Introduction à Silverlight 2

par Benjamin Roux (Retour aux articles)

Date de publication : 27/01/2008

Dernière mise à jour :

Cet article est une introduction à Silverlight 2.

1 - Introduction 3
2 - Création du projet 4
3 - Première application 8
4 - Le positionnement 11
4-1 - Canvas 11
4-2 - StackPanel 12
4-3 - Grid 14
5 - Utiliser le réseau pour récupérer des infos et remplir un DataGrid 16
6 - Les Styles 23
7 - ListBox et DataBinding 24
8 - UserControls 28
9 - Les Templates 31
10 - De Silverlight à WPF 36
11 - L'application en ligne 39
12 - Conclusion 40
13 - Remerciements 41

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

1 - Introduction

Silverlight est l'une des nouvelles technologies nées de chez Microsoft.

Avec, il est possible de créer des applications web riches, des animations, des jeux?

Voici ce qu'il est, entre autres, capable de réaliser

http://www.tafiti.com/

Des exemples sont également disponibles sur le site officiel

http://www.silverlight.net

Silverlight se décline actuellement en 2 versions :

- ***1.0*** qui permet de faire des applications côté client, tout est en Javascript.
- ***2*** (anciennement 1.1) qui intègre une version du Framework .NET avec une CLR et qui permet donc d'exécuter du code C# ou VB.NET.

Dans cet article nous apprendrons donc comment utiliser la version 2.

Quelques outils sont requis pour développer en Silverlight 2.

Les outils requis :

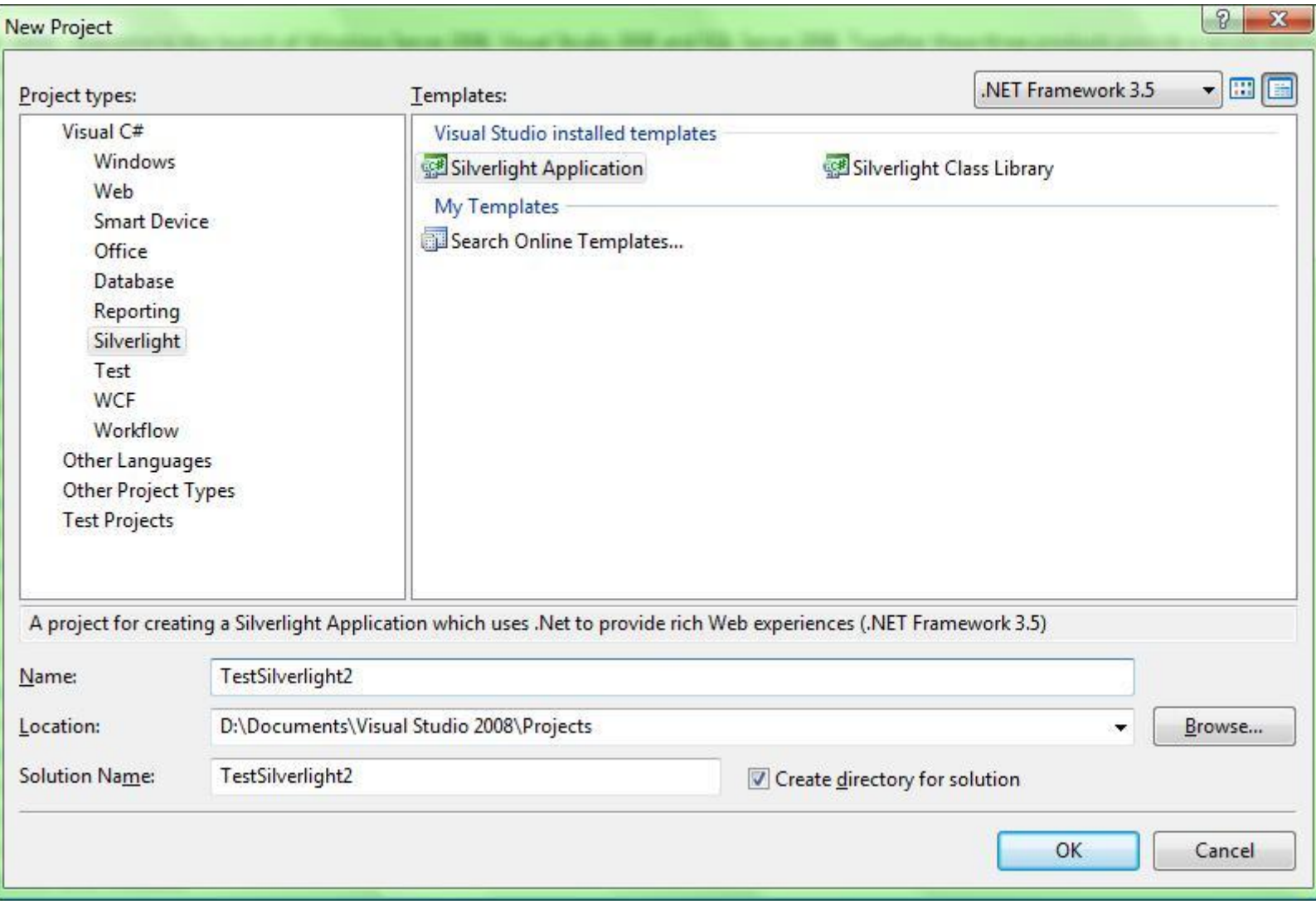
- Microsoft Visual Studio 2008 (je ne sais pas si une version Express convient)
- **Microsoft® Silverlight? 2 Software Development Kit Beta 1**
- **Microsoft Silverlight Tools Beta 1 for Visual Studio 2008**
- **Le runtime de Silverlight 2**

Une fois tous ces outils installés nous pouvons commencer.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

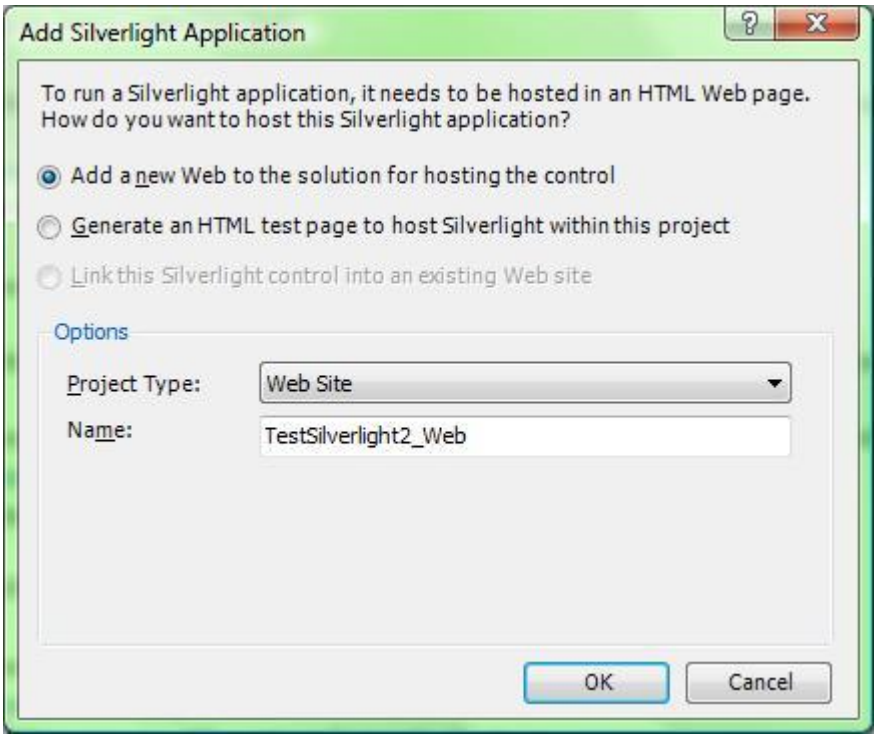
2 - Création du projet

Tout d'abord on lance **Visual Studio 2008** et on crée un nouveau projet de type **Silverlight Application**.

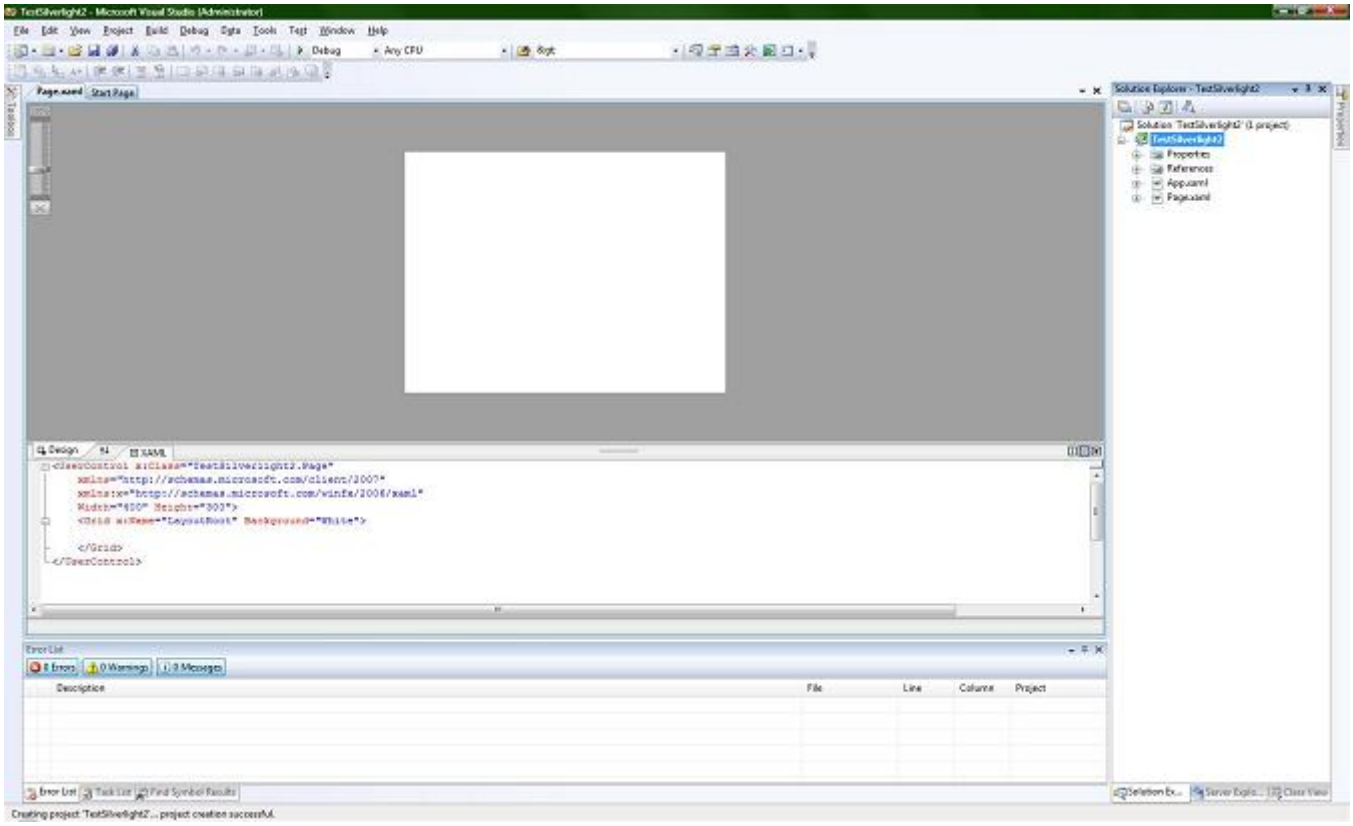


Une fois le nom du projet entré, on a le choix entre créer un nouveau projet de type **WebSite** directement dans notre solution, ou de simplement avoir une page HTML.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

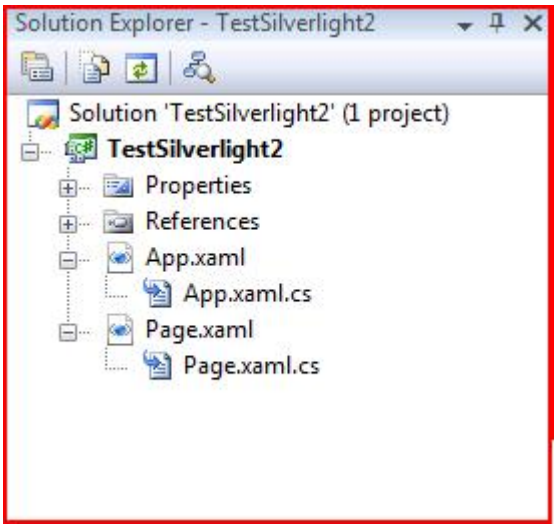


Pour notre article j'ai sélectionné la seconde option.
Nous voici donc avec notre projet fraichement créé.



