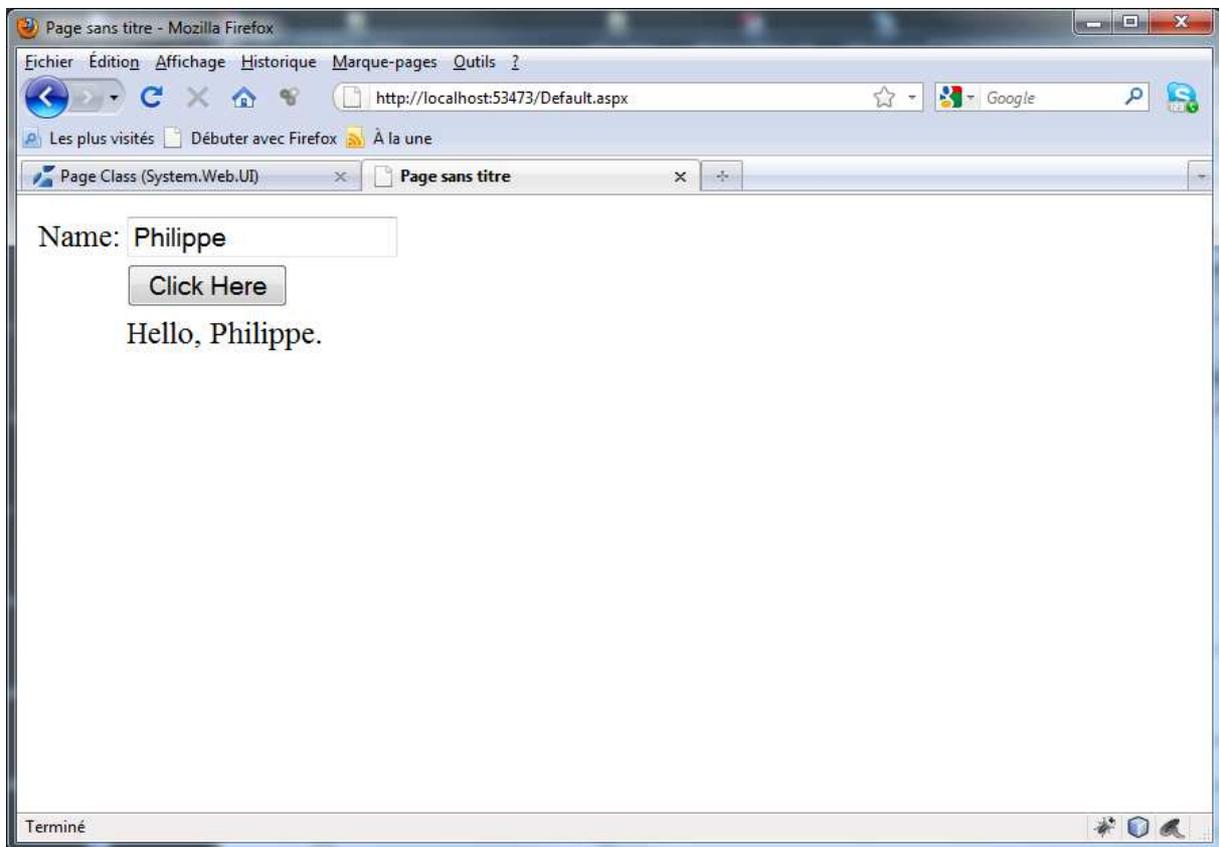
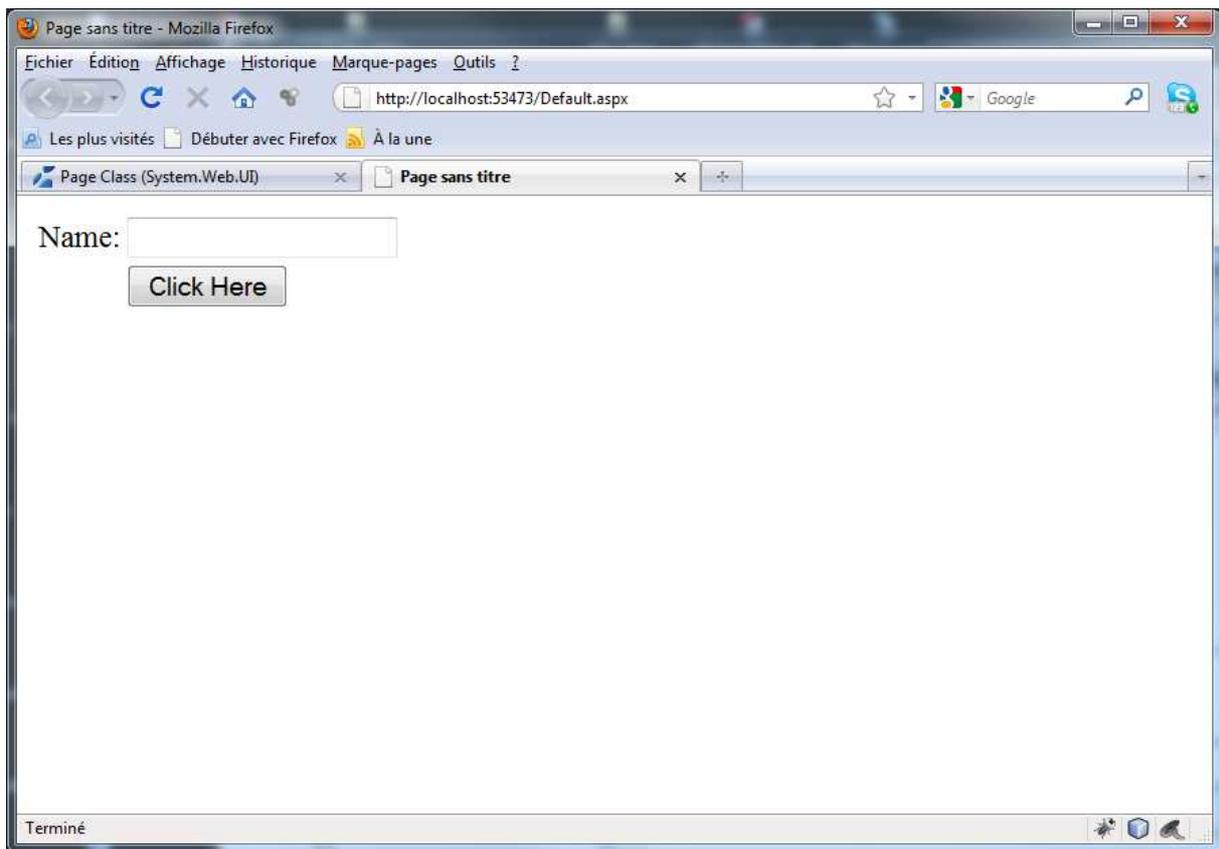
Avant de tester, vérifiez bien que le `WebApplication1_1` est bien déployé dans NetBeans :



Pour tester faites :



Le résultat donne :



Etape 2.3. Création d'une page dynamique (enfin !)

Modifier le fichier Default.aspx comme suit :

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="WebApplication1._Default" %>

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

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Page sans titre</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            <table>
                <tr>
                    <td> Name: </td>
                    <td> <asp:textbox id="MyTextBox" runat="server"/> </td>
                </tr>
                <tr>
                    <td></td>
                    <td><asp:button id="MyButton" text="Click Here"
                        onclick="SubmitBtn_Click" runat="server"/></td>
                </tr>
                <tr>
                    <td></td>
                    <td><span id="MySpan" runat="server" /></td>
                </tr>
            </table>

            <table>
                <td><asp:button id="Button1" text="Liste des clients..."
                    onclick="SubmitBtn_Click2" runat="server"/></td>
            </table>

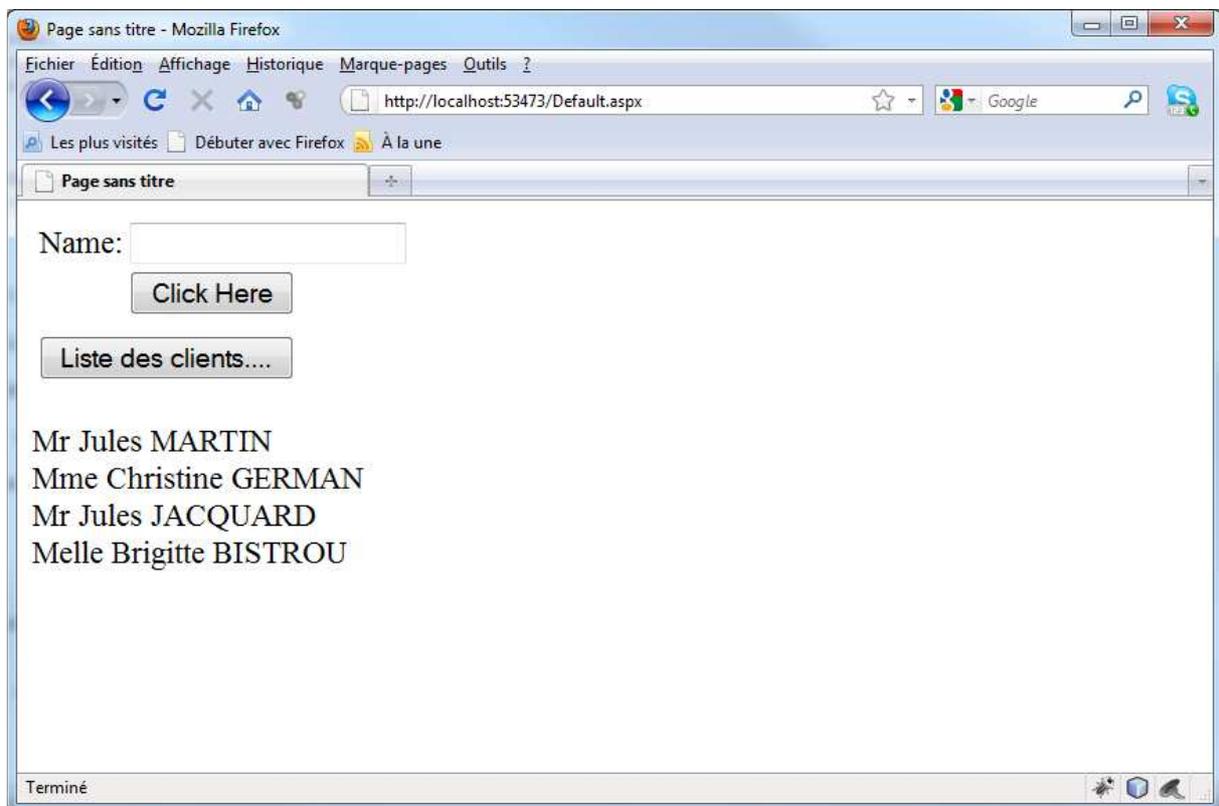
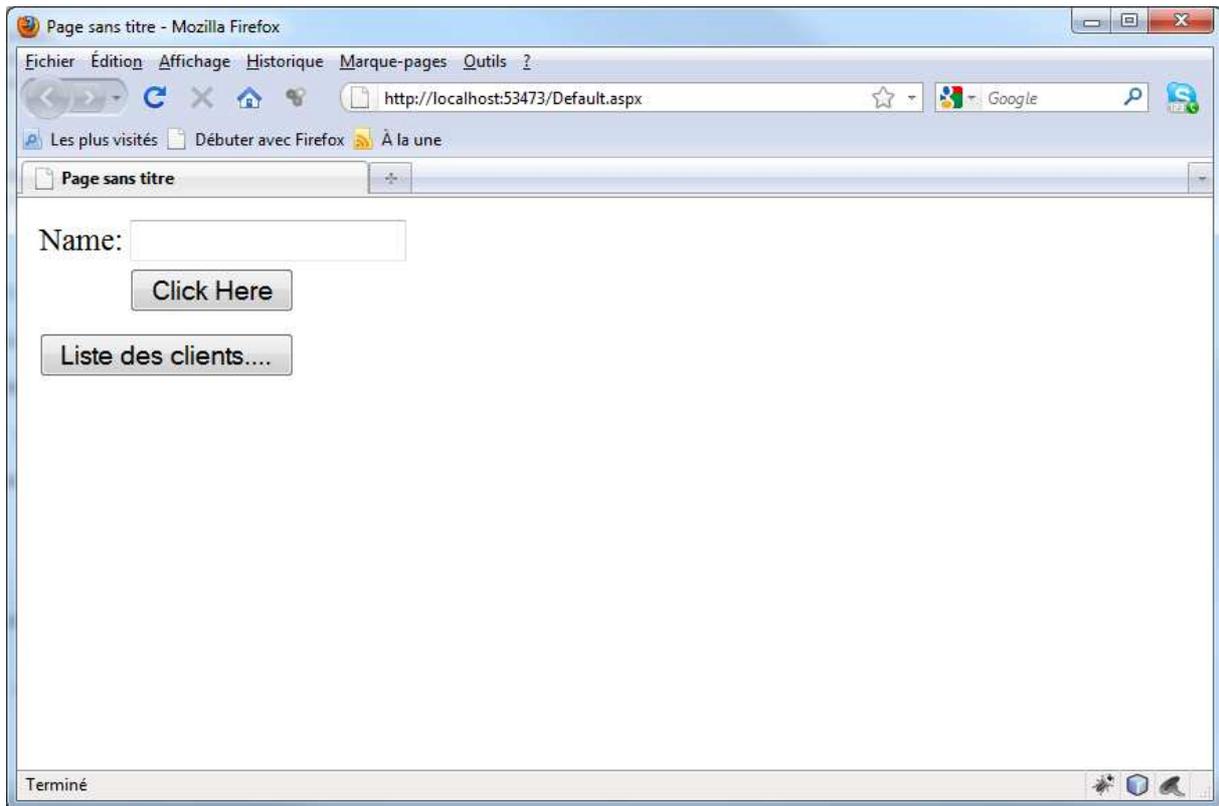
            <td><span id="Span1" runat="server" /></td>