On peut donc aperçevoir un mode splité pour notre XAML ce qui est assez agréable. Puis on peut voir les différents fichiers crées par défaut.

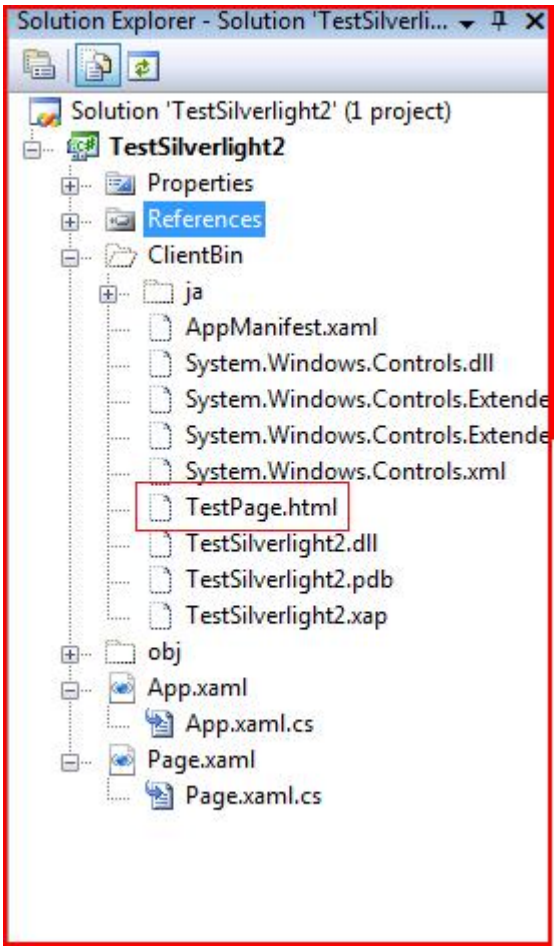
Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.



Fini donc les fichiers HTML ou JS, nous avons seulement 2 fichiers XAML (avec leur .cs correpondant pour le code-behind).

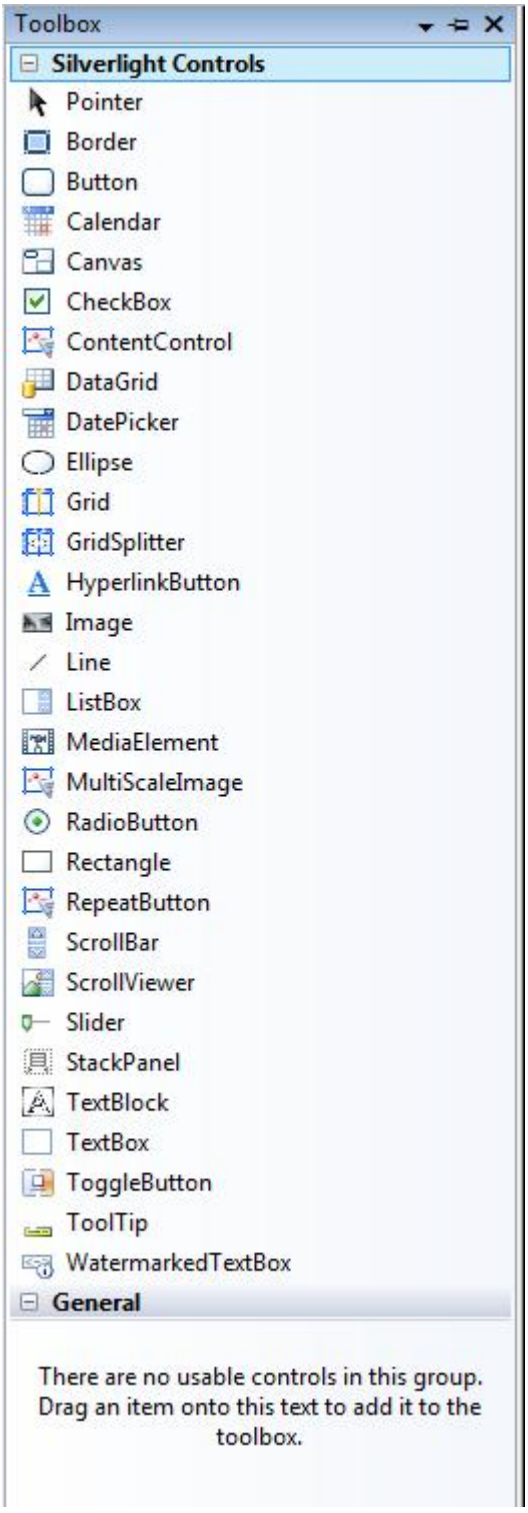
- Page.xaml : notre XAML principal, contient le code de notre application, le code-behind sert pour le code des évènements par exemple
- App.xaml : contient les Ressources de notre application (les styles par exemple), le code-behind lui sert à manipuler les évènements au niveau de l'application (Application_Startup, Application_Exit...).

Là on peut se demander où est donc notre fameux fichier HTML, c'est simple il se trouve dans le répertoire **ClientBin**.



Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

Un petit coup d'oeil à la ToolBox où on peut voir toute une tripotée de contrôle qui vont donc nous changer la vie par rapport à la version 1.0 et 1.1 alpha. Bref que du bon.

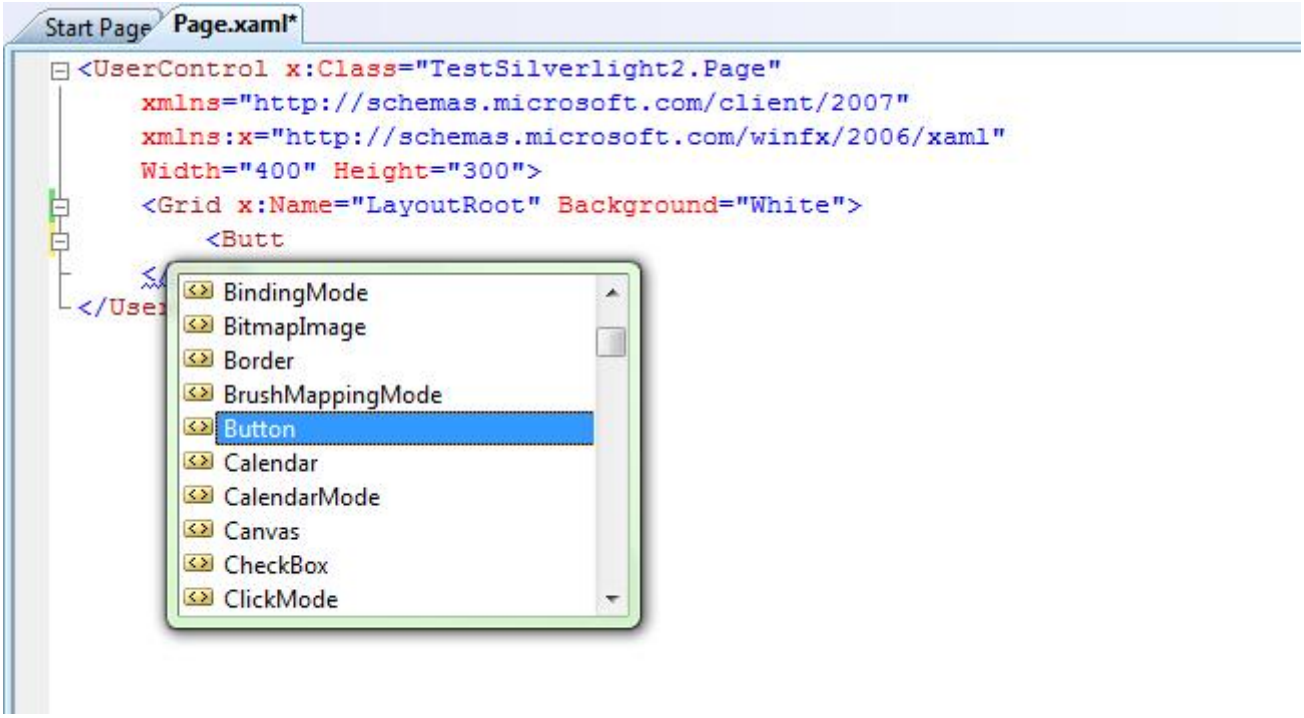


Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.
http://broux.developpez.com/articles/csharp/introduction-silverlight-2

3 - Première application

Une fois notre projet crée, on va créer notre première application.

On va donc simplement rajouter un bouton qui change de texte lorsque l'on clique dessus.



Remarquons l'Intellisense.

Ici on peut voir la création automatique d'**EventHandler** (qui n'était pas présente dans la version alpha). Si on clique dessus notre **EventHandler** est créé dans notre **Page.xaml.cs**.

Voici donc le code XAML final.

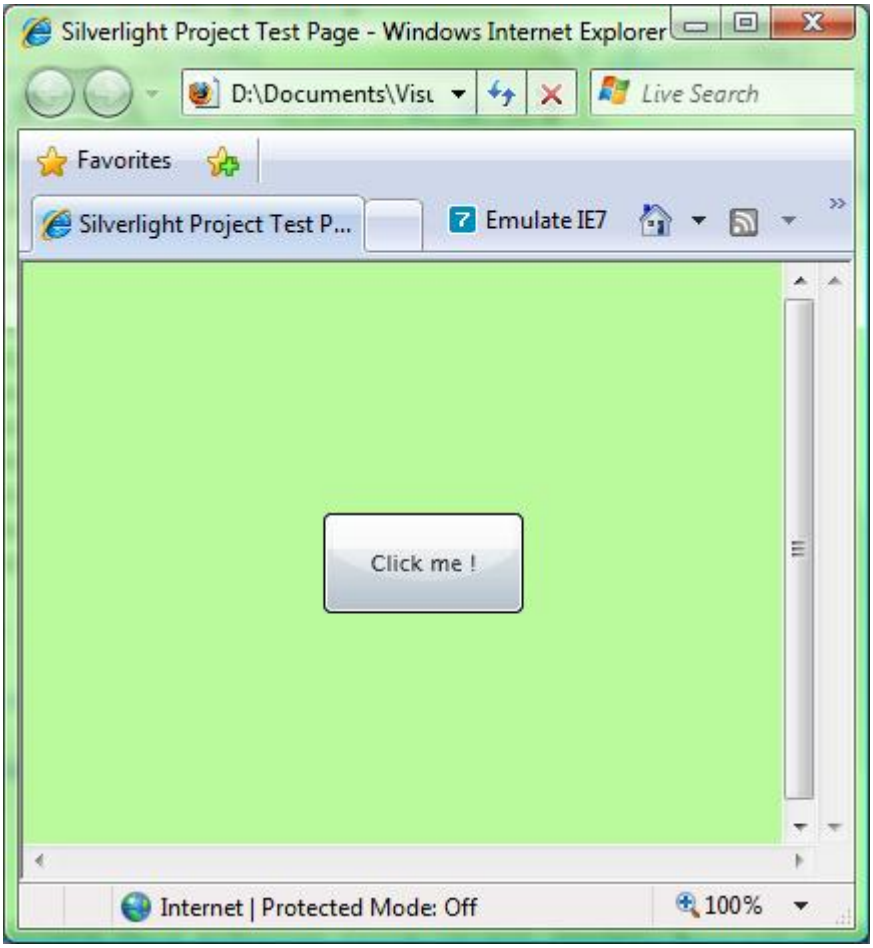
Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

```
<UserControl x:Class="TestSilverlight2.Page"
    xmlns="http://schemas.microsoft.com/client/2007"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Width="400" Height="300">
    <Grid x:Name="LayoutRoot" Background="Green">

  <Button x:Name="myButton" Content="Click me !" Click="myButton_Click" Width="100" Height="50"></Button>
    </Grid>
</UserControl>
```

Un petit tour dans le **Page.xaml.cs** pour rajouter notre code-behind.

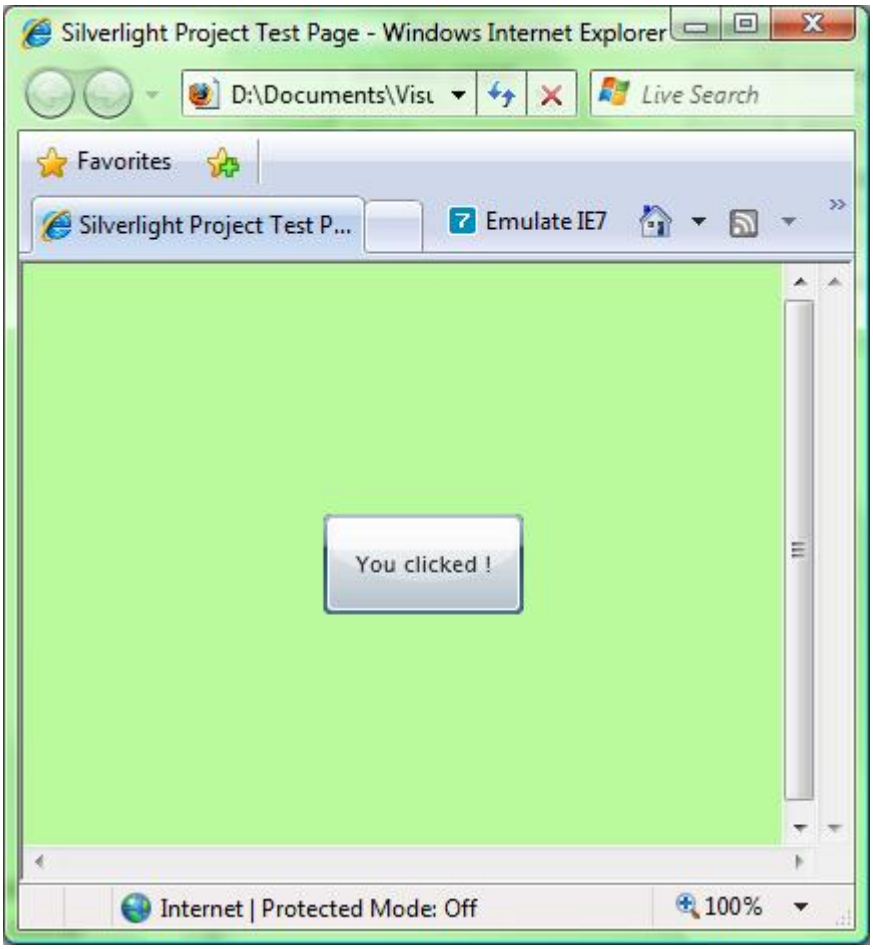
```
private void myButton_Click(object sender, RoutedEventArgs e)
{
    myButton.Content = "You clicked !";
}
```

Voici le résultat visuel.



Une fois qu'on a cliqué dessus.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.



Et voici donc notre première application en Silverlight 2.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

4 - Le positionnement

Nous voici donc dans la seconde partie sur le positionnement.

Silverlight 2 inclut 3 des systèmes de positionnement de WPF.

- Canvas
- StackPanel
- Grid

4-1 - Canvas

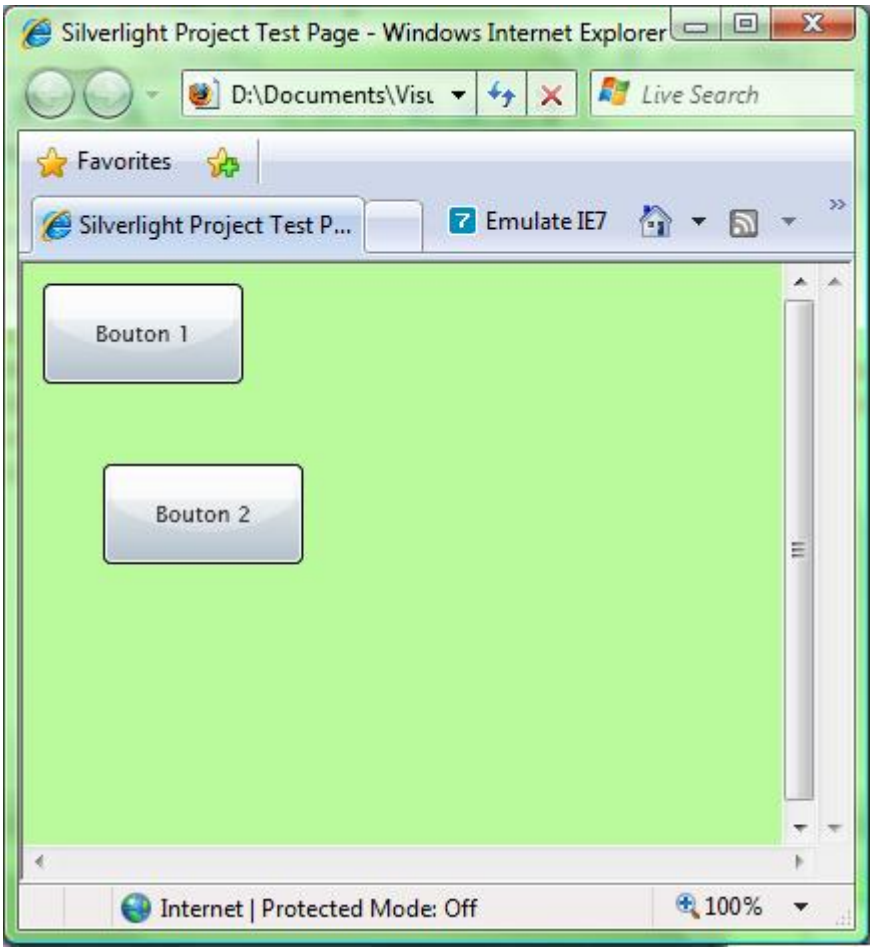
Le **Canvas** était le système de positionnement utilisé dans Silverlight 1.0 et 1.1.

Tous les éléments dans un **Canvas** sont positionnés par 2 propriétés : **Canvas.Left** et **Canvas.Top**.

```
<UserControl x:Class="TestSilverlight2.Page"
    xmlns="http://schemas.microsoft.com/client/2007"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Width="400" Height="300">
    <Canvas x:Name="LayoutRoot" Background="Green">
        <Button Content="Bouton 1" Width="100" Height="50" Canvas.Left="10" Canvas.Top="10"/>
        <Button Content="Bouton 2" Width="100" Height="50" Canvas.Left="40" Canvas.Top="100"/>
    </Canvas>
</UserControl>
```

Le résultat.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.



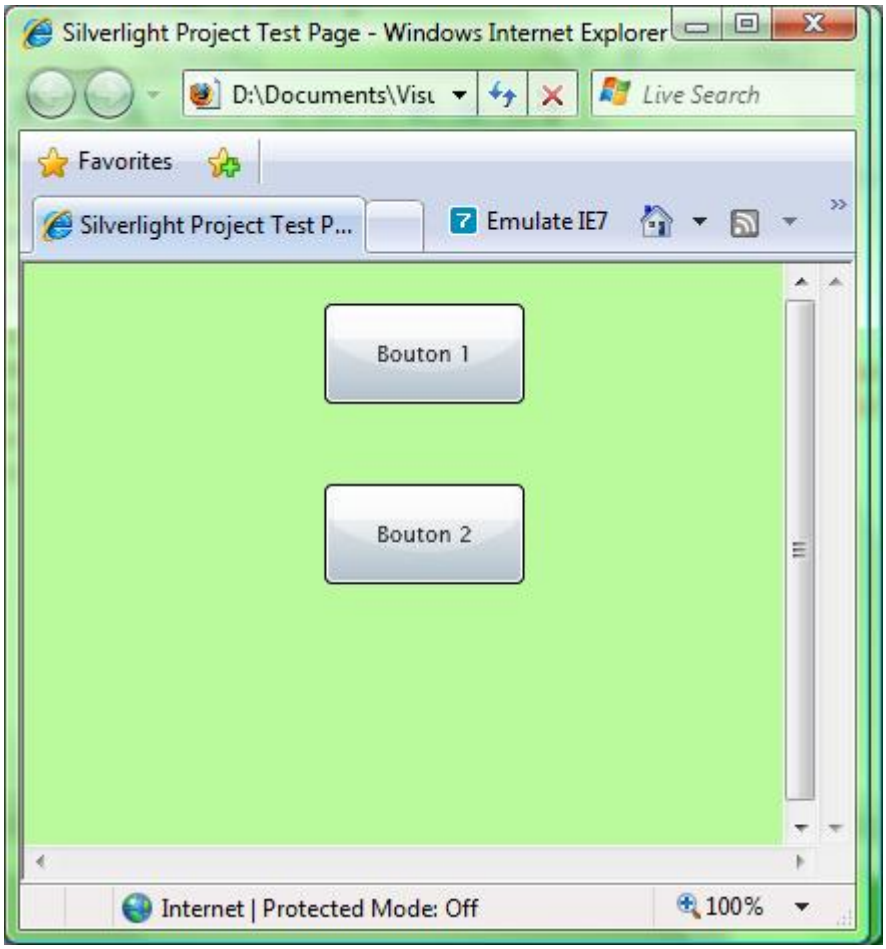
4-2 - StackPanel

Le **StackPanel** permet de positionner les éléments les uns en dessous des autres (ou les uns à côté des autres).

```
<UserControl x:Class="TestSilverlight2.Page"
    xmlns="http://schemas.microsoft.com/client/2007"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Width="400" Height="300">
    <StackPanel x:Name="LayoutRoot" Background="Green">
        <Button Content="Bouton 1" Width="100" Height="50" Margin="20"/>
        <Button Content="Bouton 2" Width="100" Height="50" Margin="20"/>
    </StackPanel>
</UserControl>
```