        </div>
    </form>
</body>
</html>
```

Ajouter la procédure submitBtn\_Click2 :

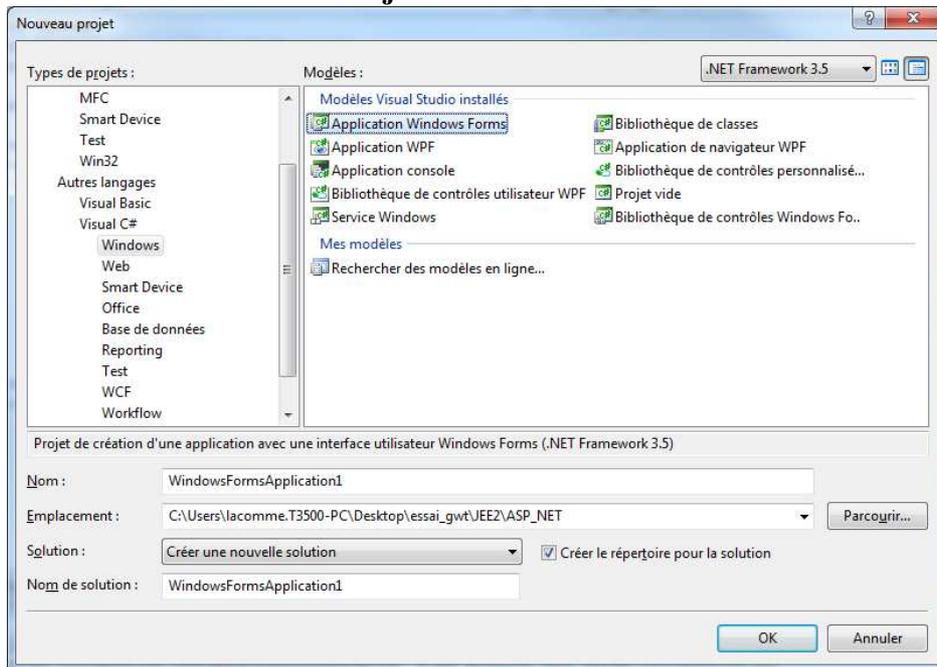
```
protected void SubmitBtn_Click2(object sender, EventArgs e)
{
    WSDaoJpaClient dao = new WSDaoJpaClient();
    foreach (Client client in dao.getAllClients())
    {
        Span1.InnerHtml = Span1.InnerHtml + "<br> "+client.titre + " "
            + client.prenom+ " "+ client.nom+"<br>";
    }
}
```

Ce qui donne à l'exécution:



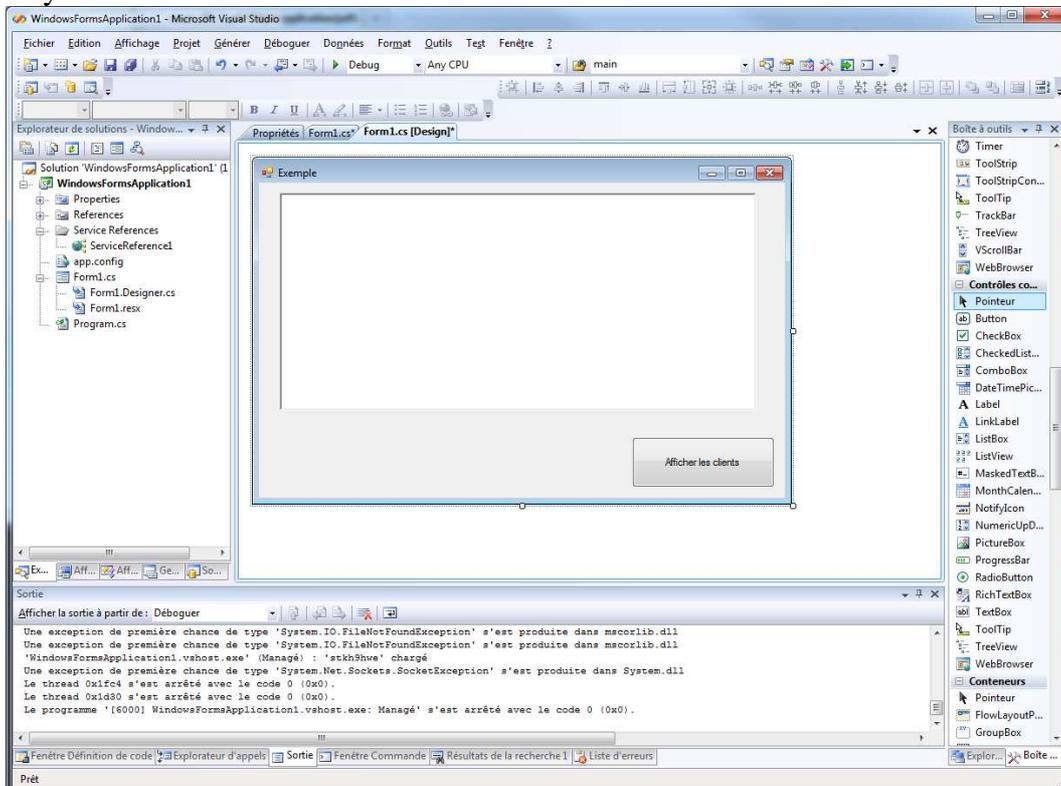
### 3) Création d'une application C# - Exemple numéro 2

#### Etape 2.1. Faire **Fichier / Nouveau Projet**



Créer une application ressemblant à ce qui suit avec :

- System.Windows.Forms.Button
- System.Windows.Forms.RichTextBox



Dans la suite, nous allons ajouter du code sur l'événement clic bouton.

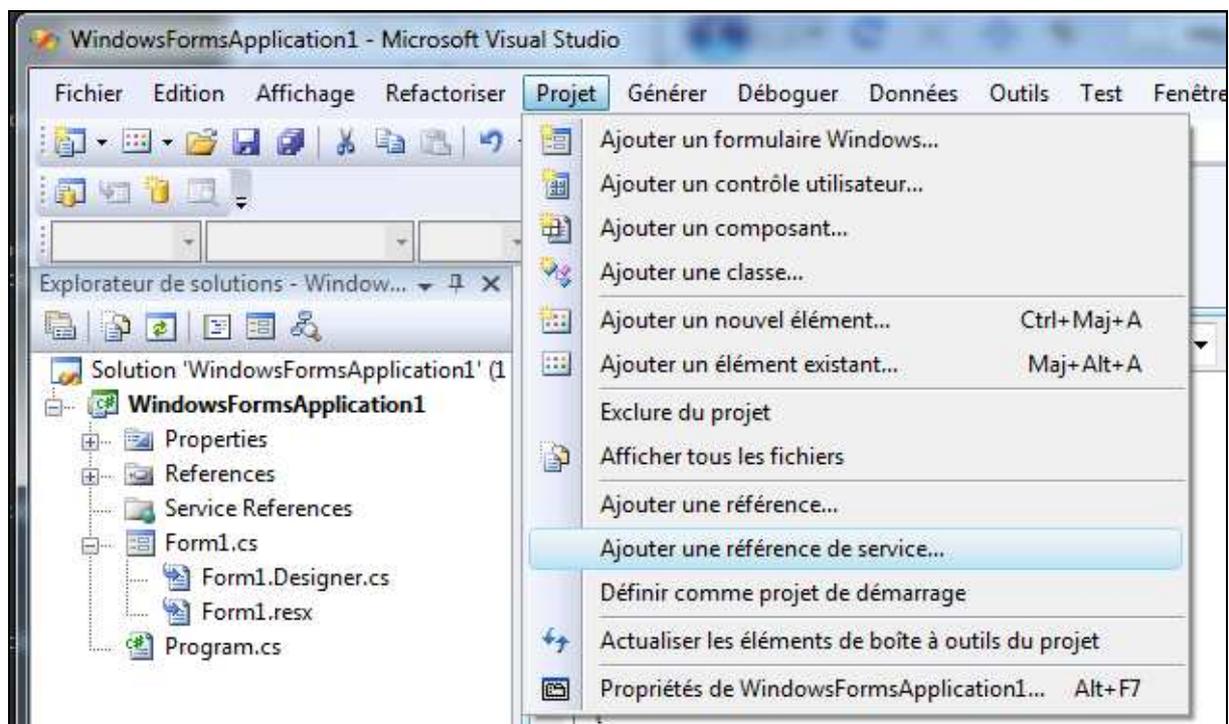
```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

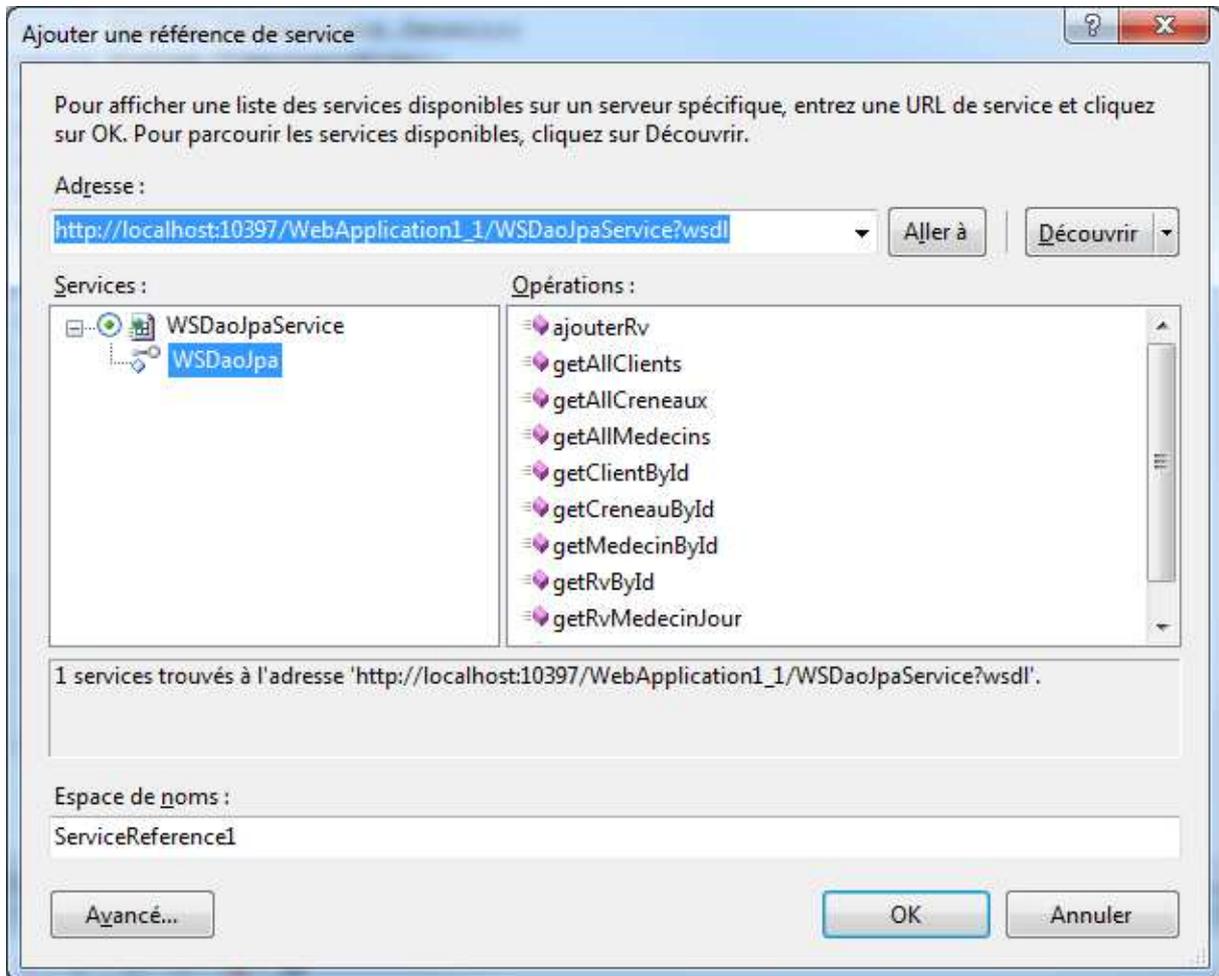
        private void button1_Click(object sender, EventArgs e)
        {
        }
    }
}
```

Etape 2.2. Créer une référence de service.

Faire **Projet / Ajouter une référence de service**



Choisir comme adresse : [http://localhost:10397/WebApplication1\\_1/WSDaoJpaService?wsdl](http://localhost:10397/WebApplication1_1/WSDaoJpaService?wsdl)



Modifier le fichier **Form1.cs** comme suit :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using WindowsFormsApplication1.ServiceReference1;
using Client = WindowsFormsApplication1.ServiceReference1.clients;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
    }
}
```

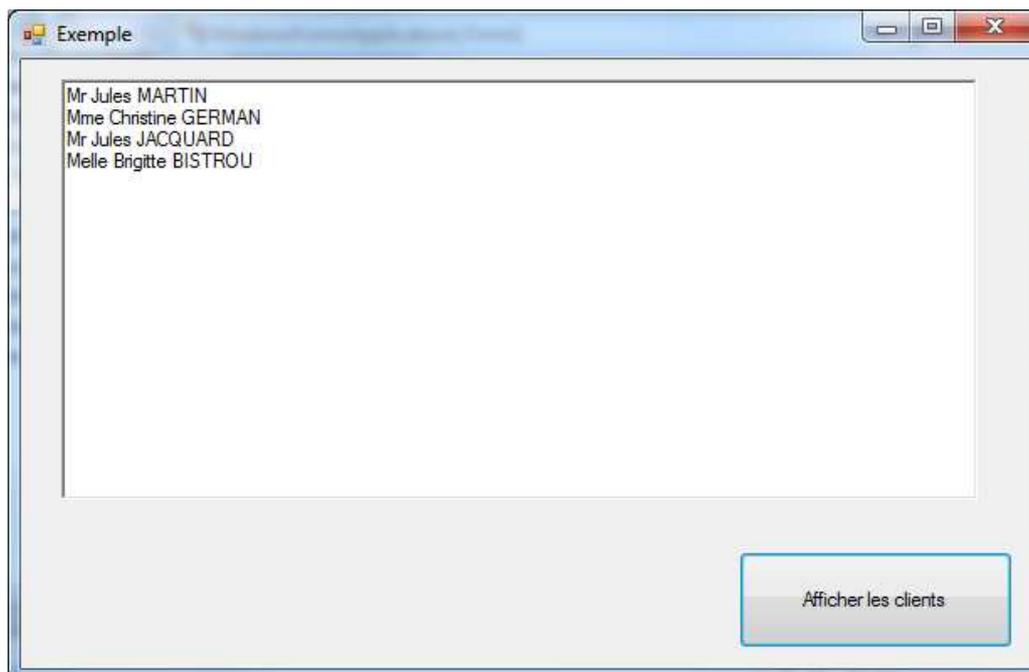
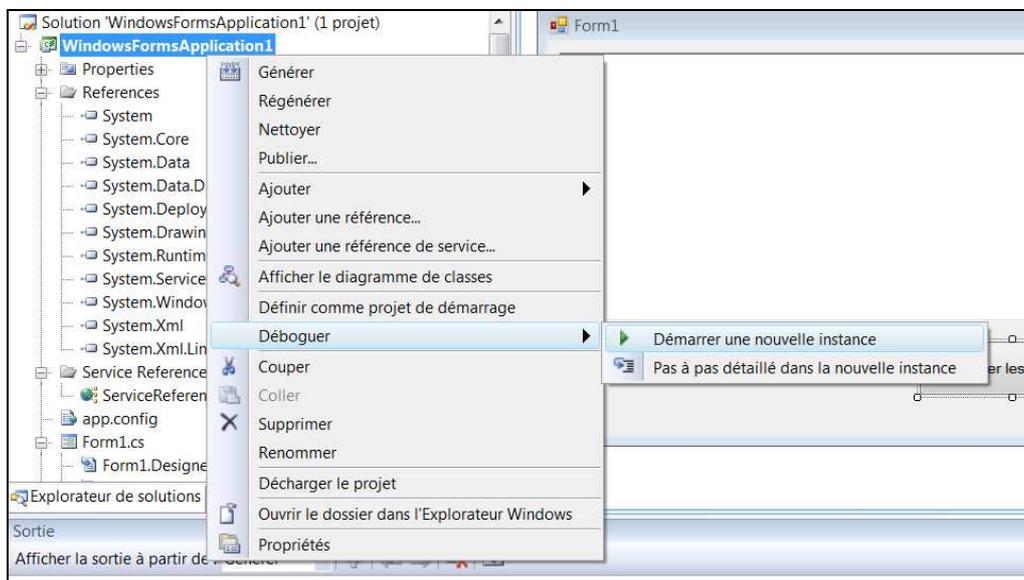
```

    {
        InitializeComponent();
    }

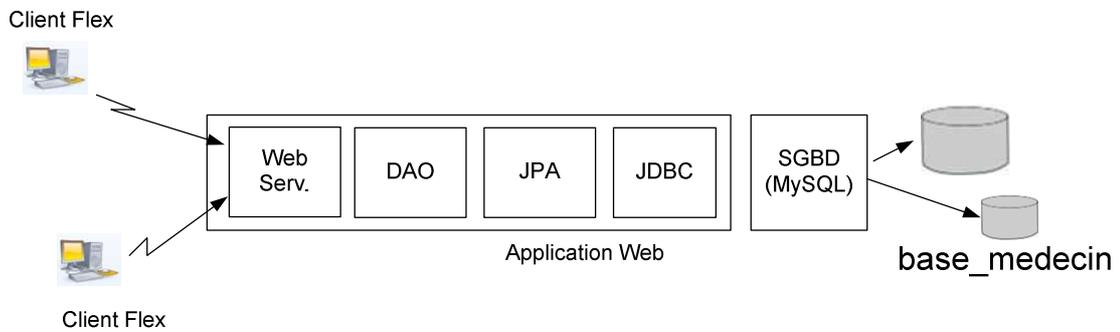
    private void button1_Click(object sender, EventArgs e)
    {
        WSDaoJpaClient dao = new WSDaoJpaClient();
        foreach (Client client in dao.getAllClients())
        {
            richTextBox1.AppendText(client.titre + " " +
                client.prenom + " " + client.nom + "\n");
        }
    }
}
}

```

Ceci donne à l'exécution :



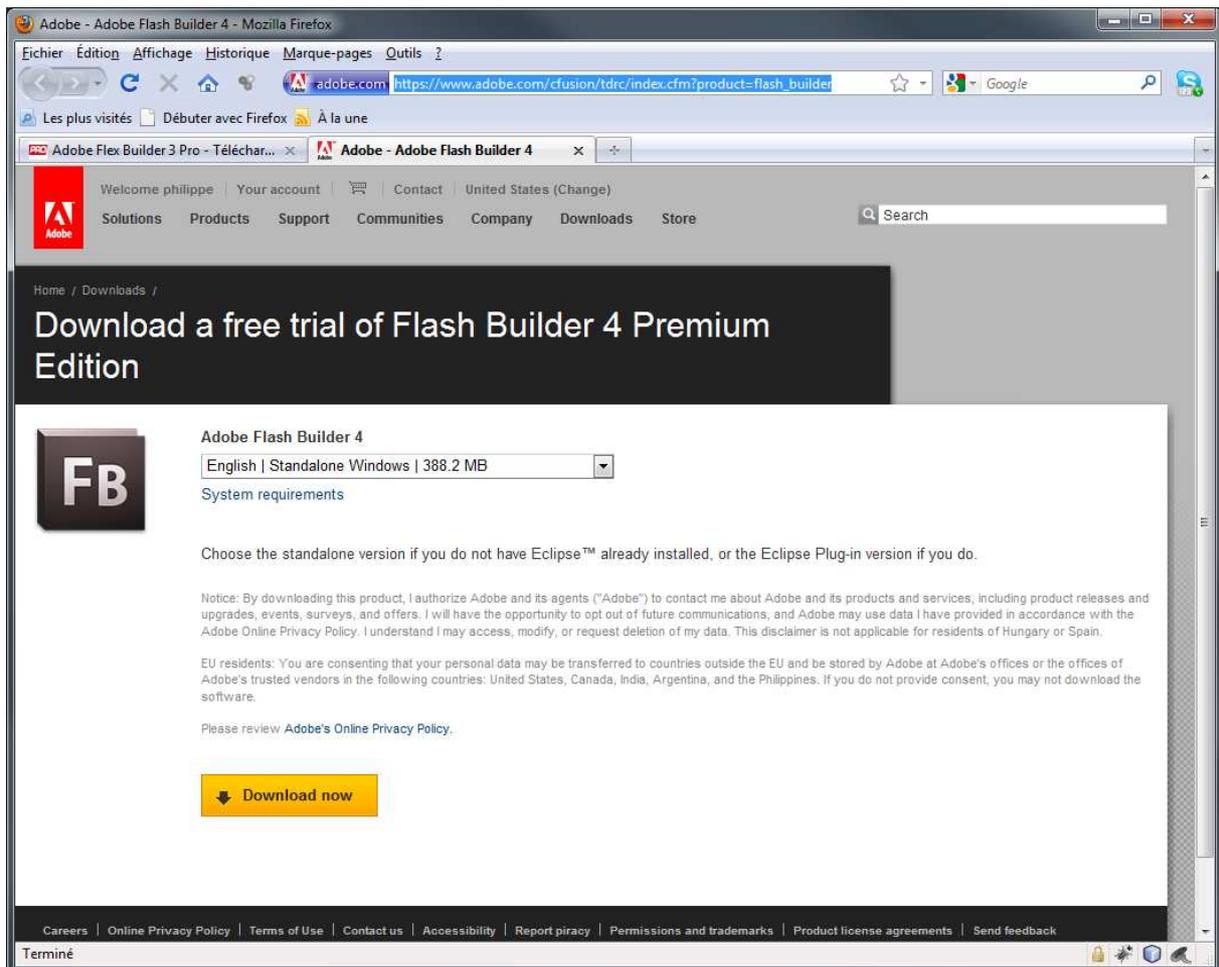
# Réalisation d'un client Flex

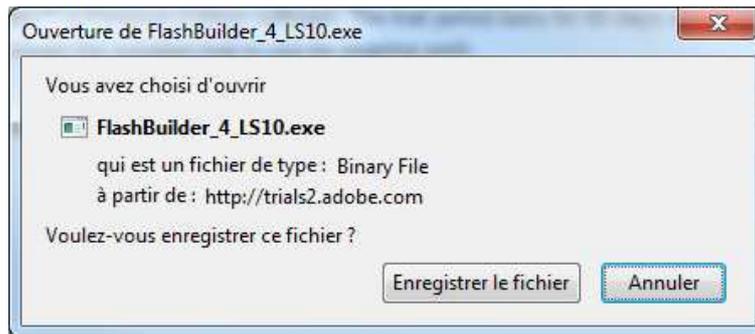


## 1) Télécharger l'environnement de développement Flex

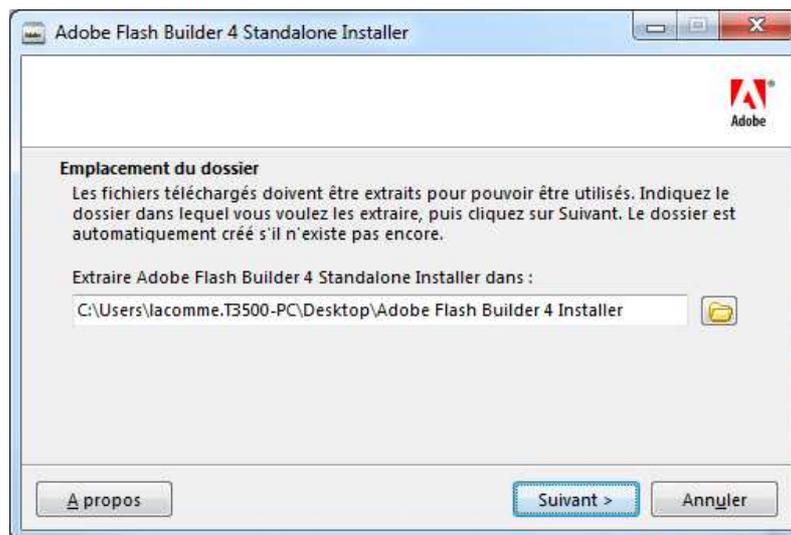
Adresse :

[https://www.adobe.com/cfusion/tdrc/index.cfm?product=flash\\_builder](https://www.adobe.com/cfusion/tdrc/index.cfm?product=flash_builder)

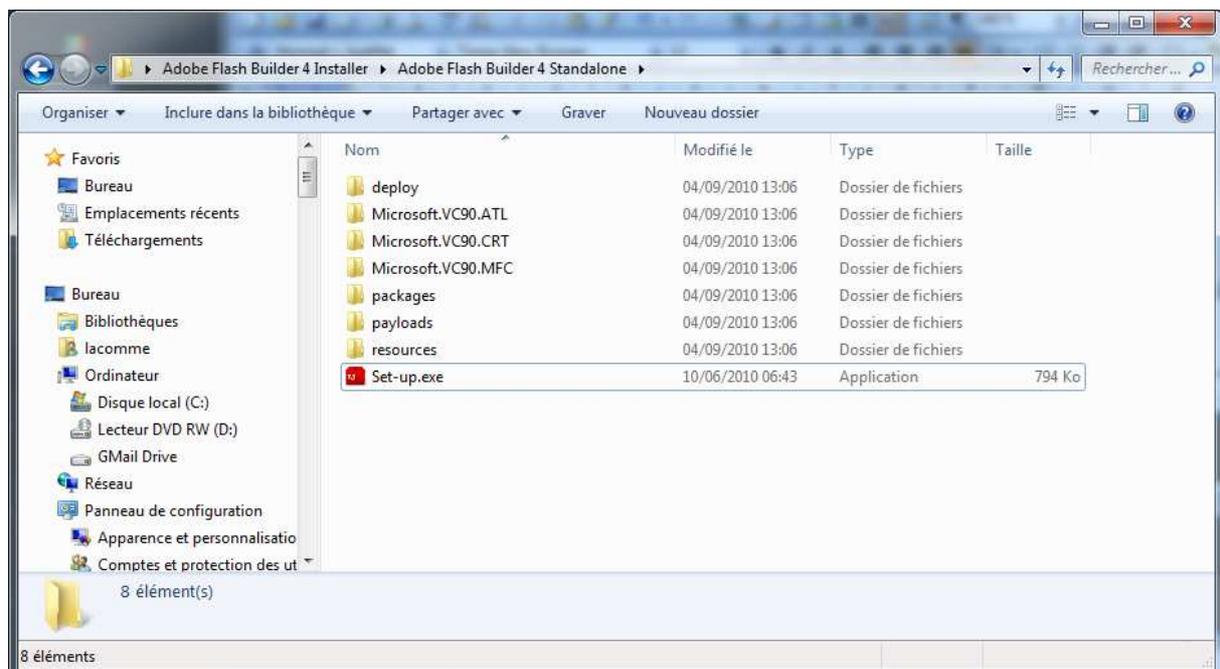


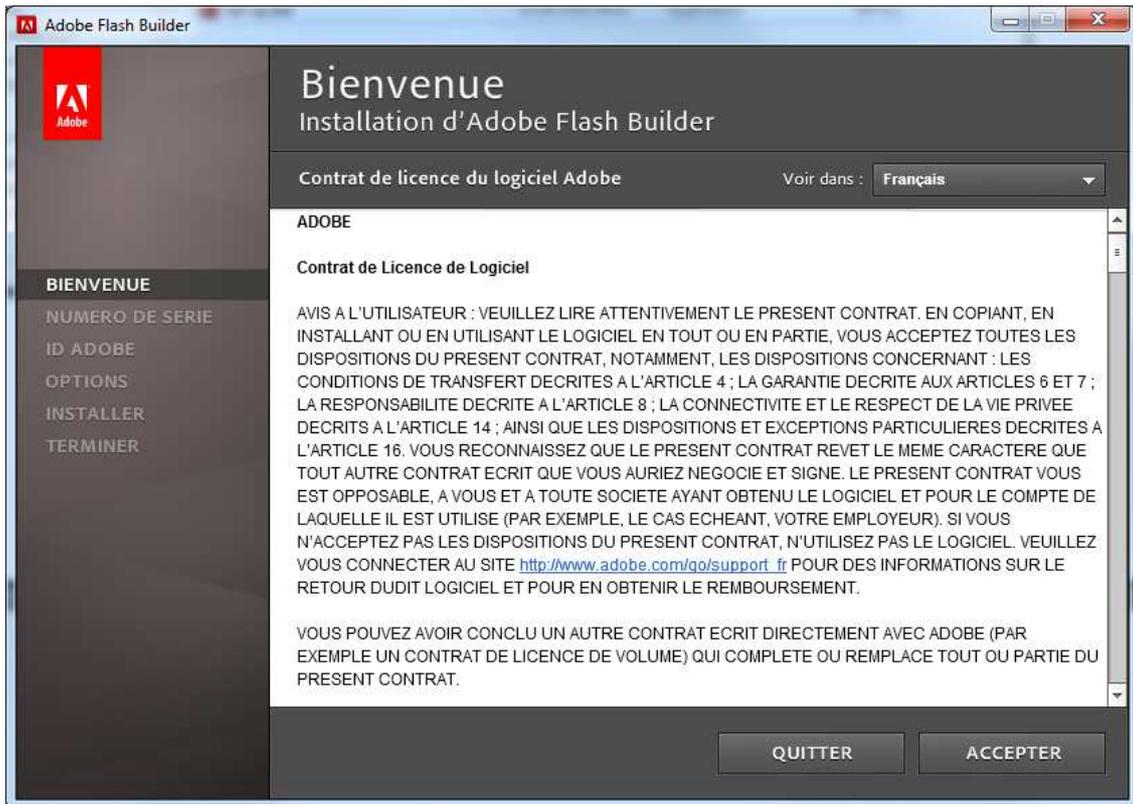


Lancer l'installation :