Le résultat.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

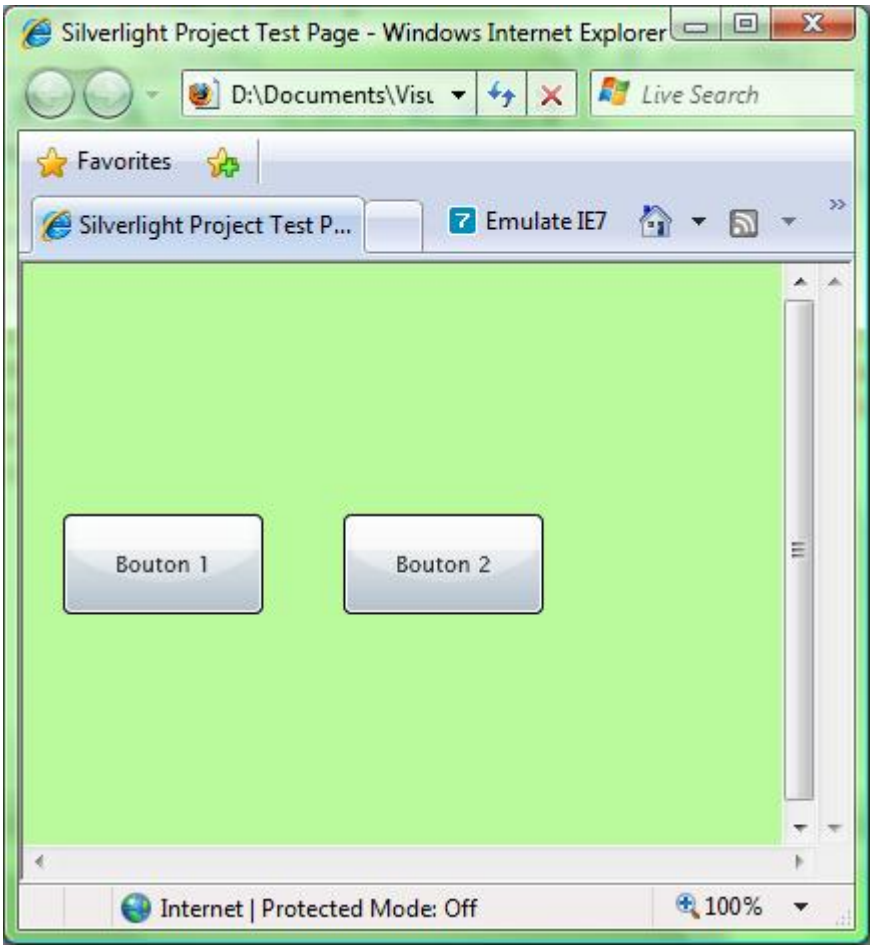


Comme vous pouvez le voir dans le code, l'écart entre les éléments est spécifié par la propriété **Margin**.

On peut également changer l'orientation via le propriété **Orientation**.

```
<UserControl x:Class="TestSilverlight2.Page"
    xmlns="http://schemas.microsoft.com/client/2007"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Width="400" Height="300">
    <StackPanel x:Name="LayoutRoot" Background="Green" Orientation="Horizontal">
        <Button Content="Bouton 1" Width="100" Height="50" Margin="20"/>
        <Button Content="Bouton 2" Width="100" Height="50" Margin="20"/>
    </StackPanel>
</UserControl>
```


Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.



4-3 - Grid

Et voici le positionnement de type **Grid**.

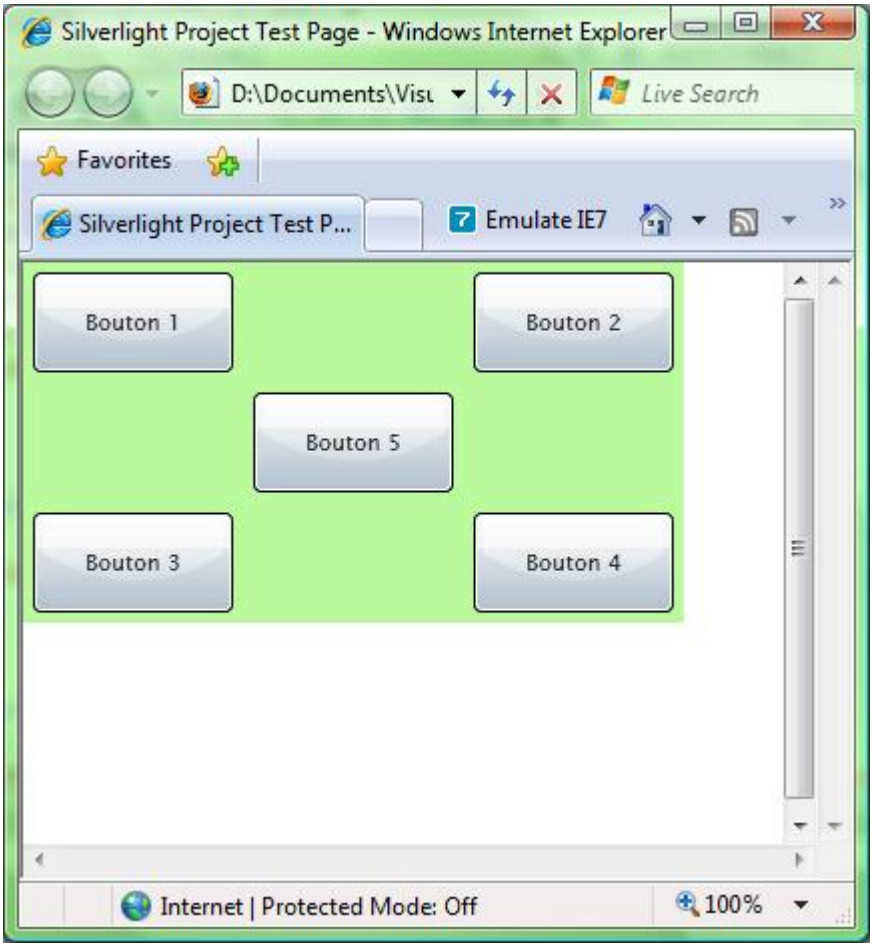
Le **Grid** et le système de positionnement le plus flexible il est basé sur des lignes et des colonnes, il est conceptuellement proche du tableau HTML.

Il suffit juste de spécifier au contrôle sa colonne ainsi que sa ligne via les propriétés **Grid.Column** et **Grid.Row**.

```
<UserControl x:Class="TestSilverlight2.Page"
    xmlns="http://schemas.microsoft.com/client/2007"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Width="330" Height="180">
    <Grid x:Name="LayoutRoot" Background="Green">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="110"/>
            <ColumnDefinition Width="110"/>
            <ColumnDefinition Width="110"/>
        </Grid.ColumnDefinitions>

        <Grid.RowDefinitions>
            <RowDefinition Height="60"/>
            <RowDefinition Height="60"/>
            <RowDefinition Height="60"/>
        </Grid.RowDefinitions>

        <Button Content="Bouton 1" Width="100" Height="50" Grid.Column="0" Grid.Row="0"/>
        <Button Content="Bouton 2" Width="100" Height="50" Grid.Column="2" Grid.Row="0"/>
        <Button Content="Bouton 3" Width="100" Height="50" Grid.Column="0" Grid.Row="2"/>
```


Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

```
        <Button Content="Bouton 4" Width="100" Height="50" Grid.Column="2" Grid.Row="2"/>
        <Button Content="Bouton 5" Width="100" Height="50" Grid.Column="1" Grid.Row="1"/>
    </Grid>
</UserControl>
```



Tout comme les tableaux en HTML, on peut fusionner des colonnes et lignes (pour mettre un contrôle sur 2 lignes par exemples) avec les propriétés **Grid.RowSpan** et **Grid.ColumnSpan**.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

5 - Utiliser le réseau pour récupérer des infos et remplir un DataGrid

Dans cette partie vous allons voir comment récupérer des informations via le réseau et comment remplir un **DataGrid**.

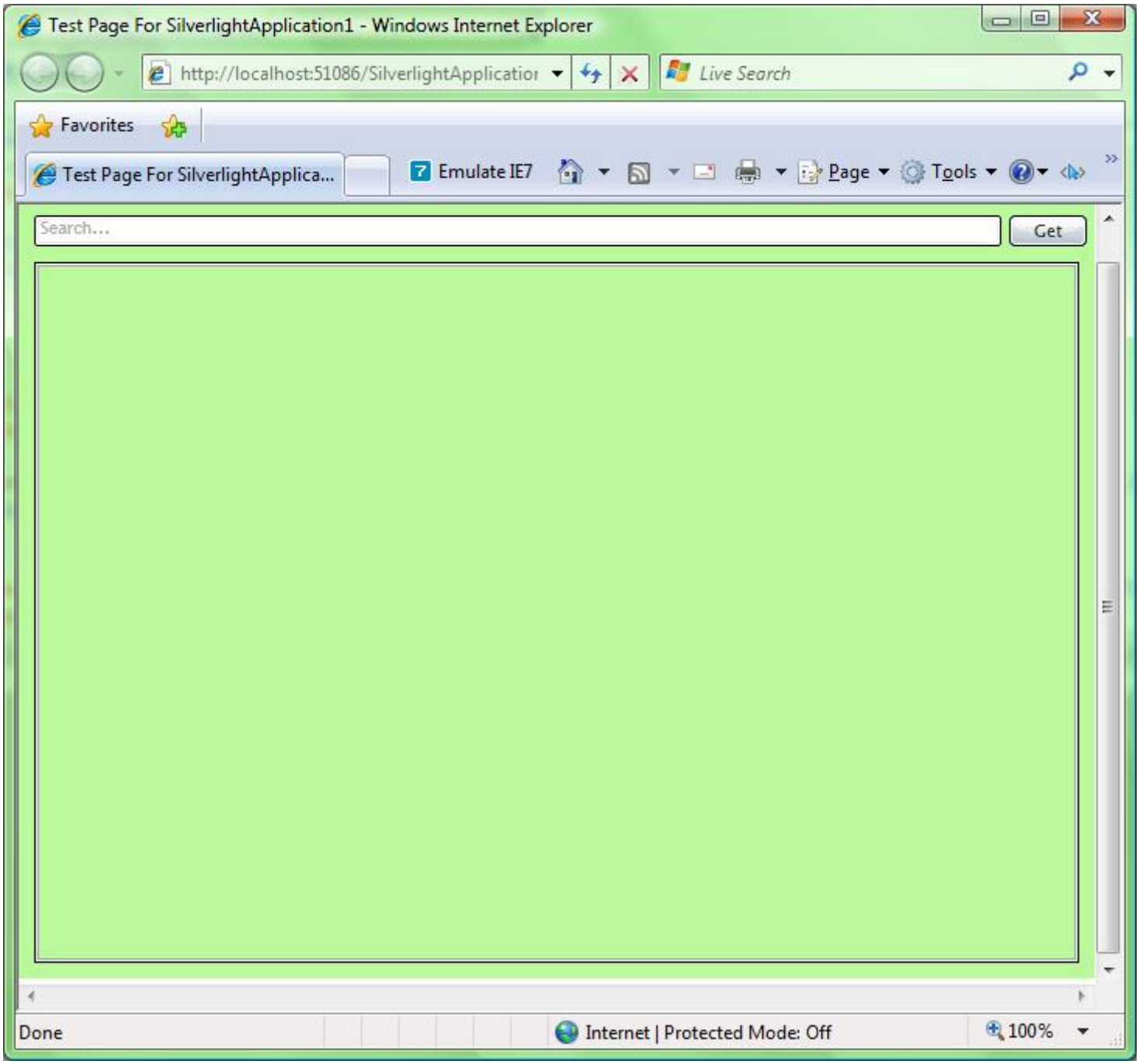
Tout d'abord nous allons créer un nouveau projet mais cette fois nous allons aussi créer un projet de type Web Site.