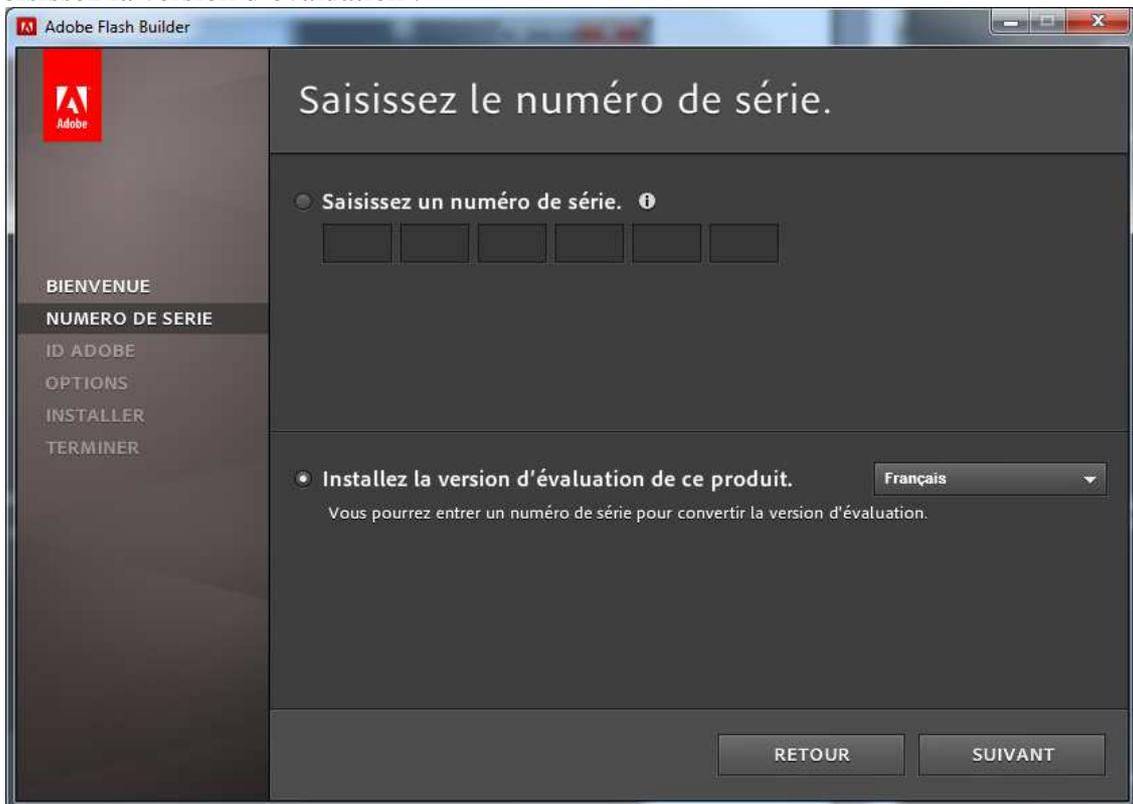


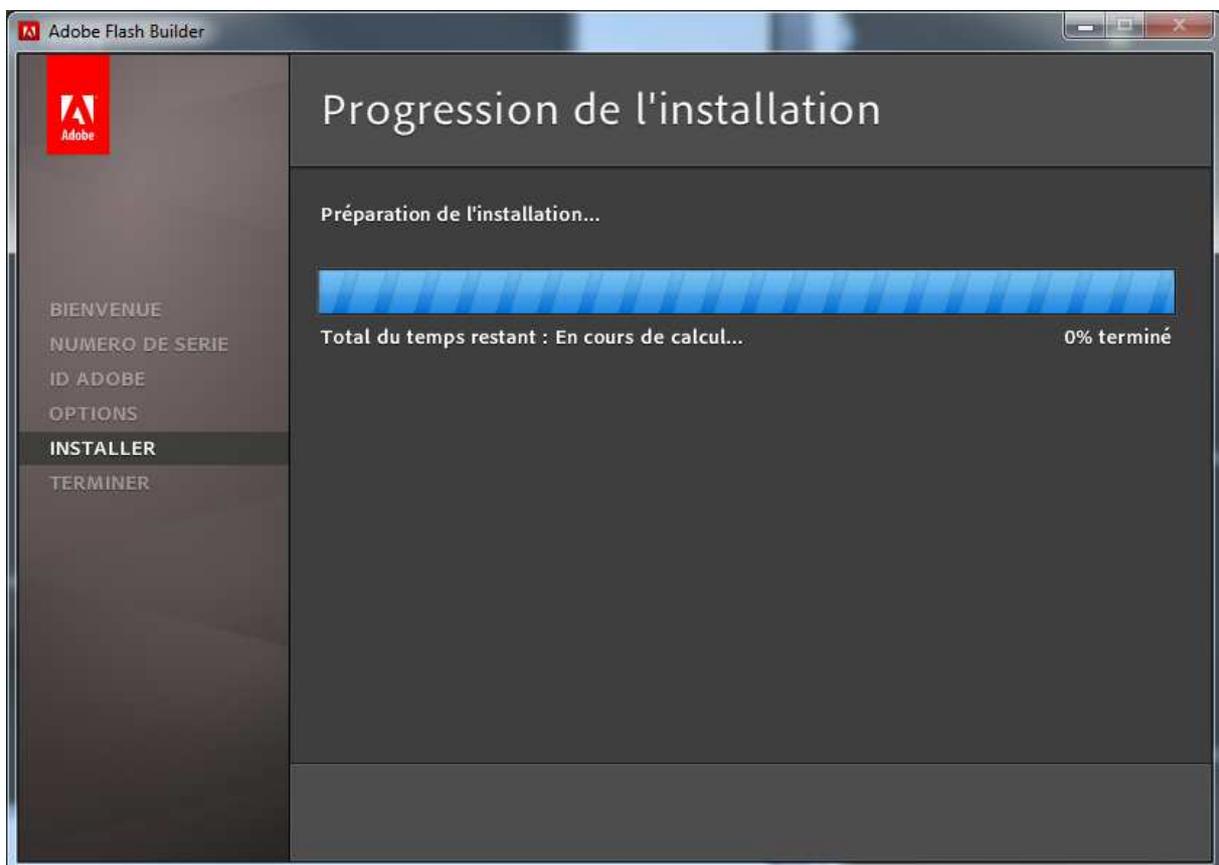
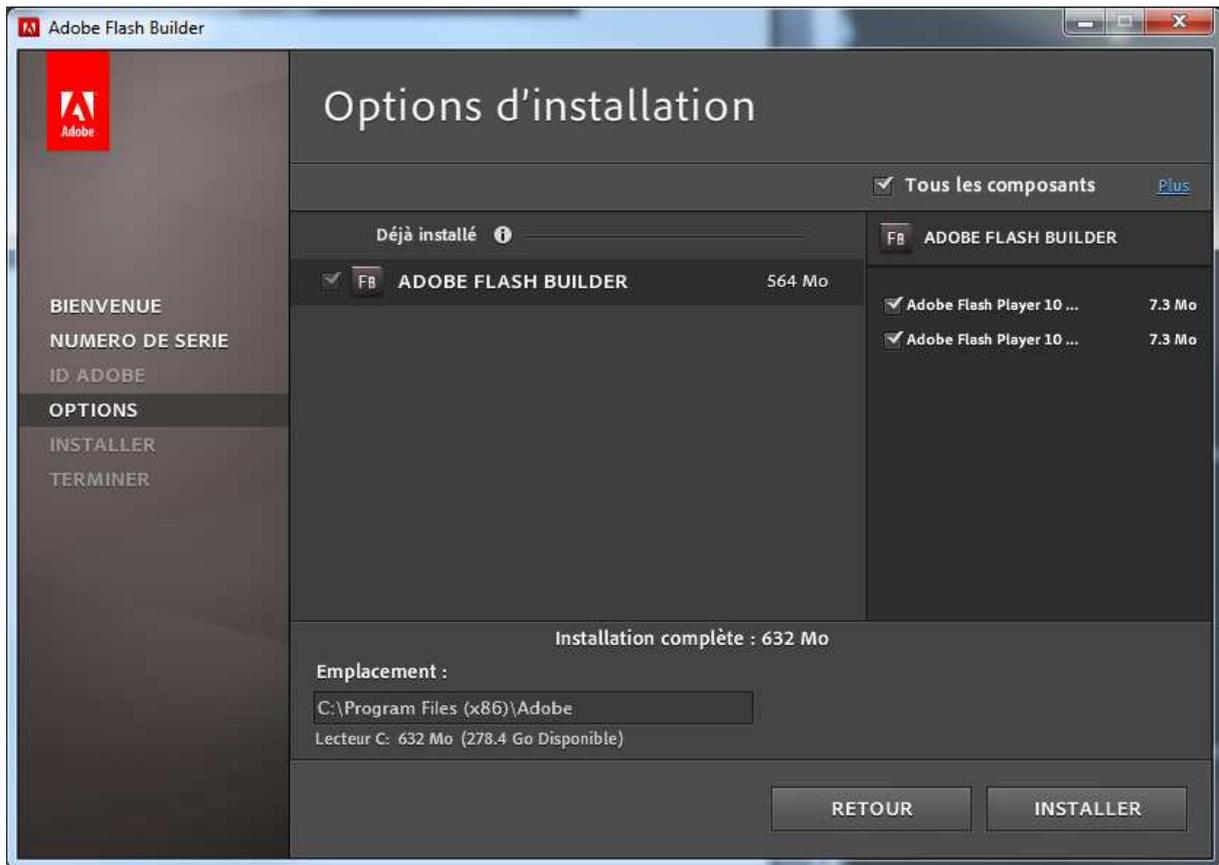
A partir d'une répertoire lancer ensuite **Set-up.exe** :



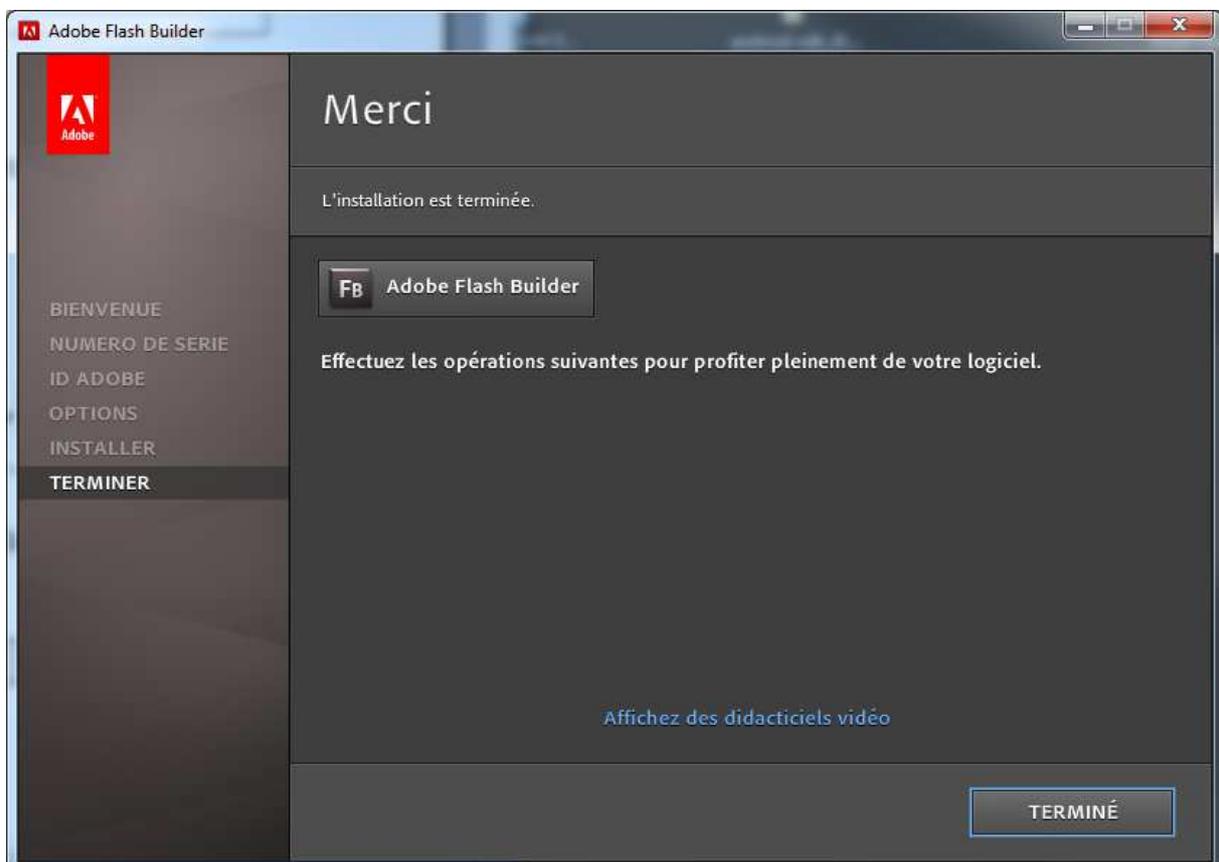
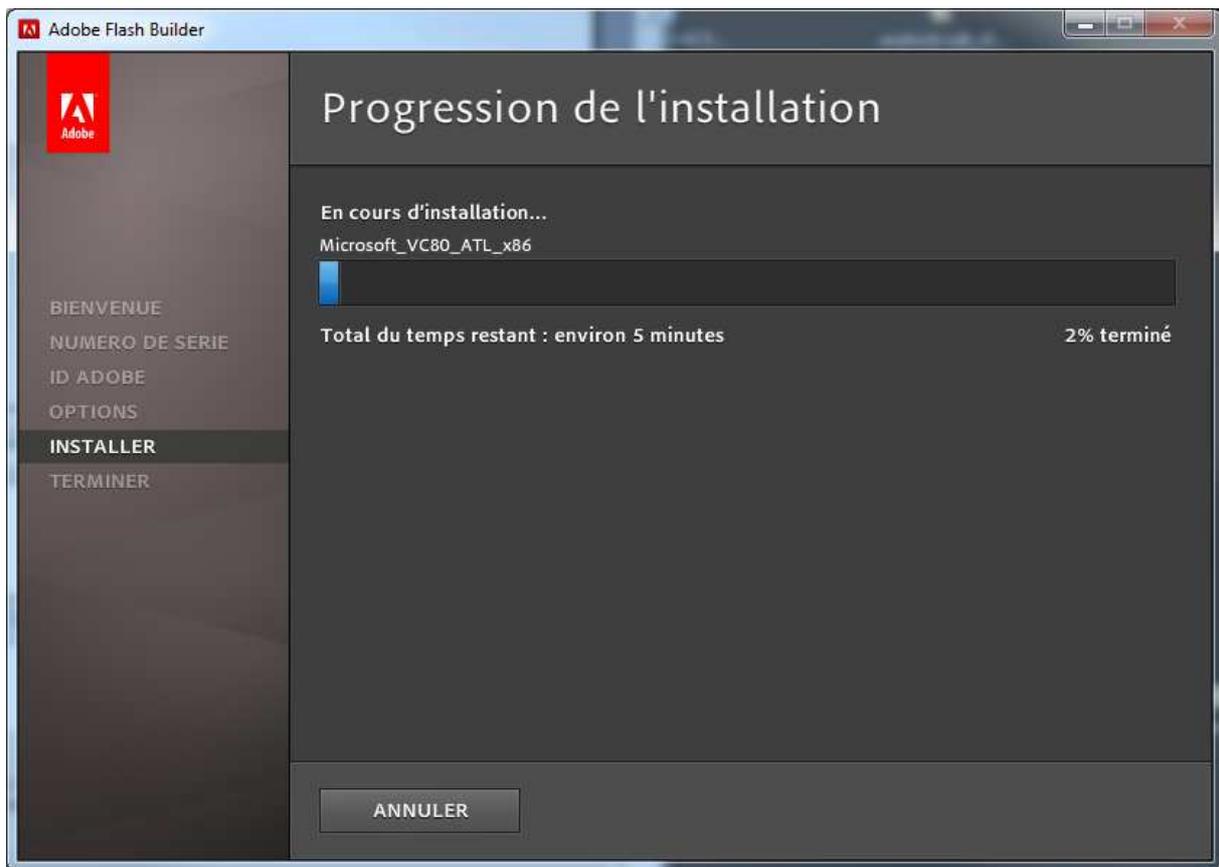


Choisissez la version d'évaluation :



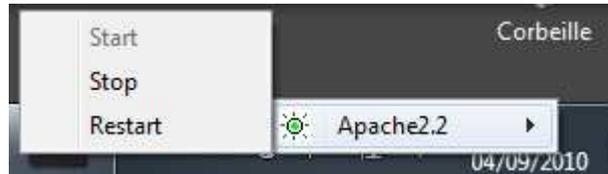


Patientez...



## 2) Un serveur Apache

Les applications Flex nécessitent un serveur Apache qui exécutera le service web distance implanté en Flex. Vérifier que vous avez un service de type Apache en cours de fonctionnement sur votre machine.



Sur ma machine, le répertoire d'installation de Apache est :

**C:\Program Files (x86)\Apache Software Foundation\Apache2.2**

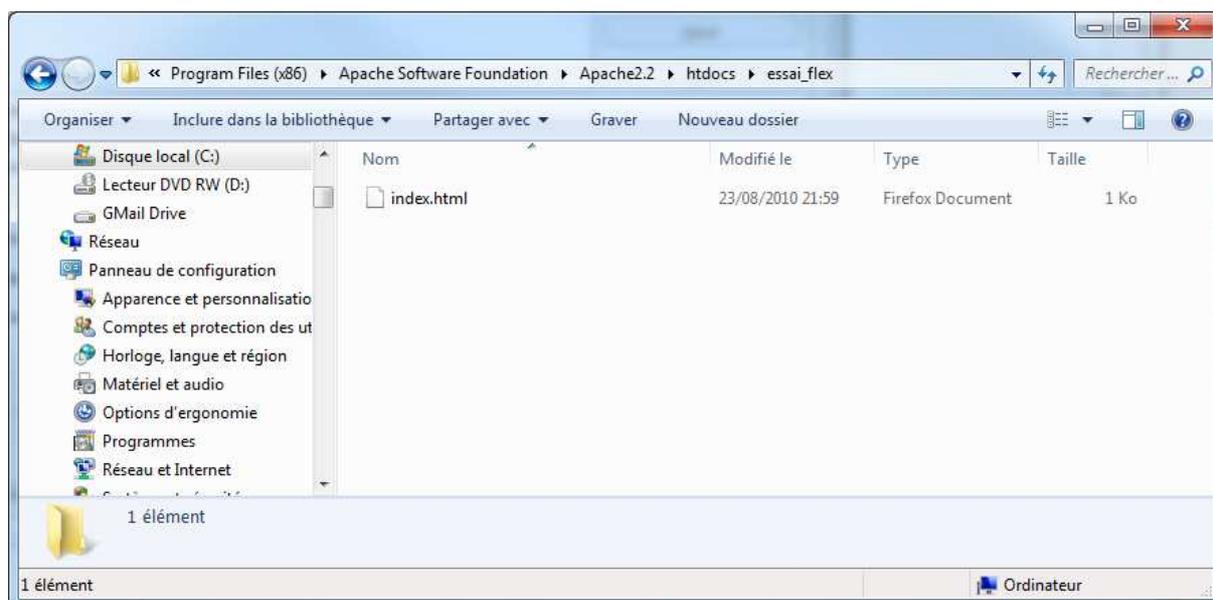
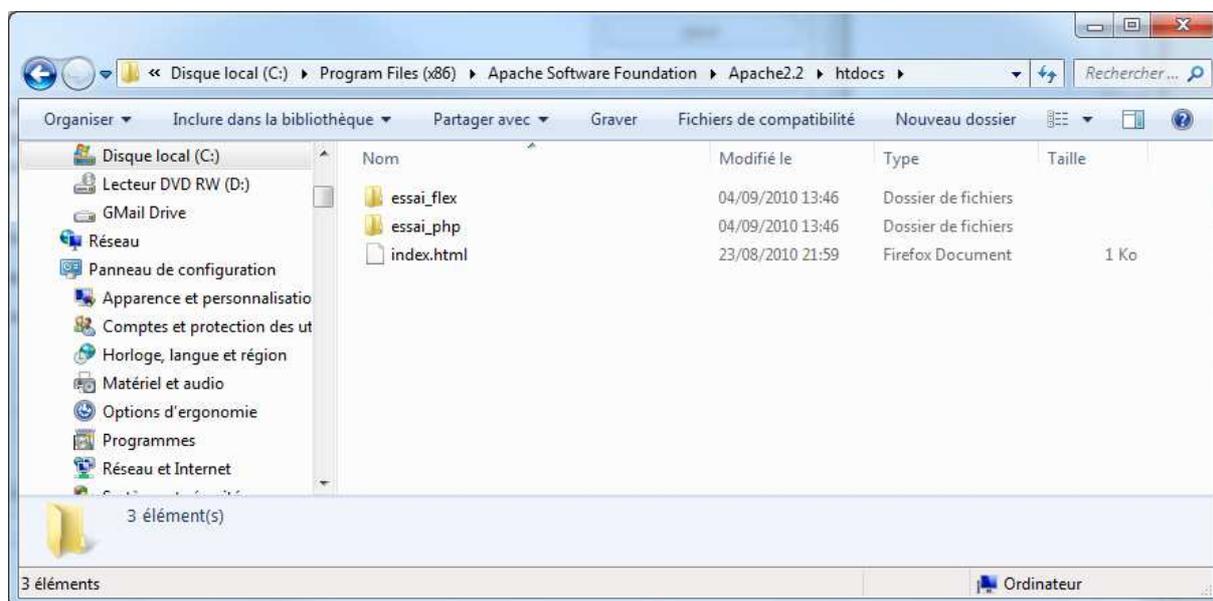
Le contenu se présente comme suit :