Une fois ceci fait nous allons simplement ajouter une **TextBox**, un **Button** et un **GridView** dans notre XAML.

On place le tout dans un **Grid**.

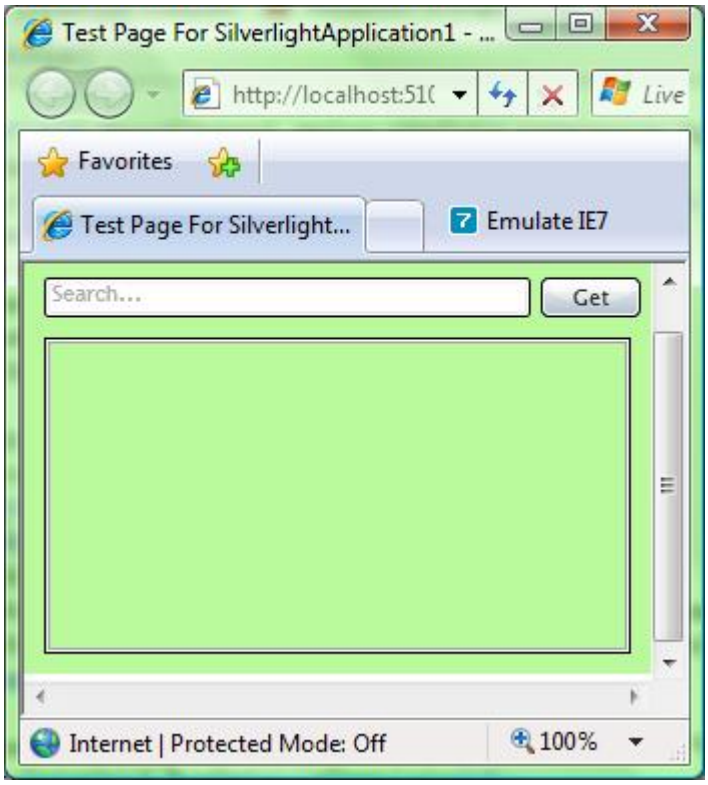
```
<UserControl xmlns:my="clr-namespace:System.Windows.Controls;assembly=System.Windows.Controls.Data"
    x:Class="FlickrPhotoViewer.Page"
    xmlns="http://schemas.microsoft.com/client/2007"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:Flickr="clr-namespace:FlickrPhotoViewer;assembly=FlickrPhotoViewer">
    <Grid x:Name="LayoutRoot" Background="#BBF99D">
        <Grid.RowDefinitions>
            <RowDefinition Height="40"/>
            <RowDefinition Height="*"/>
        </Grid.RowDefinitions>

        <Grid Grid.Row="0">
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="*"/>
                <ColumnDefinition Width="60"/>
            </Grid.ColumnDefinitions>
            <TextBox x:Name="textSearch" Grid.Column="0"
                            VerticalAlignment="Center" Margin="10,10,0,10"/>

  <Button x:Name="buttonSearch" Click="buttonSearch_Click" Width="50" Height="20" Content="Get" Grid.Column="1"/>
        </Grid>


  <my:DataGrid x:Name="Photos" AutoGenerateColumns="True" Grid.Row="1" Margin="10,0,10,10"></my:DataGrid>
    </Grid>
</UserControl>
```


Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.



Le fait d'avoir enlevé les propriétés **Width** et **Height** dans la balise **UserControl** permet à notre contrôle Silverlight d'être redimensionné en même temps que la fenêtre.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.



Ensuite nous allons aller dans le code-behind pour mettre le code du clic sur le bouton.

Pour cet exemple nous allons récupérer des données via l'API de Flickr. Je vais utiliser leur méthode **flickr.photos.search**. Pour cela nous allons utiliser une nouveauté de Silverlight 2, c'est le namespace **System.Net**, qui contient la classe **WebClient**.

*Vous pouvez voir le XML généré par cette méthode **ici**.*

```
private void buttonSearch_Click(object sender, RoutedEventArgs e)
{
    string url = string.Format("http://api.flickr.com/services/rest/?
method=flickr.photos.search&amp;" +

  "api_key=1cb12f30d9a23d70f00e4a238b4cf1fd&amp;text={0}&amp;machine_tag_mode=&amp;per_page=20",
                              textSearch.Text);
    WebClient webClient = new WebClient();

    webClient.DownloadStringCompleted += new
 DownloadStringCompletedEventHandler(webClient_DownloadStringCompleted);
    webClient.DownloadStringAsync(new Uri(url));
}

void webClient_DownloadStringCompleted(object sender, DownloadStringCompletedEventArgs e)
{
    if (e.Error == null)
    {
        // TODO
    }
}
```

Maintenant nous allons créer une classe qui représentera une image sur Flickr. On va donc ajouter une nouvelle classe à notre projet et mettre ce code.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

```
public class FlickrPhoto
{
    public long Id { get; set; }
    public string Owner { get; set; }
    public string Title { get; set; }
    public string Image { get; set; }
}
```

Nous allons donc stocker l'id, le propriétaire, le titre et l'adresse de l'image. Pour récupérer l'adresse de l'image nous allons utiliser les infos retournées dans le XML à savoir le farm-id, le server-id, l'id et le secret. En effet les photos peuvent être retrouvées via cette URL : http://farm{farm-id}.static.flickr.com/{server-id}/{id}_{secret}.jpg

Maintenant nous allons rajouter le code pour exploiter le XML, pour cela nous allons utiliser LINQ donc il faut avant tout ajouter la référence **System.Xml.Linq** à notre projet.

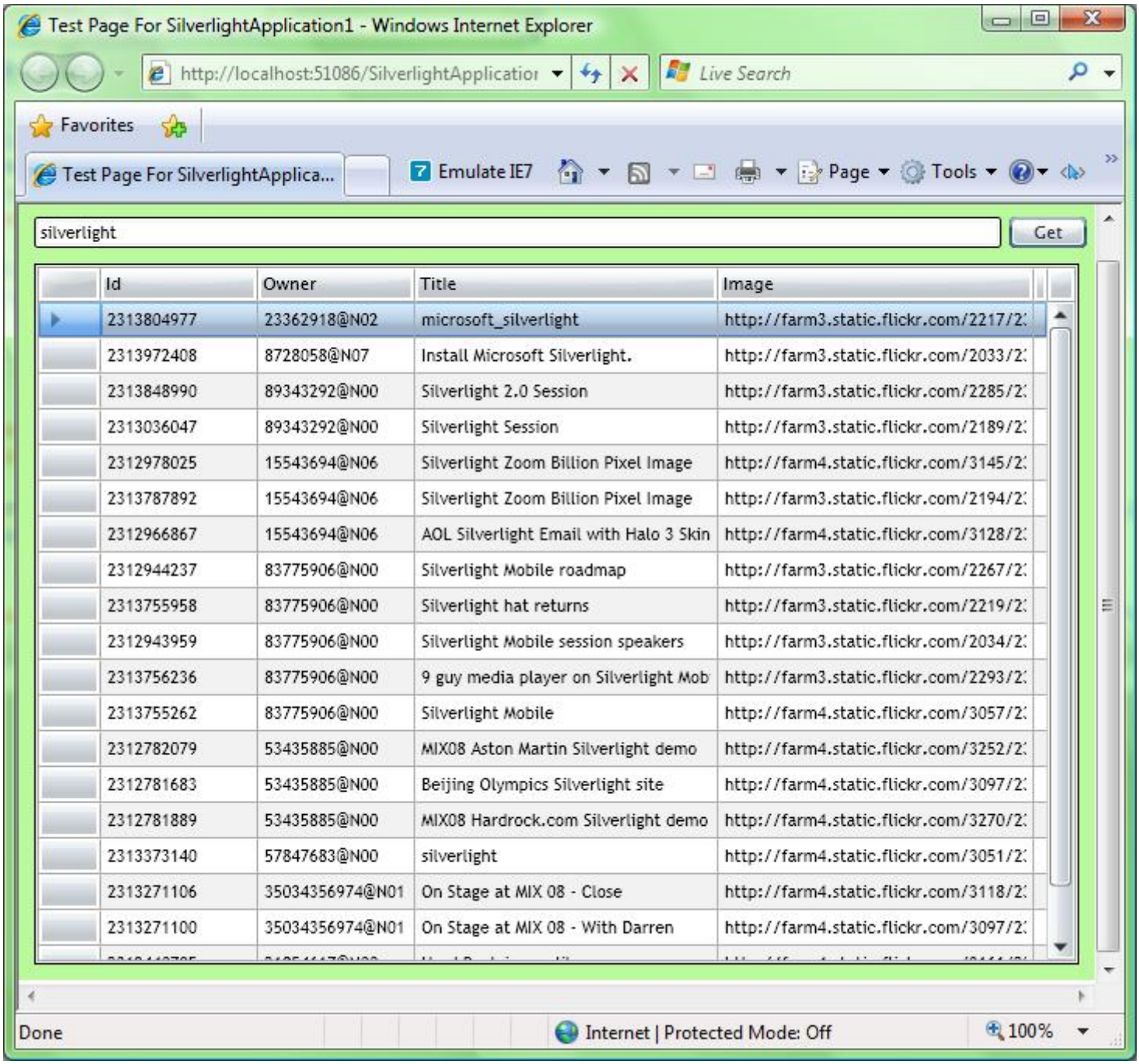
```
private void DisplayFlickrPhotos(string xml)
{
    XDocument doc = XDocument.Parse(xml);

    var photos = from photo in doc.Descendants("photo")
                 select new FlickrPhoto
                 {
                     Id = (long)photo.Attribute("id"),
                     Owner = (string)photo.Attribute("owner"),
                     Title = (string)photo.Attribute("title"),
                     Image = string.Format("http://farm{0}.static.flickr.com/{1}/{2}_{3}.jpg",
                                           (int)photo.Attribute("farm"),
                                           (int)photo.Attribute("server"),
                                           (long)photo.Attribute("id"),
                                           (string)photo.Attribute("secret"))
                 };

    Photos.ItemsSource = photos;
}
```

On n'oublie pas d'ajouter l'appel à cette méthode à la place du *TODO*.

```
void webClient_DownloadStringCompleted(object sender, DownloadStringCompletedEventArgs e)
{
    if (e.Error == null)
    {
        DisplayFlickrPhotos(e.Result);
    }
}
```

Et voici le résultat une fois que l'on a cliqué sur le bouton.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.



Voilà nous venons de voir qu'il était très simple de faire désormais du **Cross Domain**, et de jouer avec le XML avec **Linq**.

Nous allons maintenant améliorer notre **GridView**, en affichant la photo lorsque l'on clique sur une ligne.

Pour cela nous allons rajouter un **Template** lorsqu'une ligne est sélectionnée.

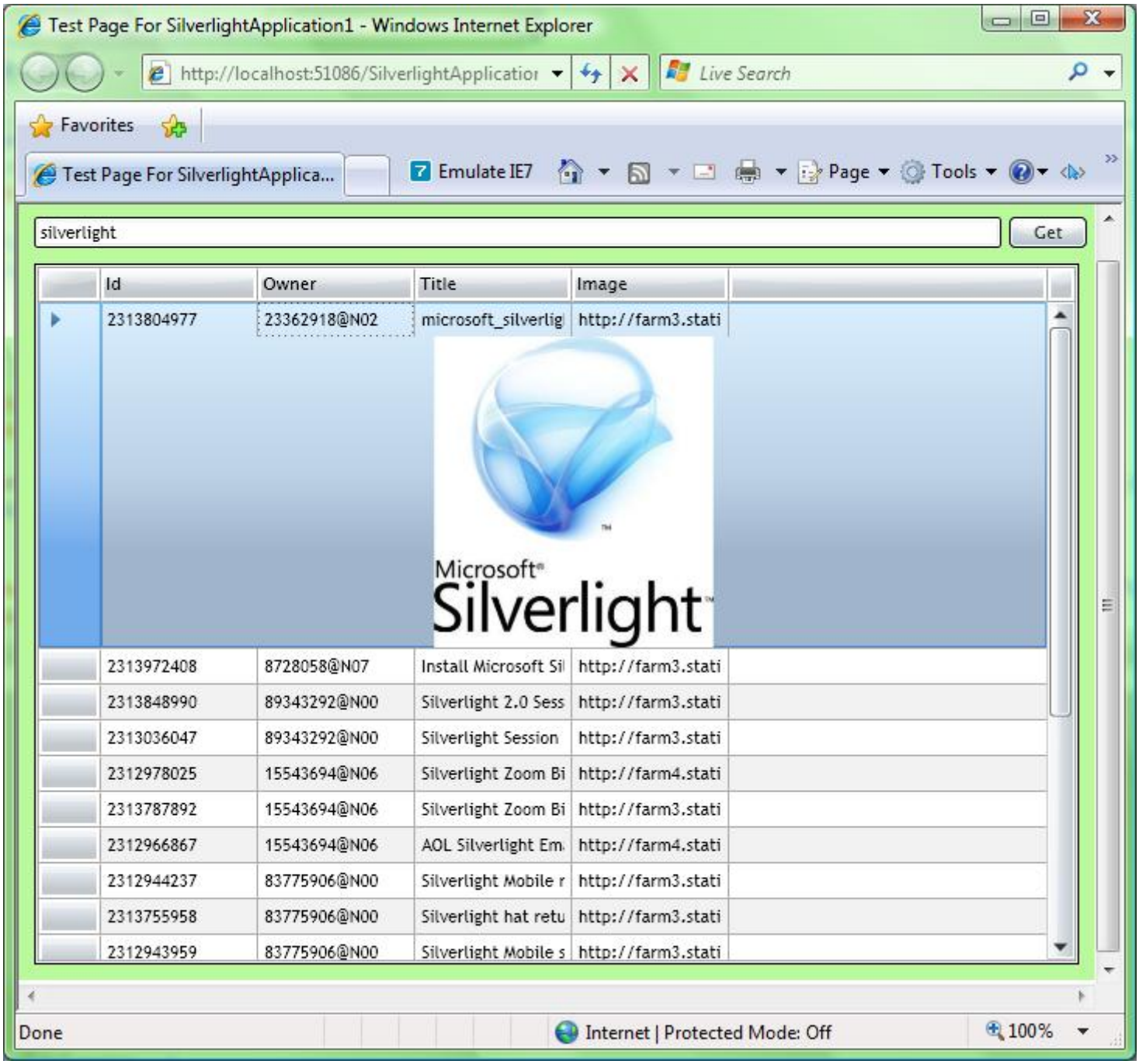
```
<UserControl xmlns:my="clr-namespace:System.Windows.Controls;assembly=System.Windows.Controls.Data"
    x:Class="FlickrPhotoViewer.Page"
    xmlns="http://schemas.microsoft.com/client/2007"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:Flickr="clr-namespace:FlickrPhotoViewer;assembly=FlickrPhotoViewer">
        <Grid.RowDefinitions>
            <RowDefinition Height="40"/>
```


Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

```
            <RowDefinition Height="*"/>
        </Grid.RowDefinitions>

        <Grid Grid.Row="0">
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="*"/>
                <ColumnDefinition Width="60"/>
            </Grid.ColumnDefinitions>
            <TextBox x:Name="textSearch" Grid.Column="0"
                                VerticalAlignment="Center" Margin="10,10,0,10"/>

   <Button x:Name="buttonSearch" Click="buttonSearch_Click" Width="50" Height="20" Content="Get" Grid.Column="1"/>
        </Grid>


   <my:DataGrid x:Name="Photos" AutoGenerateColumns="True" RowDetailsVisibilityMode="VisibleWhenSelected"
                     Grid.Row="1" Margin="10,0,10,10">
            <my:DataGrid.RowDetailsTemplate>
                <DataTemplate>
                    <Grid>
                        <Image Source="{Binding Image}" Width="200" Height="200" />
                    </Grid>
                </DataTemplate>
            </my:DataGrid.RowDetailsTemplate>
        </my:DataGrid>
    </Grid>
</UserControl>
```

Désormais la photo sera affichée quand on cliquera sur une ligne.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.



Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

6 - Les Styles

Les **Styles** en Silverlight (et WPF) permettent d'appliquer un style à un ensemble de contrôle à l'instar des fichiers .skin en ASP.NET ou encore des fichiers CSS.

Leur déclaration se fait dans le fichier **App.xaml**.

```
<Application.Resources>
    <Style x:Key="ButtonStyle" TargetType="Button">
        <Setter Property="Width" Value="50"/>
        <Setter Property="Height" Value="20"/>
    </Style>
</Application.Resources>
```

Ici tous les boutons auxquels j'applique ce style auront une largeur de 50 et une hauteur de 20.

Pour appliquer un style c'est extrêment simple. Il suffit de le spécifier dans la propriété **Style** des contrôles.

```
<Button x:Name="buttonSearch" Click="buttonSearch_Click" Content="Get" Style="{StaticResource ButtonStyle}"/>
```


Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

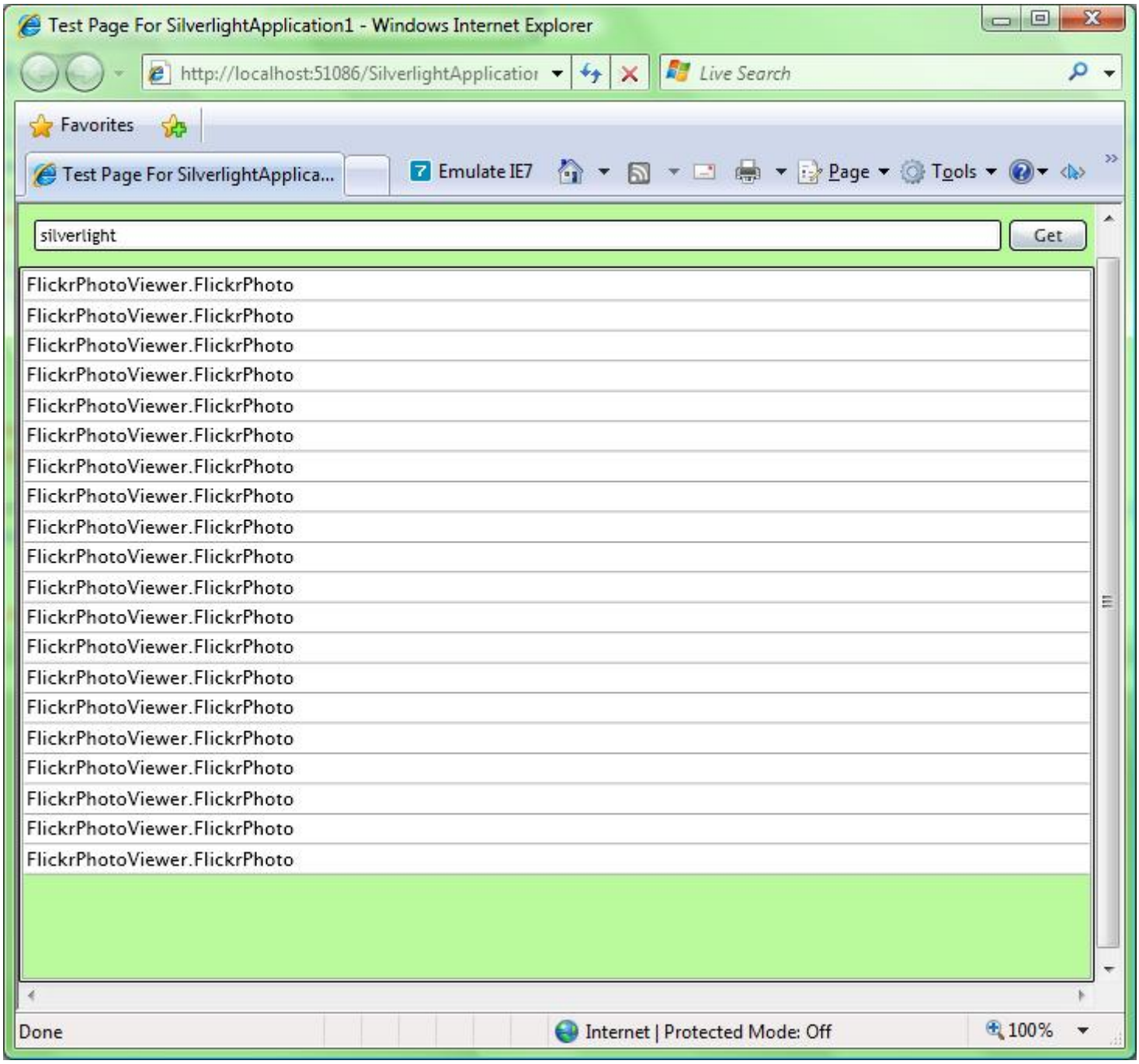
7 - ListBox et DataBinding

Dans cette partie nous allons voir comment afficher des données dans une **ListBox**.

Tout d'abord nous allons supprimer notre **DataGrid** pour le remplacer par une **ListBox**.

```
<ListBox x:Name="Photos" Grid.Row="1"></ListBox>
```

On lance.