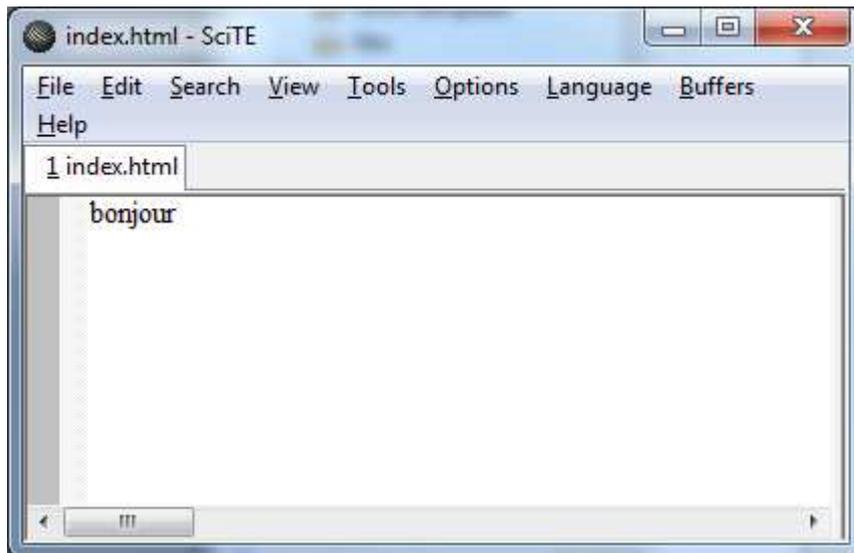
Nom	Modifié le	Type	Taille
bin	25/08/2010 15:53	Dossier de fichiers	
cgi-bin	25/08/2010 15:53	Dossier de fichiers	
conf	25/08/2010 16:00	Dossier de fichiers	
error	25/08/2010 15:53	Dossier de fichiers	
htdocs	25/08/2010 16:50	Dossier de fichiers	
icons	25/08/2010 15:53	Dossier de fichiers	
logs	25/08/2010 15:59	Dossier de fichiers	
manual	25/08/2010 15:53	Dossier de fichiers	
modules	25/08/2010 15:53	Dossier de fichiers	
ABOUT_APACHE.txt	21/11/2004 12:50	Document texte	15 Ko
CHANGES.txt	21/07/2010 13:25	Document texte	110 Ko
INSTALL.txt	18/09/2008 13:16	Document texte	5 Ko
LICENSE.txt	30/07/2010 16:29	Document texte	36 Ko
NOTICE.txt	30/07/2010 16:29	Document texte	2 Ko
OPENSSL-NEWS.txt	30/07/2010 16:29	Document texte	24 Ko
OPENSSL-README.txt	30/07/2010 16:29	Document texte	11 Ko
README.txt	09/01/2007 23:50	Document texte	6 Ko
README-win32.txt	15/10/2008 08:22	Document texte	2 Ko

Les fichiers doivent être stockés dans le répertoire htdocs :

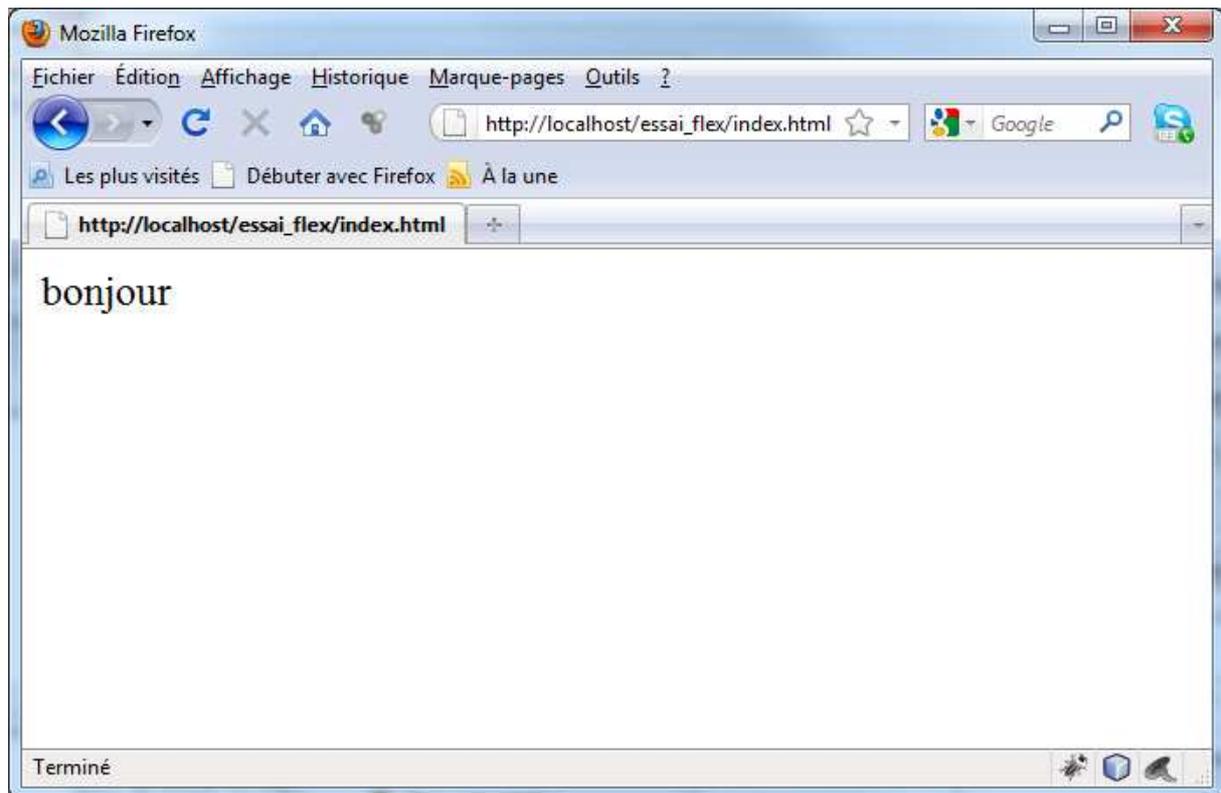
Nom	Modifié le	Type	Taille
essai_php	25/08/2010 16:50	Dossier de fichiers	
index.html	23/08/2010 21:59	Firefox Document	1 Ko

Je propose de créer un répertoire nommé **essai\_flex** et contenant un fichier **index.html** avec à l'intérieur un texte basique comme **bonjour** par exemple.





Ce qui donne dans votre navigateur :



### 3) Utilisation de Adobe Air

Il est possible de ne pas utiliser le serveur Apache et de créer des applications de bureau dont l'exécution nécessite Adobe AIR.

Si cela correspond à vos besoins, télécharger Adobe AIR ici :

<http://get.adobe.com/fr/air/>



Lancer l'installation :

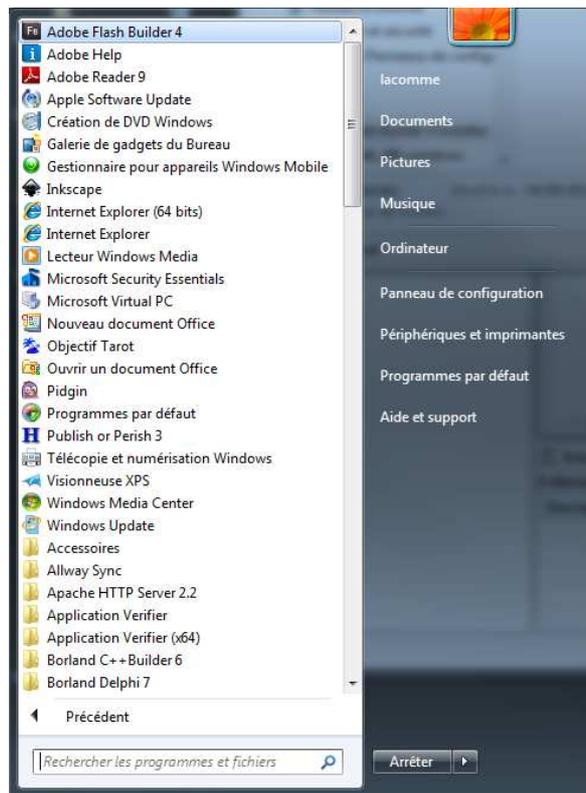


#### 4) Présentation rapide de Flex – Exemple 1.

Nous allons présenter ici les principaux objets graphiques sous Flex et la notion de classe.

##### **Etape 4.1.** Création d'un projet Flex.

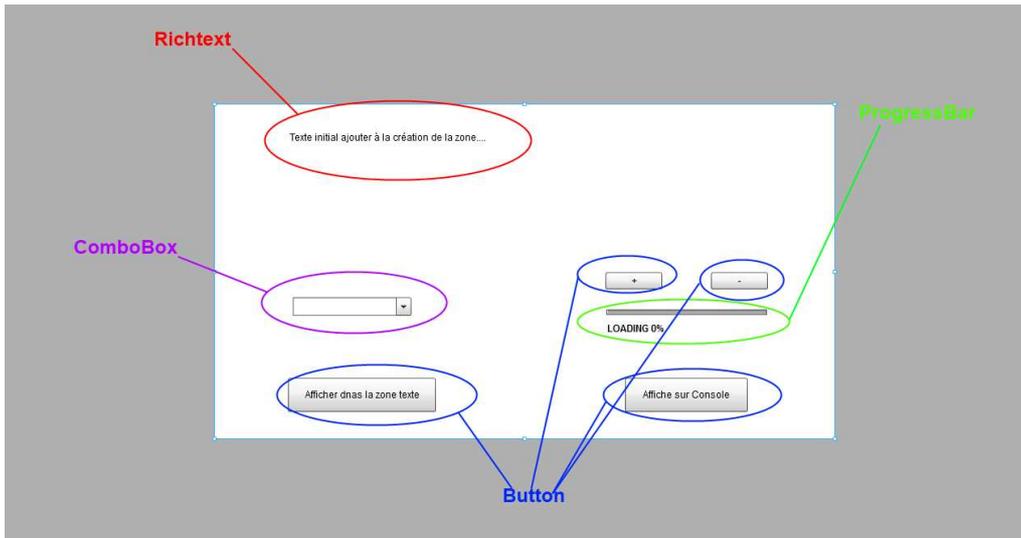
Lancer **Adobe Flash Builder 4** à partir du menu démarrer.



Faire Fichier / Nouveau Projet et choisir « Projet Flex ».

Prenez comme nom : **essai\_sur\_cours** par exemple.





Attention aux champs **id** de vos objets !

```

<s:Button x="263" y="302" label="Affiche sur Console" width="179" height="37"
click="button1_clickHandler(event)"/>
<s:RichEditableText id="Zone" x="35" y="35" text="Texte initial ajouté à la création de la
zone....&#xd;" width="408" height="130"/>

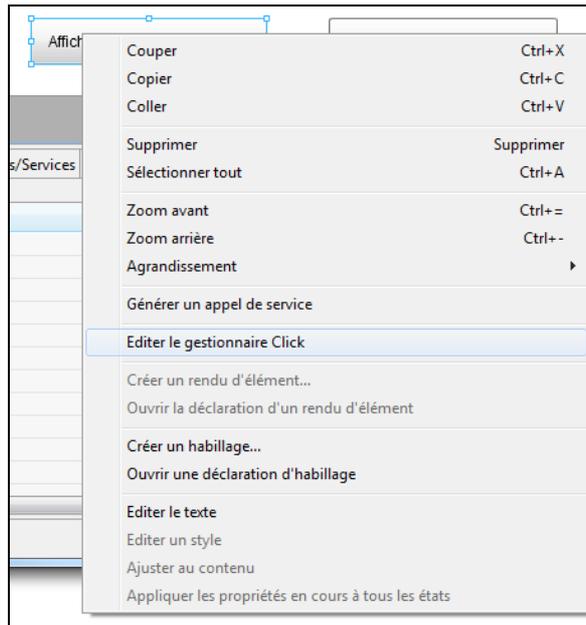
<s:ComboBox x="39" y="217">
  <s:ArrayCollection>
    <fx:String> Madame</fx:String>
    <fx:String> Monsieur</fx:String>
  </s:ArrayCollection>
</s:ComboBox>

<s:Button x="30" y="302" label="Afficher dans la zone de texte" width="185" height="37"
click="button2_clickHandler(event)"/>
<mx:ProgressBar id="Mabarre" x="233" y="249" mode="manual"/>
<s:Button x="237" y="219" label="+" click="button3_clickHandler(event)"/>
<s:Button x="363" y="219" label="-" click="button4_clickHandler(event)"/>

```

Ils sont très importants car ils sont utilisés dans le code gérant les événements par la suite.

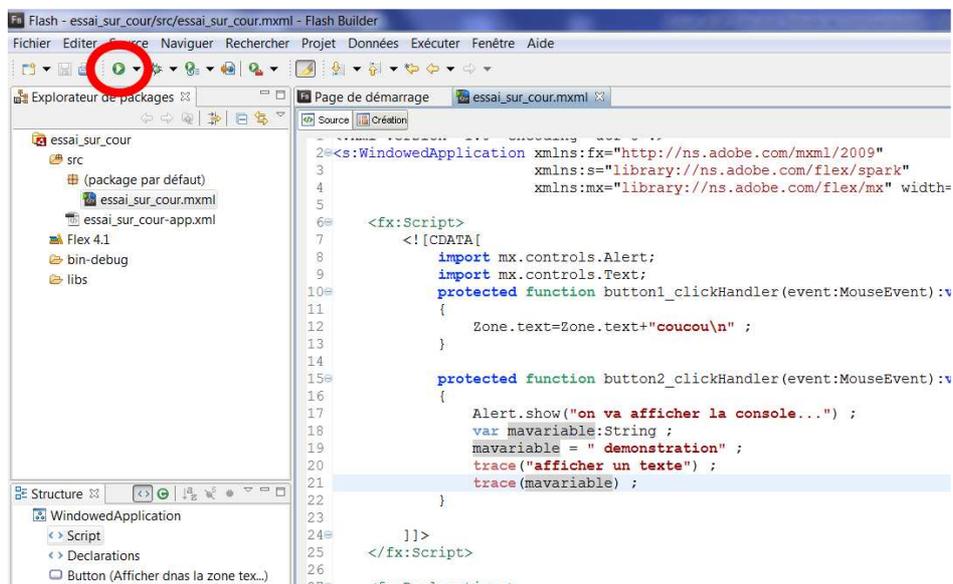
Choisir le bouton de gauche et faire un clic droit puis **Générer/Éditer le gestionnaire de Click**.

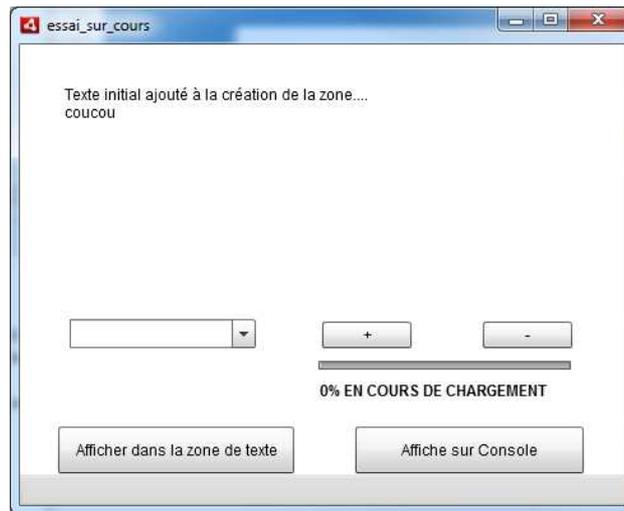


Ajouter le code suivant :

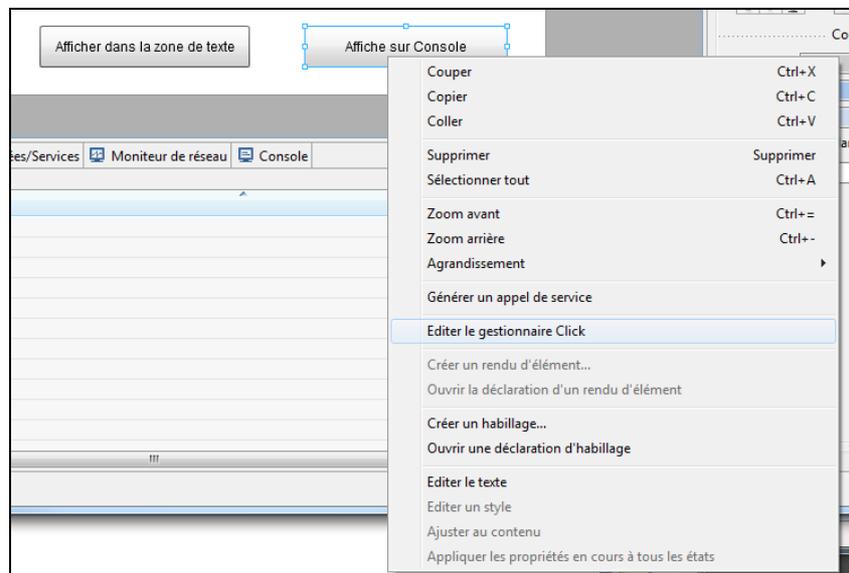
```
protected function button2_clickHandler(event:MouseEvent):void
{
    // TODO Auto-generated method stub
    Zone.text=Zone.text+"coucou\n";
}
```

Puis lancer le programme :





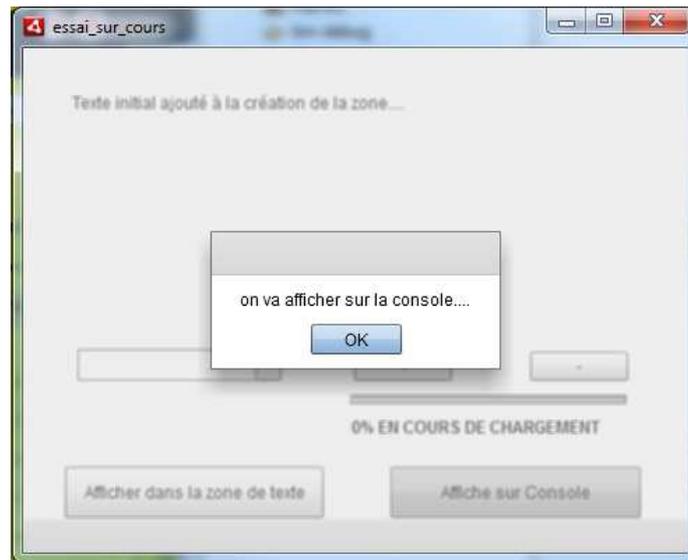
Choisir le bouton de droite et faire un clic droit puis **Générer/Éditer le gestionnaire de Click**.