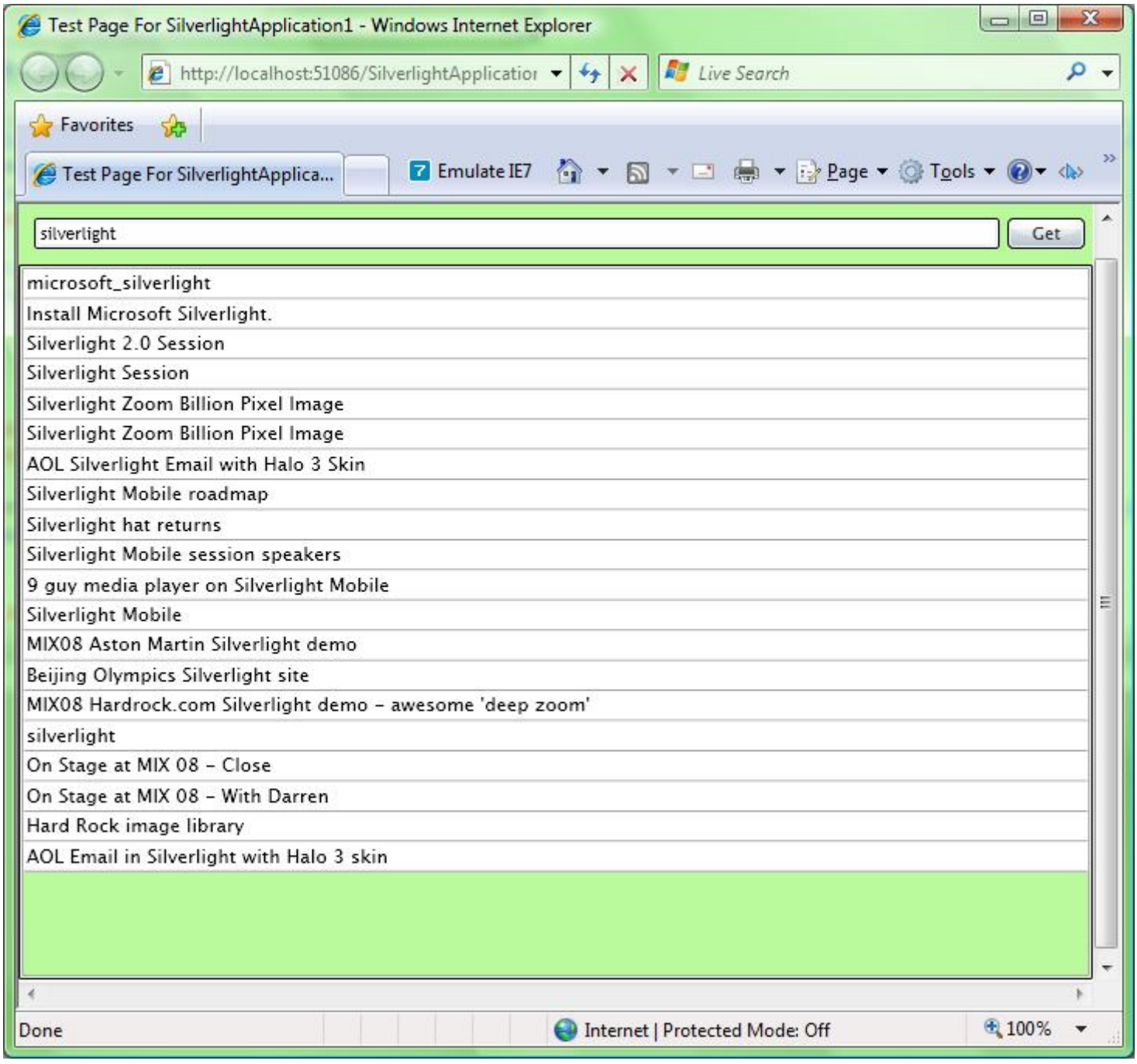


Seulement l'affichage n'est pas top, en effet c'est la méthode **ToString()** qui est appelée pour afficher les éléments et par défaut elle affiche le namespace et le nom de la classe. Nous allons donc rendre l'affiche plus joli.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

Pour cela nous allons utiliser la propriété **DisplayMemberPath** pour spécifier quelles propriétés de notre classe **FilckrPhoto** afficher.

```
<ListBox x:Name="Photos" DisplayMemberPath="Title">

</ListBox>
```



Si on veut afficher plusieurs éléments (par exemple l'id, le titre et même l'image) alors il faut modifier l'**ItemTemplate** de notre **ListBox**.

```
<ListBox x:Name="Photos" Grid.Row="1">
    <ListBox.ItemTemplate>
        <DataTemplate>
```


Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

```
            <StackPanel Orientation="Horizontal">
                <Image Source="{Binding Image}" Style="{StaticResource ThumbnailStyle}"/>
                <TextBlock Text="{Binding Title}" Style="{StaticResource TextBlockStyle}" />
            </StackPanel>
        </DataTemplate>
    </ListBox.ItemTemplate>
</ListBox>
```

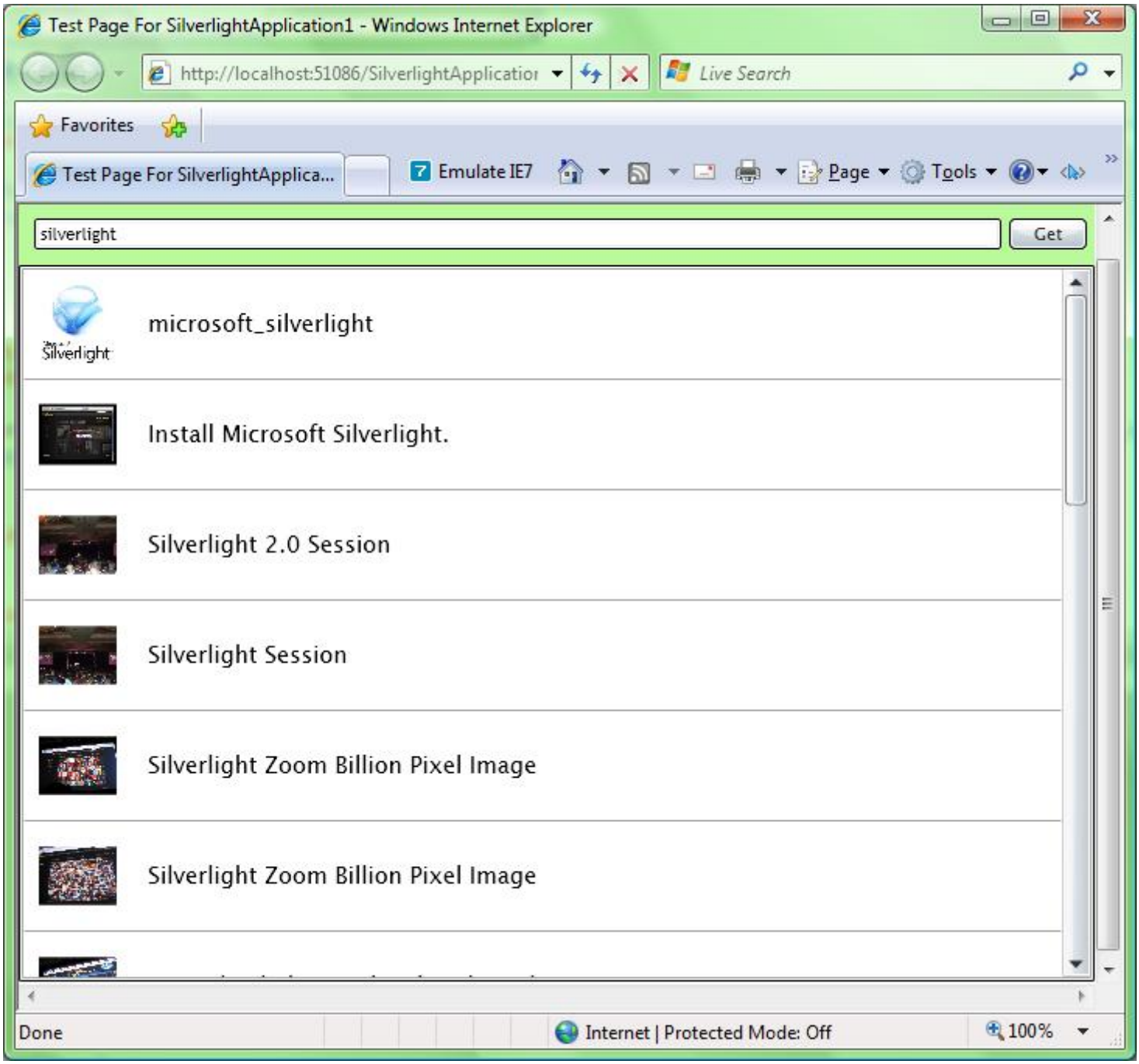
*On n'oublie alors pas d'enlever la propriété **DisplayMemberPath**.*

Avec les styles suivants.

```
<Style x:Key="ThumbnailStyle" TargetType="Image">
    <Setter Property="Margin" Value="10"/>
    <Setter Property="Height" Value="50"/>
    <Setter Property="Width" Value="50"/>
</Style>
<Style x:Key="TextBlockStyle" TargetType="TextBlock">
    <Setter Property="Margin" Value="10"/>
    <Setter Property="VerticalAlignment" Value="Center"/>
</Style>
```

Chaque item sera donc composé d'une miniature de l'image suivi du titre.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.



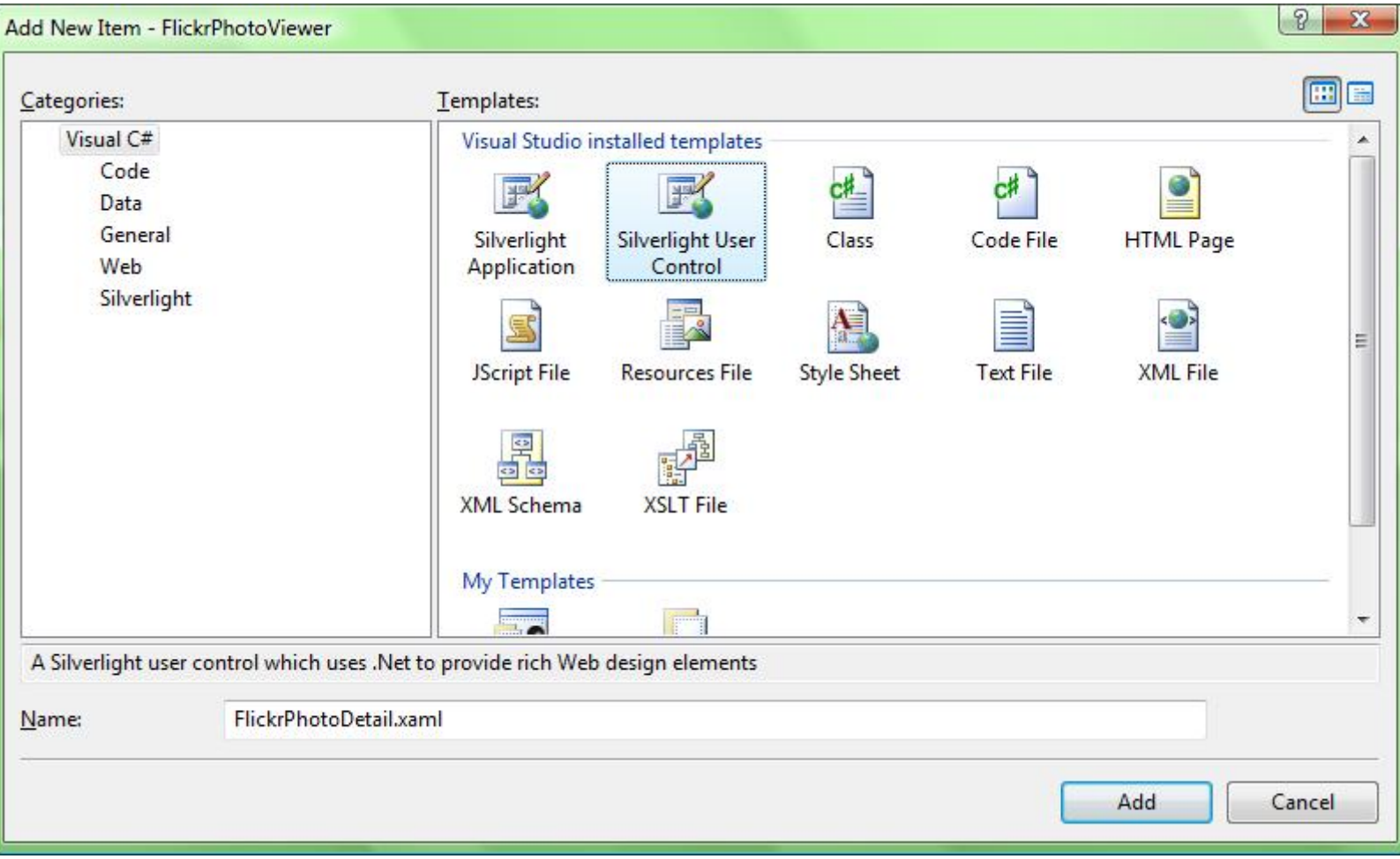
La **ListBox** supporte la séléction ainsi que le déplacement via les flèches du clavier.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

8 - UserControls

Dans cette partie nous allons voir comment créer et utiliser un **UserControl** en Silverlight. Nous allons donc reprendre l'application précédente et rajouter un panel contenant les informations d'une image lorsque l'on clique sur un item de la **ListBox**.