Modifier le code sous le bouton de la manière suivante :

```
import mx.controls.Alert;
protected function button1_clickHandler(event:MouseEvent):void
{
    // TODO Auto-generated method stub
    Alert.show("on va afficher sur la console....");
    var mvariable:String;
    mvariable = " demonstration";
    trace("afficher un texte");
    trace(mvariable);
}
```

Ce qui donne à l'exécution :



Choisir le bouton « + » et faire un clic droit puis **Editer le gestionnaire de Click**.

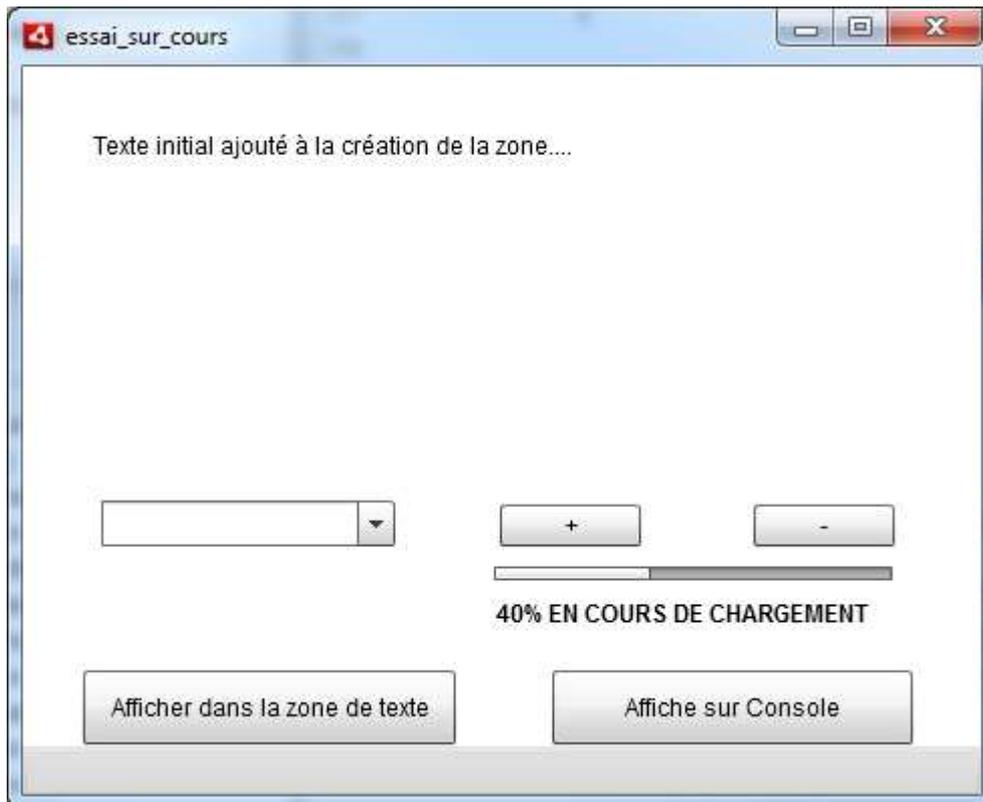
Ajouter le code suivant :

```
protected function button3_clickHandler(event:MouseEvent):void
{
    // TODO Auto-generated method stub
    Mabarre.mode="manual" ;
    var courant:int = Mabarre.value;
    if (courant<50)
        Mabarre.setProgress(courant+5,50);
}
```

Procéder de manière similaire avec le bouton « - » :

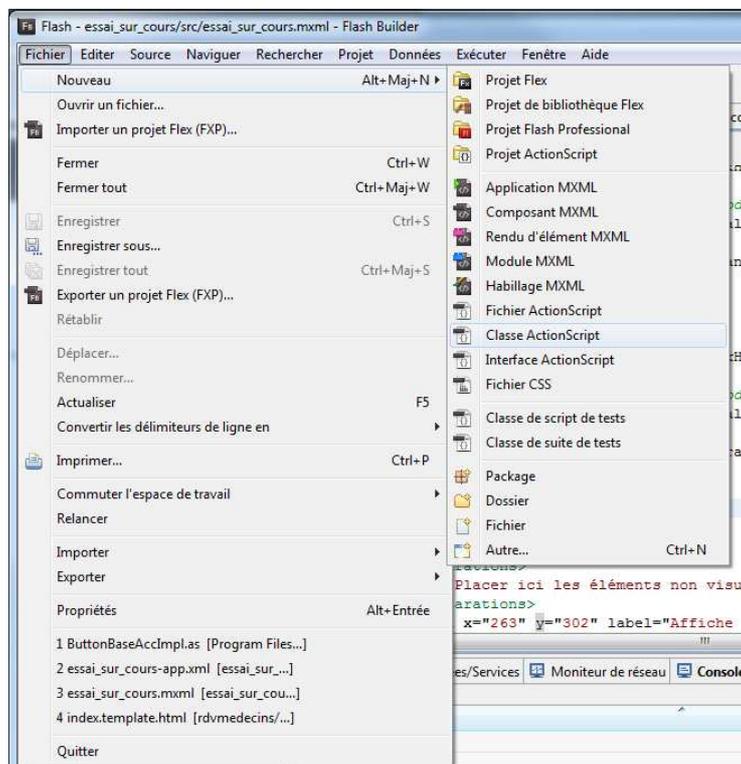
```
protected function button4_clickHandler(event:MouseEvent):void
{
    // TODO Auto-generated method stub
    Mabarre.mode="manual" ;
    var courant:int = Mabarre.value;
    if (courant>0)
        Mabarre.setProgress(courant-5,50);
}
```

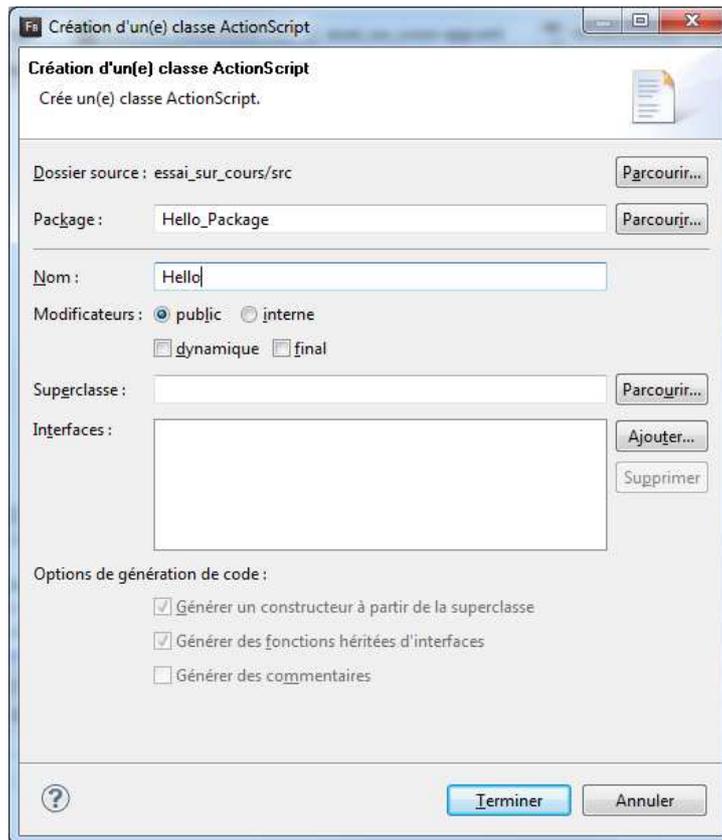
Ceci doit donner à l'exécution :



**Etape 4.2.** Création d'une classe.

Faire **Fichier / Nouveau / Classe ActionScript**





Le code généré ressemble à ceci :

```

package Hello_Package
{
    public class Hello
    {
        public function Hello()
        {
        }
    }
}

```

On peut par exemple, modifier la classe Hello comme suit :

```

package Hello_Package
{
    import mx.controls.Alert;
    public class Hello
    {
        public var Chaine:String;

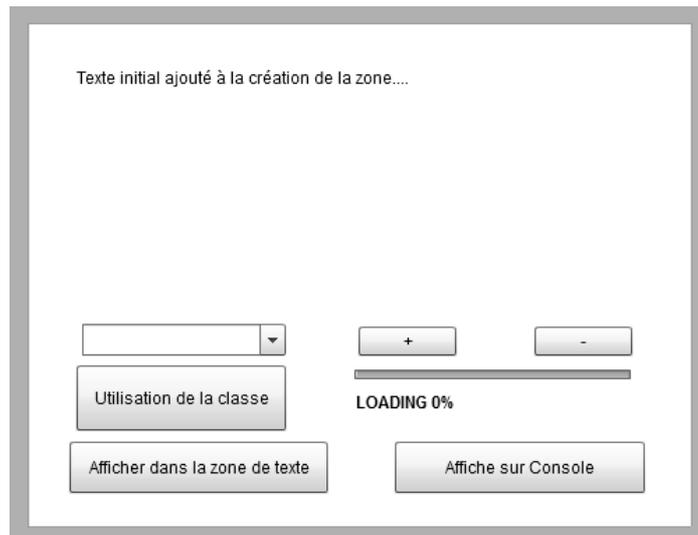
        public function Hello()
        {
            Chaine = "essai \n";
        }

        public function Afficher_Texte ()
        {
            Alert.show(Chaine);
        }

        public function Affecter_Texte (ma_chaine:String)
        {
            Chaine = ma_chaine;
        }
    }
}

```

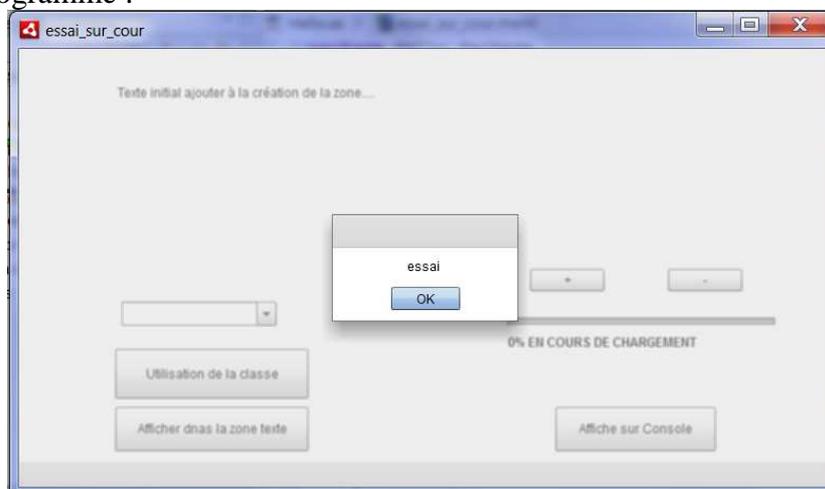
Rajouter un bouton sur l'interface...



Rajouter une procédure pour gérer l'événement clic.

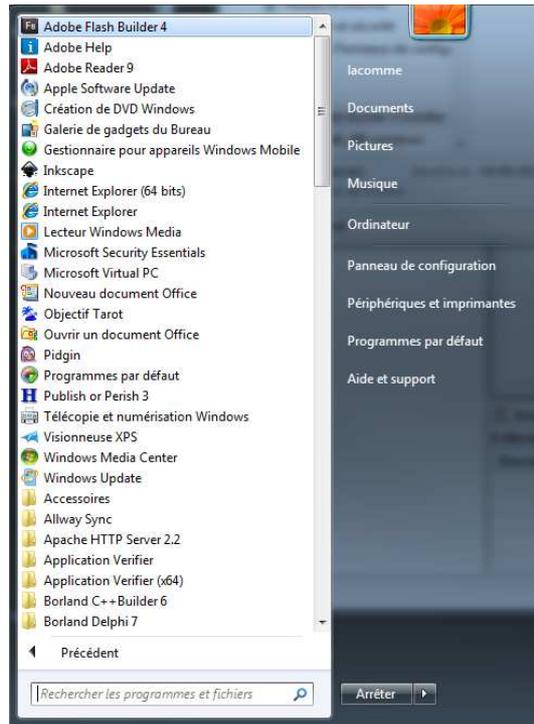
```
protected function button5_clickHandler(event:MouseEvent):void
{
    // TODO Auto-generated method stub
    var mon_instance : Hello;
    mon_instance = new Hello();
    mon_instance.Afficher_Texte();
}
```

Exécuter le programme :



## 5) Réalisation d'un client Flex

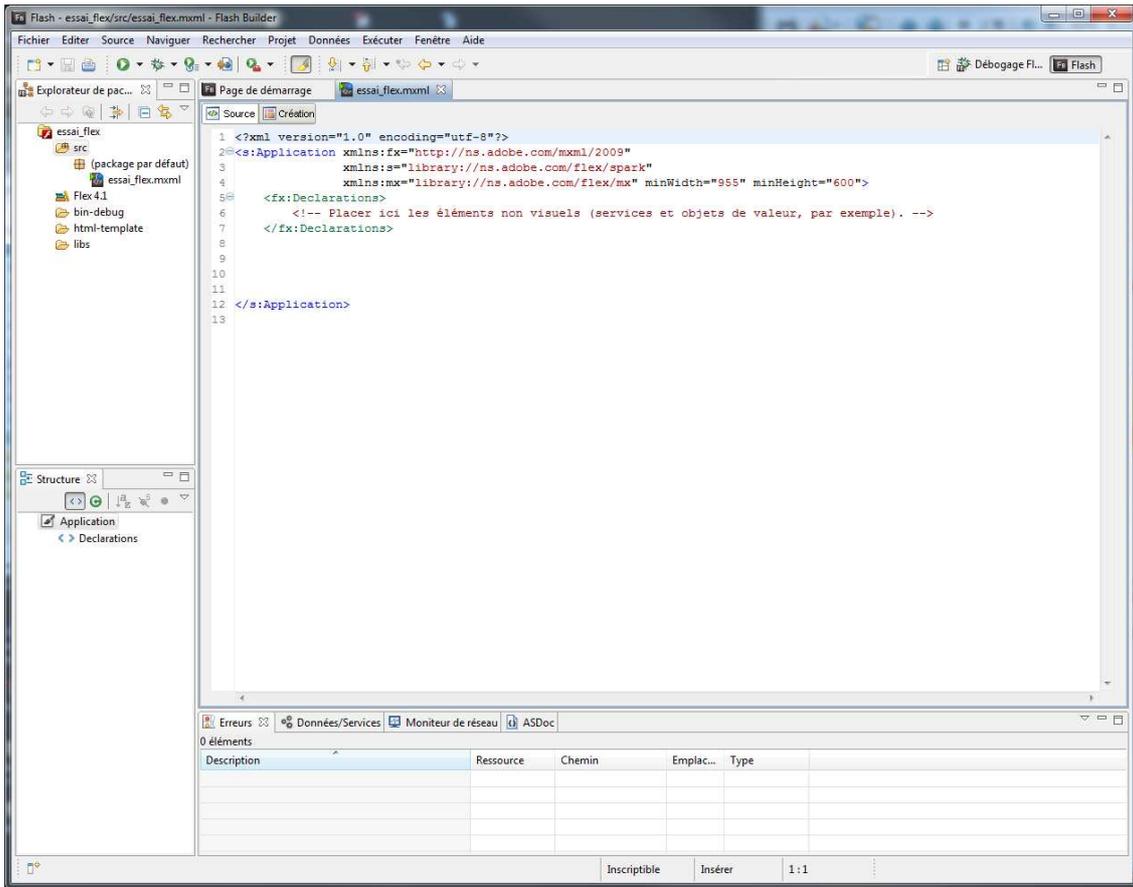
Etape 5.1. Lancer l'environnement Flex à partir du menu Démarrer.



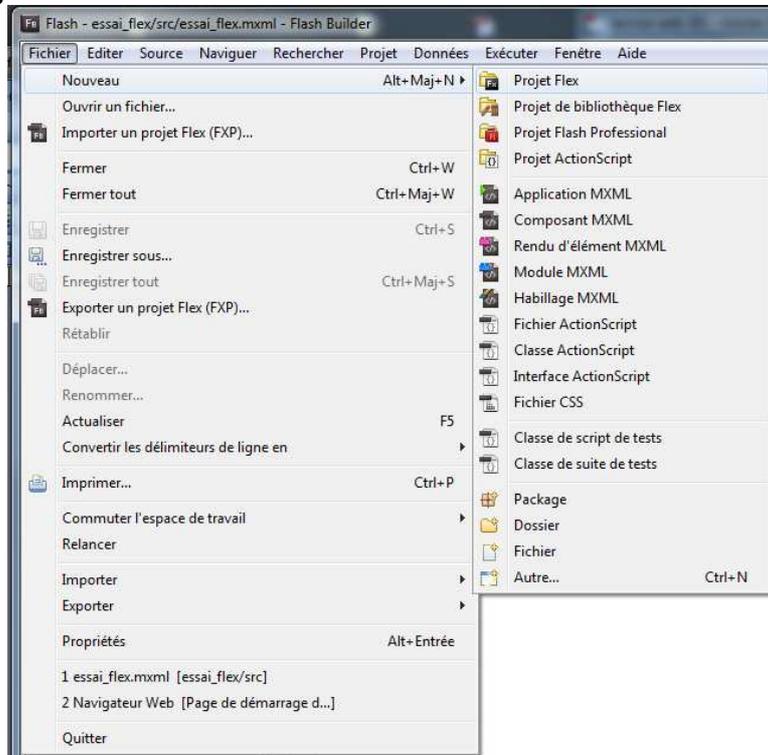
Débarassez vous de la fenêtre popup... en choisissant le bouton **Continuer**.

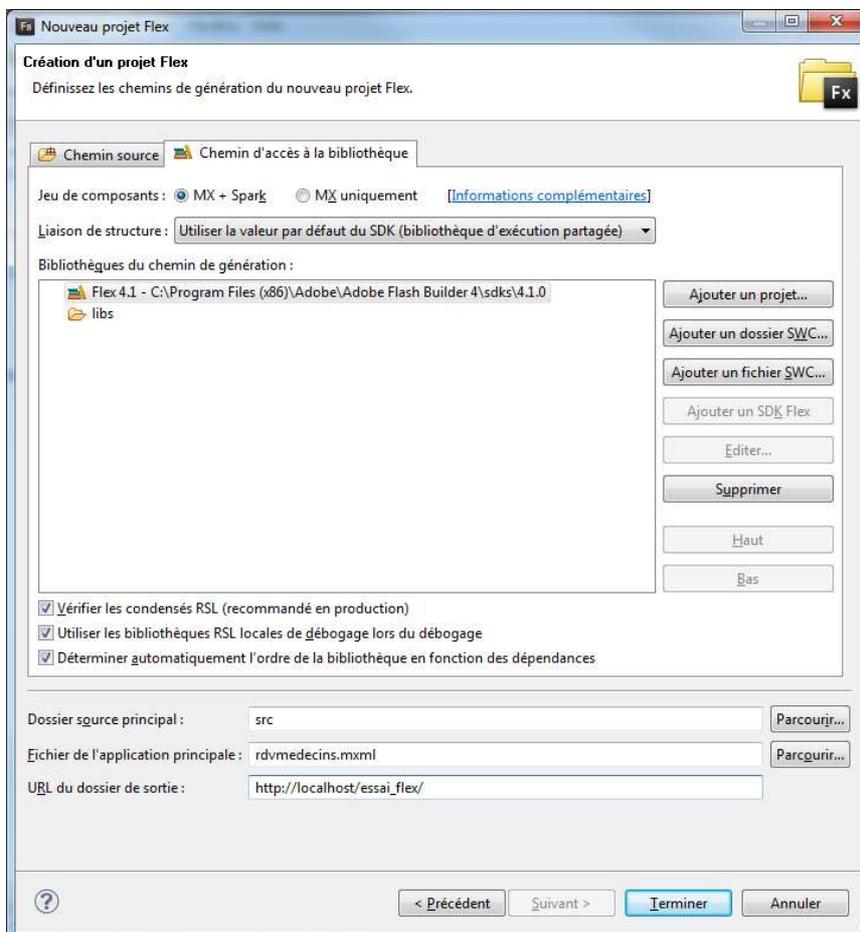
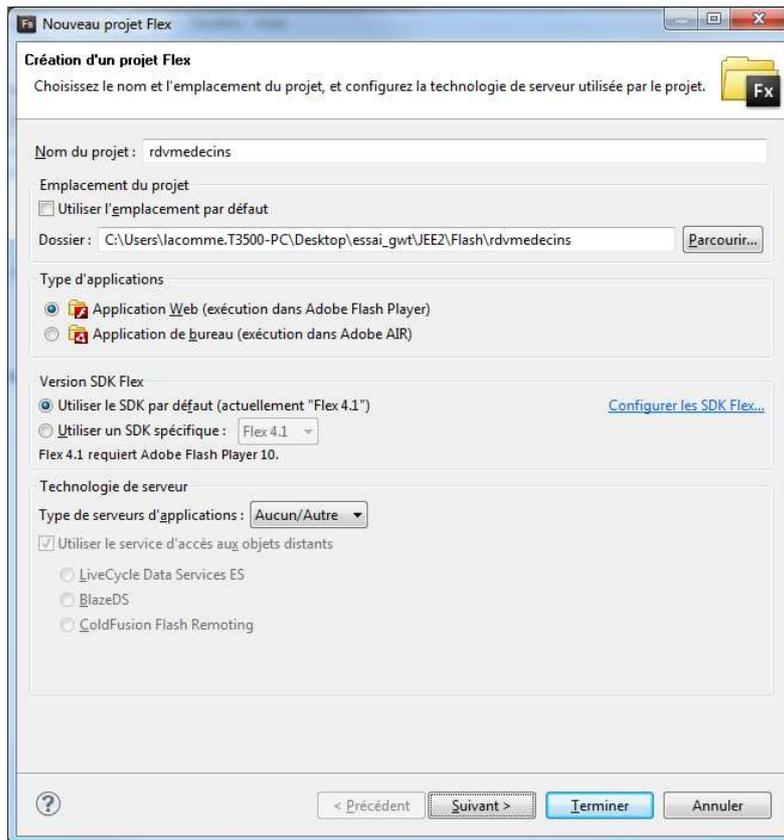


L'environnement de travail se présente comme suit :

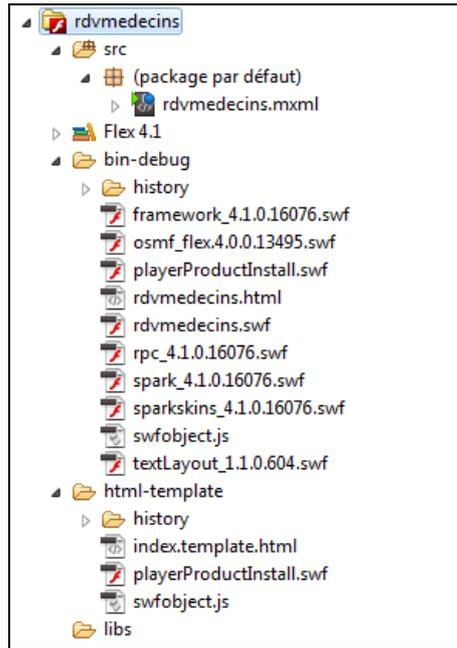


## Faire New / Project Flex



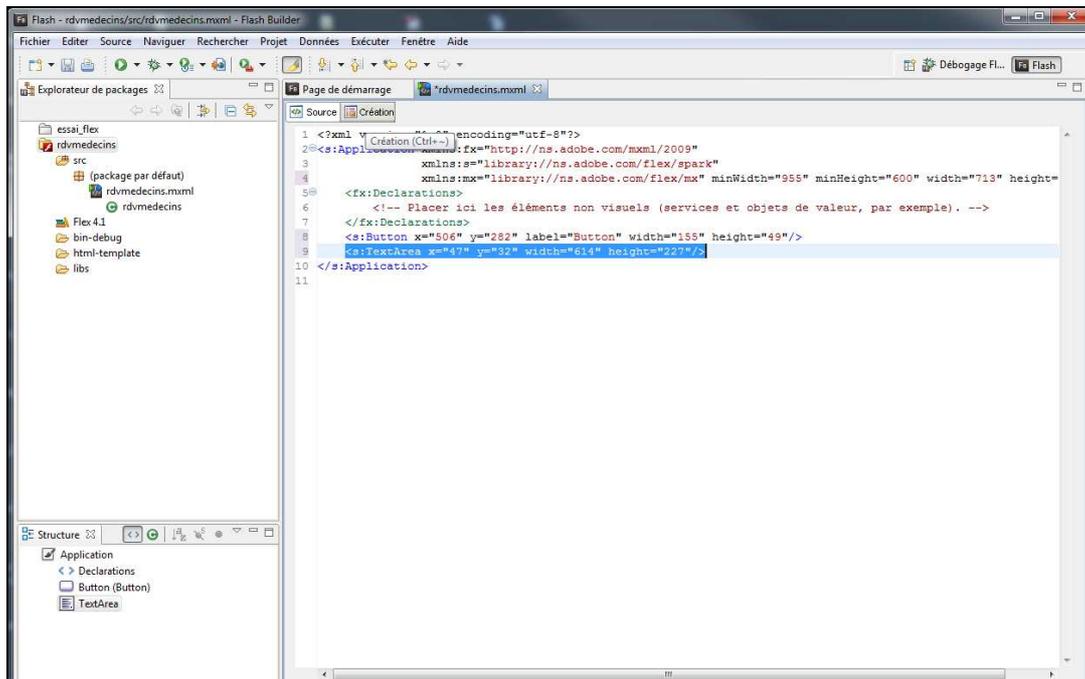


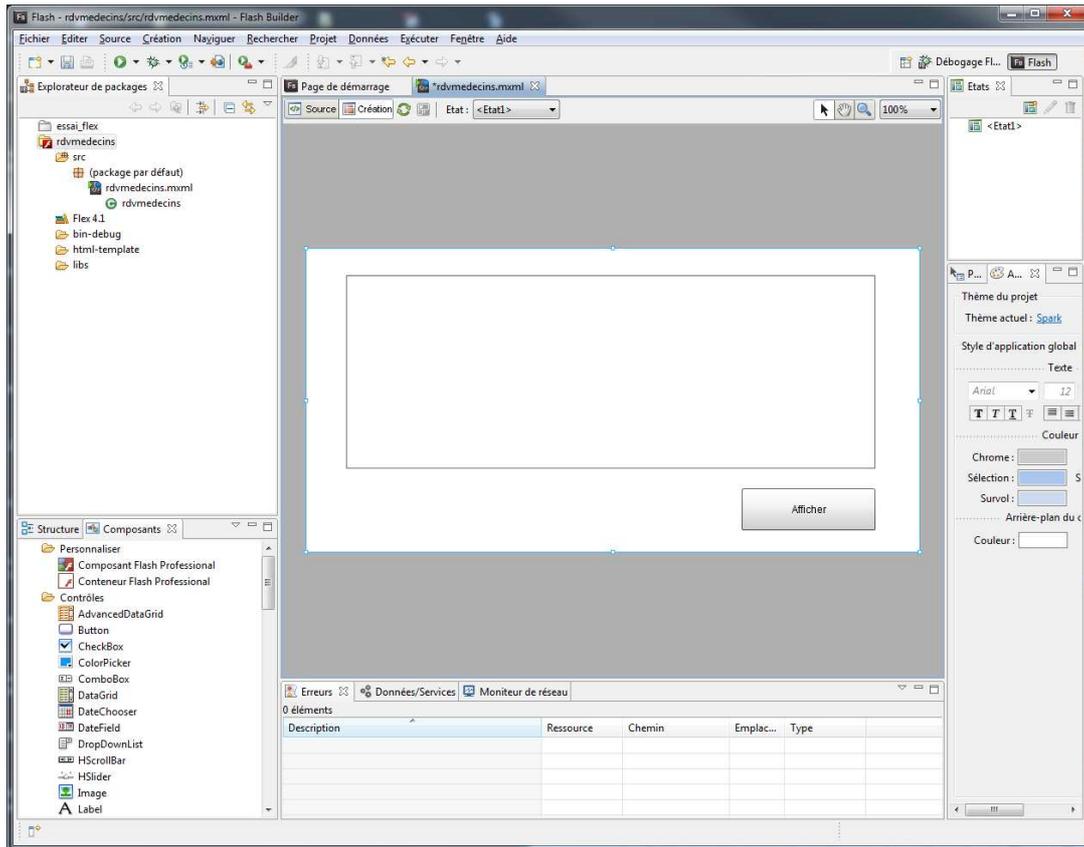
Le projet une fois créé se présente comme suit :



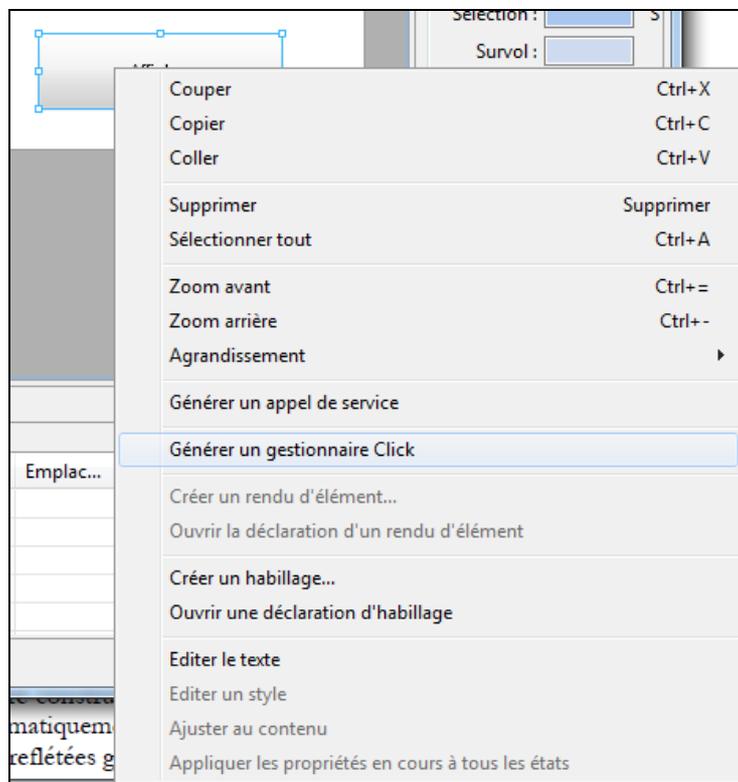
Etape 4.2. Concevez l'interface de votre client

Utilisez le bouton Création pour lancer le GUI.





Faites un clic droit sur le bouton et associez un gestionnaire clic au bouton.

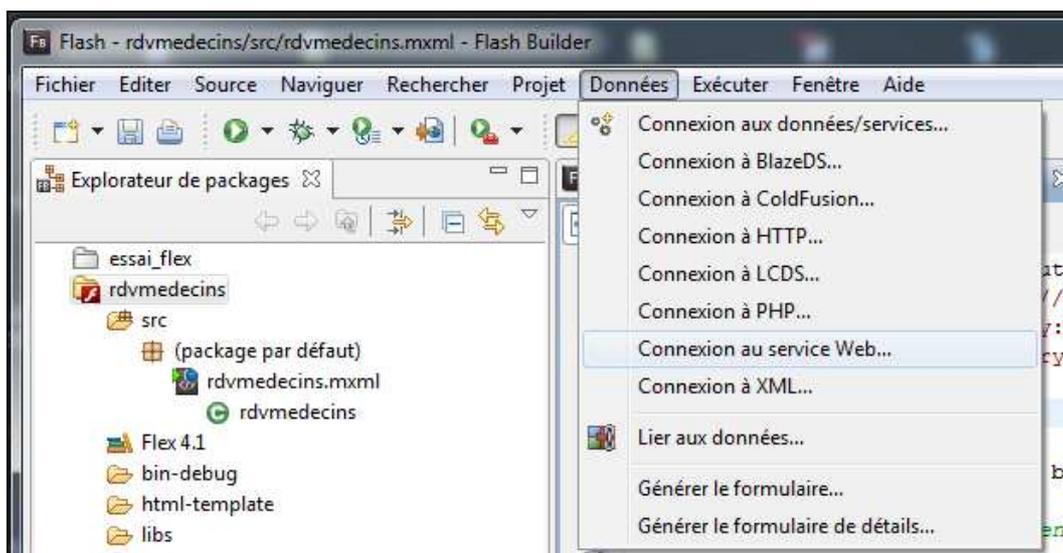


Le code doit ressembler à ceci :

```
Source Création
1 <?xml version="1.0" encoding="utf-8"?>
2 <s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
3     xmlns:s="library://ns.adobe.com/flex/spark"
4     xmlns:mx="library://ns.adobe.com/flex/mx" minWidth="955" minHeight="600" width="713" height=
5
6     <fx:Script>
7         <![CDATA[
8             protected function button1_clickHandler(event:MouseEvent):void
9             {
10                 // TODO Auto-generated method stub
11             }
12         ]]>
13     </fx:Script>
14
15     <fx:Declarations>
16         <!-- Placer ici les éléments non visuels (services et objets de valeur, par exemple). -->
17     </fx:Declarations>
18     <s:Button x="506" y="292" label="Afficher" width="155" height="49" click="button1_clickHandler(event)"/>
19     <s:TextArea x="47" y="32" width="614" height="227"/>
20 </s:Application>
21
```

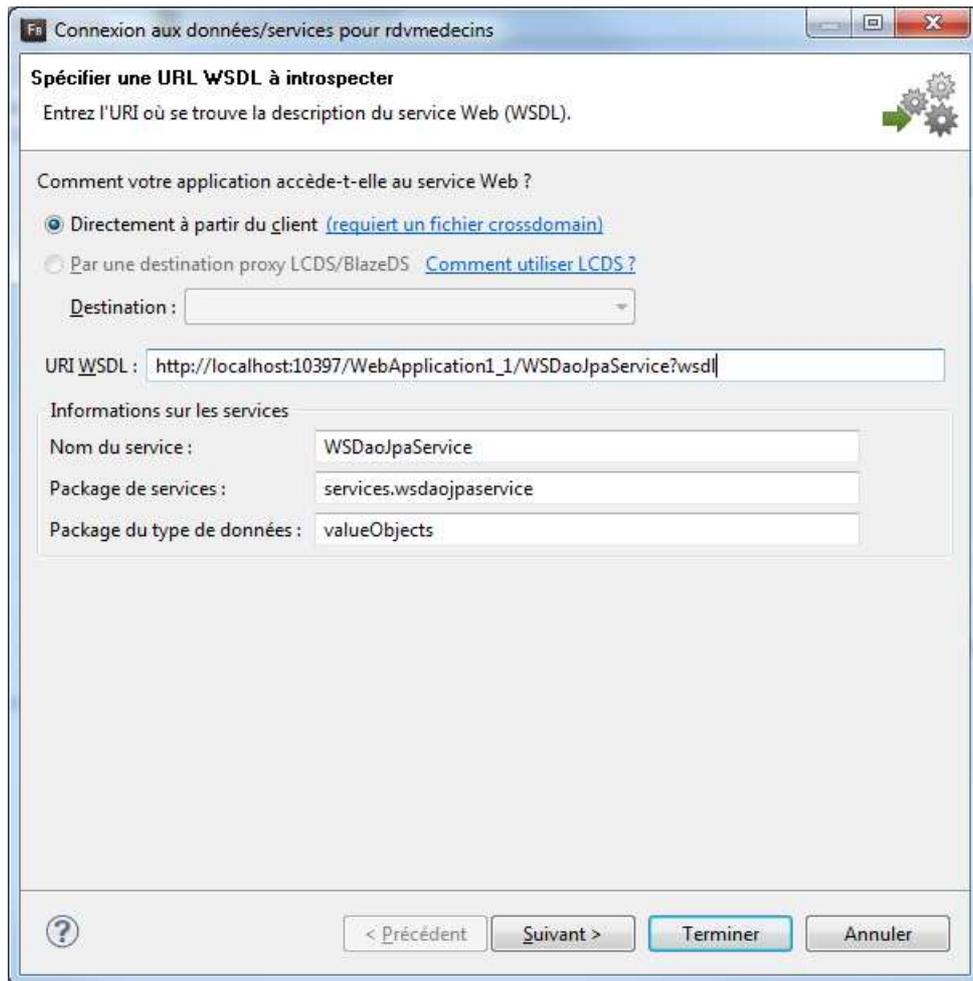
Etape 4.3. Connexion aux web services

Faire **Données / Connexion au service Web.**

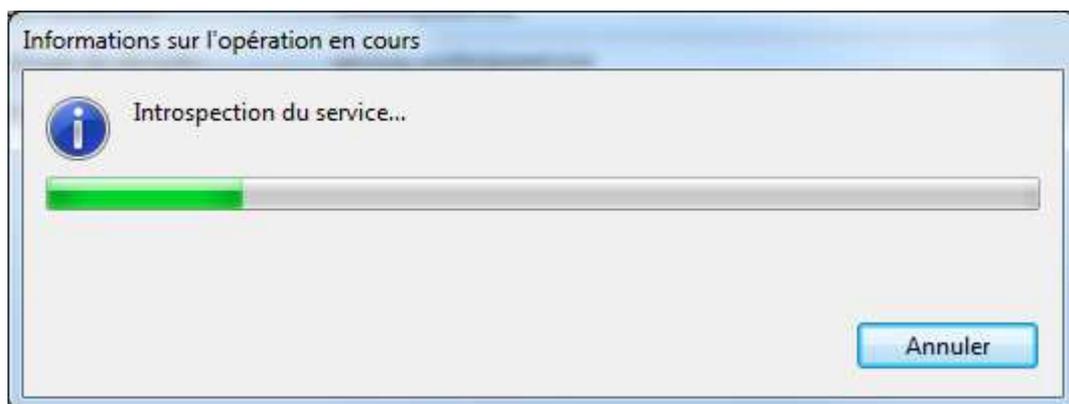


Saisir comme pour le client ASP.NET l'adresse WSDL de WebApplication1\_1.

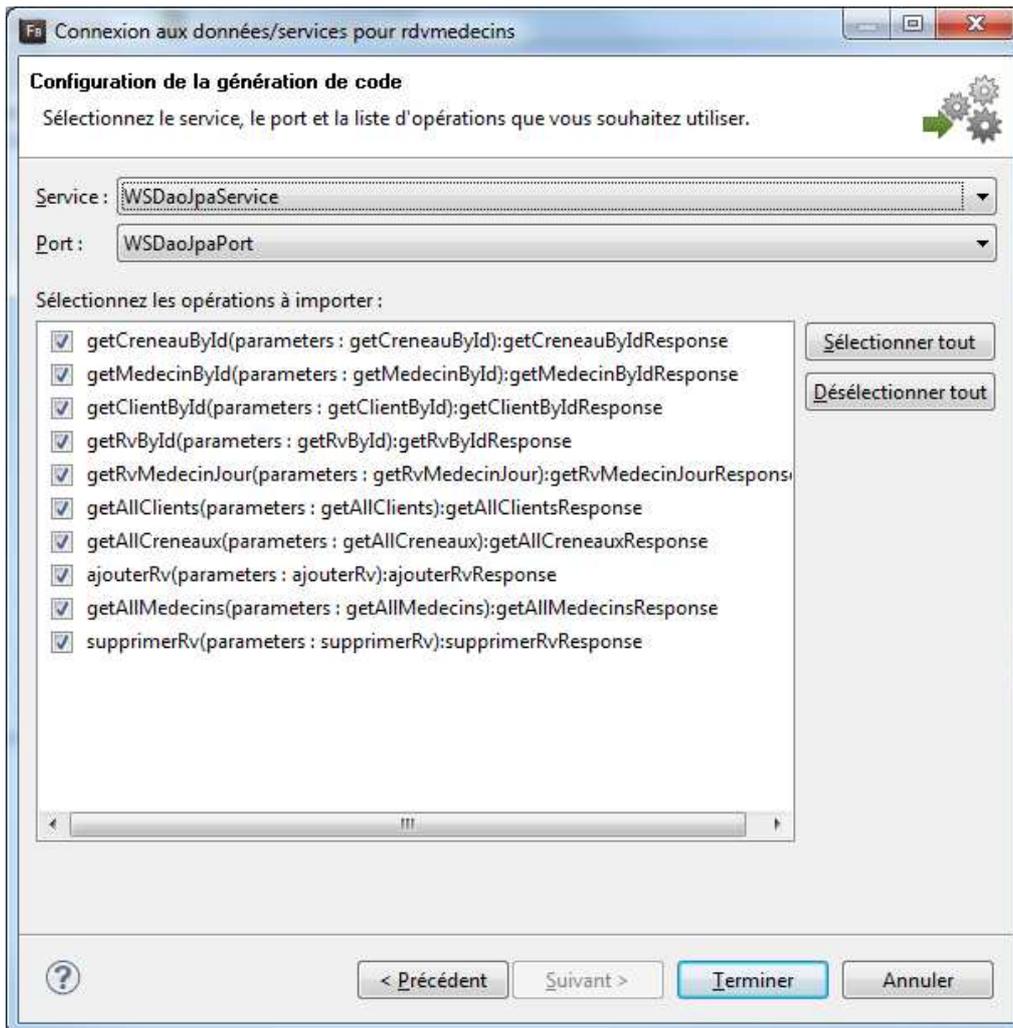
**NB : Penser à bien déployé l'application WebApplication1\_1 à partir de NetBeans.**



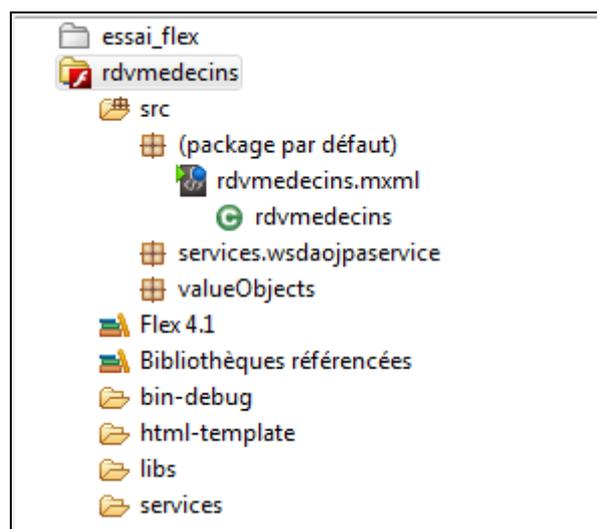
Après quelques secondes ...



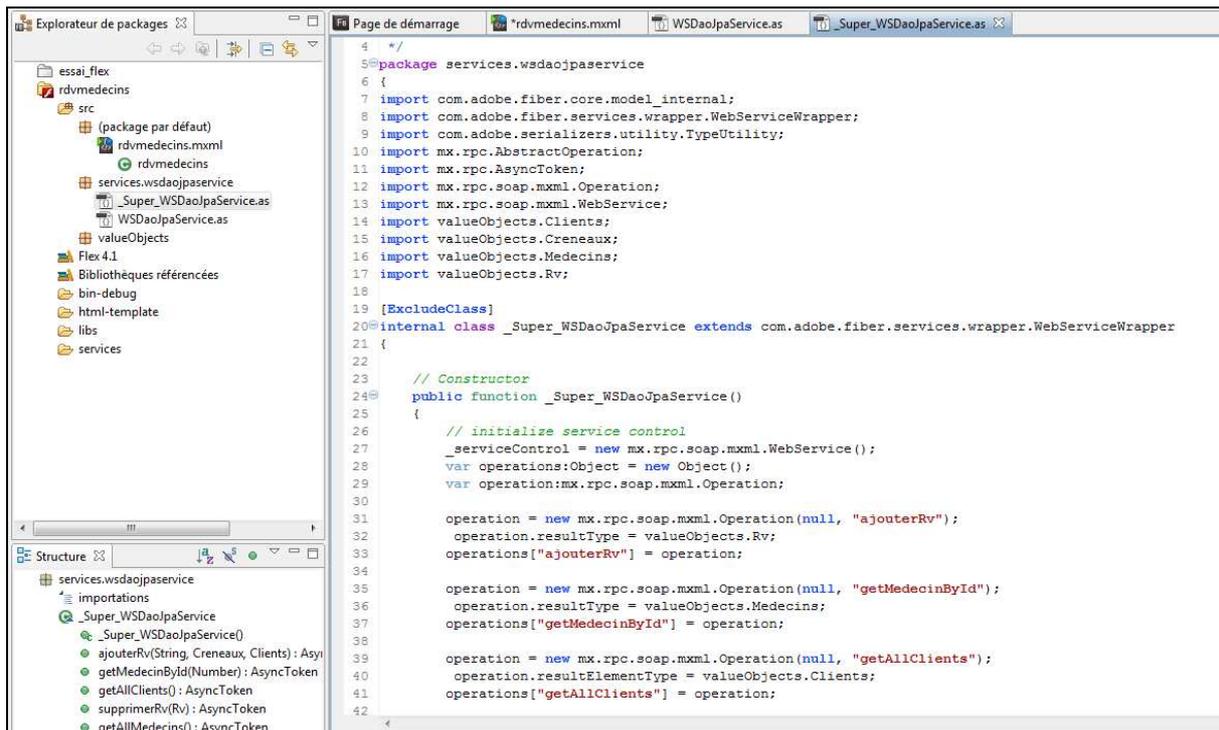
On obtient :



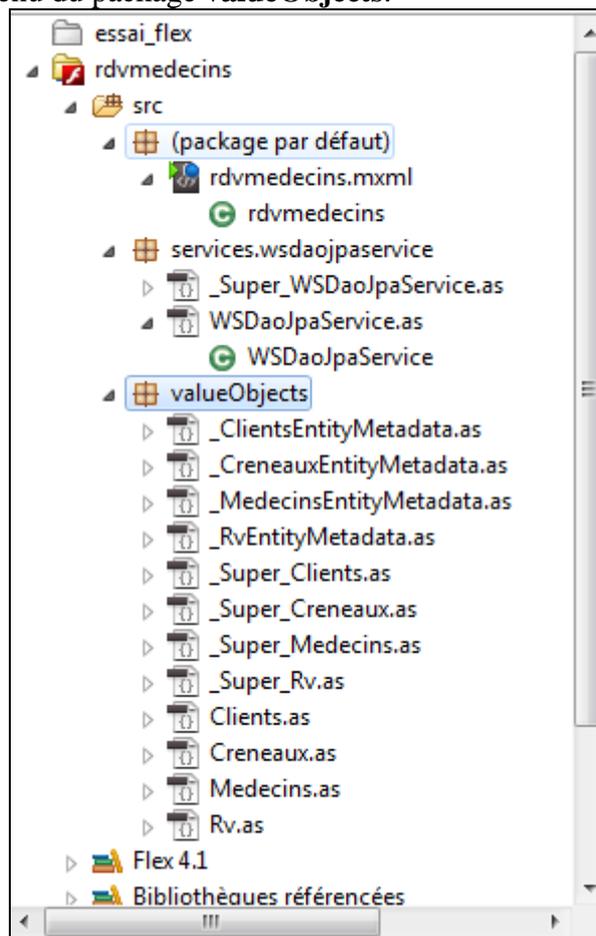
Le projet Flex se présente alors comme suit :



Remarquez la classe **Super\_WSDaoJpaService**



Examinez aussi le contenu du package **valueObjects**.



Etape 4.4. Connexion aux web services

Comme en ASP.NET on peut déclarer une variable ws de type WSDaoJpaService.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
  xmlns:s="library://ns.adobe.com/flex/spark"
  xmlns:mx="library://ns.adobe.com/flex/mx" minWidth="955" minHeight="600"
width="713" height="357">

  <fx:Script>
  <![CDATA[

import services.wsdajpaservice.WSDaoJpaService;

private var ws:WSDaoJpaService;

protected function button1_clickHandler(event:MouseEvent):void
{
    // TODO Auto-generated method stub
}

]]>
</fx:Script>

<fx:Declarations>
  <!-- Placer ici les éléments non visuels (services et objets de
valeur, par exemple). -->
</fx:Declarations>
<s:Button x="506" y="282" label="Afficher" width="155" height="49"
  click="button1_clickHandler(event)"/>
<s:TextArea x="47" y="32" width="614" height="227"/>
</s:Application>
```

----- FIN -----