On va tout d'abord créer notre UserControl : Add -> New Item



Ajoutons lui quelques contrôles pour afficher le titre, l'image, le propriétaire et le chemin de l'image ainsi qu'un bouton pour cacher le contrôle.

```
<UserControl x:Class="FlickrPhotoViewer.FlickrPhotoDetail"
    xmlns="http://schemas.microsoft.com/client/2007"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
    <Grid x:Name="LayoutRoot" Background="White">
        <StackPanel HorizontalAlignment="Center">
            <Button x:Name="btnClose" Content="Fermer" Click="btnClose_Click"/>
            <TextBlock Text="{Binding Title}" Style="{StaticResource TitleStyle}"/>
            <Image Source="{Binding Image}" Style="{StaticResource ImageStyle}"/>
            <TextBlock Text="{Binding Owner}" Style="{StaticResource OwnerStyle}"/>
            <TextBox Text="{Binding Image}" IsReadOnly="True"/>
        </StackPanel>
    </Grid>
</UserControl>
```


Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

Voici le code du clic sur le bouton.

```
private void btnClose_Click(object sender, RoutedEventArgs e)
{
    this.Visibility = Visibility.Collapsed;
}
```

On cachera donc notre **UserControl** quand on cliquera sur le bouton.

Et voici maintenant les styles.

```
<Style x:Key="TitleStyle" TargetType="TextBlock">
    <Setter Property="FontSize" Value="30"/>
    <Setter Property="FontFamily" Value="Verdana"/>
    <Setter Property="Margin" Value="10"/>
</Style>
<Style x:Key="ImageStyle" TargetType="Image">
    <Setter Property="Margin" Value="10"/>
    <Setter Property="Height" Value="300"/>
    <Setter Property="Width" Value="300"/>
</Style>
<Style x:Key="OwnerStyle" TargetType="TextBlock">
    <Setter Property="FontSize" Value="12"/>
    <Setter Property="FontFamily" Value="Verdana"/>
    <Setter Property="Margin" Value="10"/>
</Style>
```

Alors comme vous pouvez voir nous utilisons du **DataBinding**, et vous vous demandez peut-être où est-ce qu'il va aller chercher ses infos. C'est simple il va aller les chercher dans sa propriété **DataContext** qui si elle n'est pas définie est celle du contrôle de niveau supérieur. On va donc devoir spécifier le **DataContext** de notre **UserControl** au moment de l'afficher.

Tout d'abord nous allons l'ajouter à notre application.

Pour cela il faut d'abord dire où se trouve notre contrôle et lui donner un namespace. On va donc rajouter cette ligne dans notre **Page.xaml** dans la balise **UserControl**.

```
xmlns:Flickr="clr-namespace:FlickrPhotoViewer;assembly=FlickrPhotoViewer"
```

On va donc rajouter notre **UserControl** à notre application, nous allons le placer tout à la fin et faire en sorte qu'il soit sur les 2 lignes de notre **Grid**.

```
<Flickr:FlickrPhotoDetail x:Name="PhotoDetail" Grid.RowSpan="2" Visibility="Collapsed"/>
```

On met sa visibilité à **Collapsed** (caché) par défaut.

Maintenant il ne nous reste plus qu'à l'afficher et définir son **DataContext** quand on clique sur un item de la **ListBox**.

On s'abonne donc à l'évènement **SelectionChanged**.

```
<ListBox x:Name="Photos" Grid.Row="1" DisplayMemberPath="Title" SelectionChanged="Photos_SelectionChanged">
```

Et voici le code-behind.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

```
private void Photos_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    FlickrPhoto photo = (FlickrPhoto)Photos.SelectedItem;

    PhotoDetail.DataContext = photo;
    PhotoDetail.Visibility = Visibility.Visible;
}
```

Voici le résultat quand on clique sur un item.

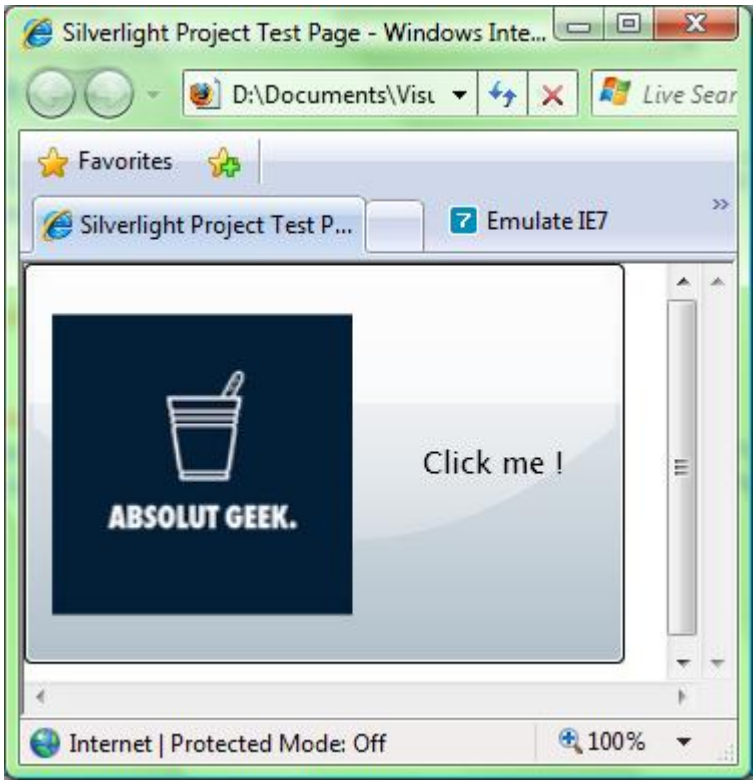
Il ne reste plus qu'à rendre le tout plus joli, mais ça je vous laisse le faire.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

9 - Les Templates

Silverlight nous permet de customiser l'apparence de nos contrôles à volonté, par exemple un bouton peut ne pas contenir que du texte.

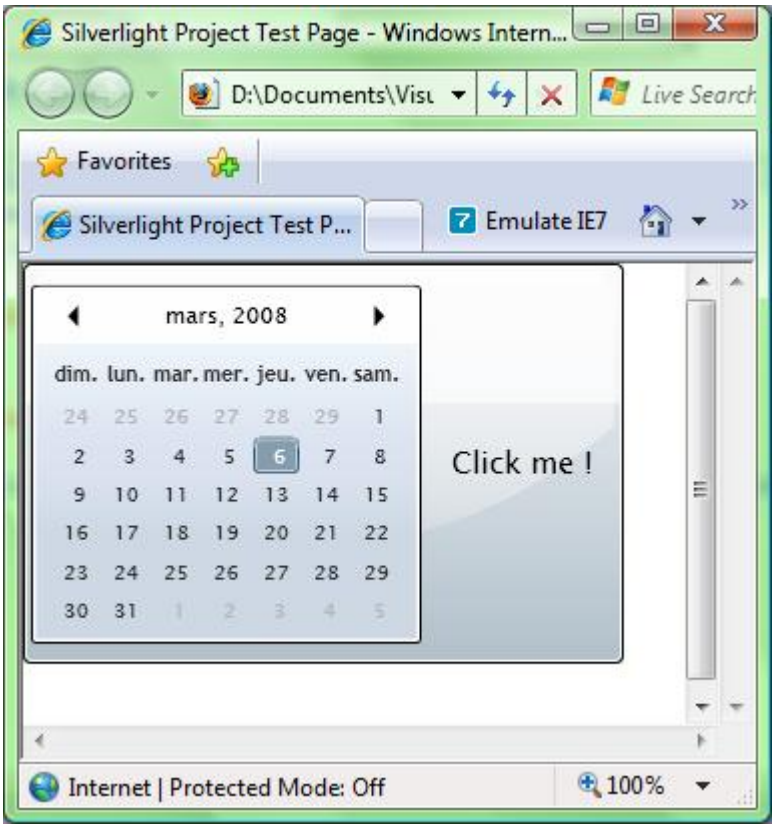
```
<Button>
    <Button.Content>
        <StackPanel Orientation="Horizontal">
            <Image Source="absolut.jpg" Width="150" Height="150" Margin="10"/>
            <TextBlock Text="Click me !" VerticalAlignment="Center" Margin="25"/>
        </StackPanel>
    </Button.Content>
</Button>
```



Ou même :

```
<Button>
    <Button.Content>
        <StackPanel Orientation="Horizontal">
            <Calendar/>
            <TextBlock Text="Click me !" VerticalAlignment="Center" Margin="15"/>
        </StackPanel>
    </Button.Content>
</Button>
```


Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.



Maintenant on va voir comment complètement changer l'apparence d'un contrôle. Imaginons que dans toute notre application nous voulions des boutons ayant ce look.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

Nous pouvons réaliser cela en ajoutant un style dans notre **App.xaml**, ou nous allons définir la propriété **Template**.

```
<Style x:Key="StarButton" TargetType="Button">
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate>
                <Grid>
                    <Image Source="etoile.png" Width="200" Height="200"/>

  <TextBlock Text="Push Me !" HorizontalAlignment="Center" VerticalAlignment="Center"/>
                </Grid>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>
```

Et lorsque nous créons un bouton ça ne change pas :

```
<Button Style="{StaticResource StarButton}"/>
```

Le problème ici est que la taille et le texte du bouton sont mis en dur dans le code, mais nous pouvons évidemment remédier à ça.

```
<Style x:Key="StarButton" TargetType="Button">
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="Button">
                <Grid>

  <Image Source="etoile.png" Width="{TemplateBinding Width}" Height="{TemplateBinding Height}"/>
                    <ContentPresenter HorizontalAlignment="Center"
                                      VerticalAlignment="Center"
                                      Content="{TemplateBinding Content}"/>
                </Grid>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>
```

Maintenant la taille de l'image sera la même que la taille spécifiée lors de la création du bouton.

J'ai aussi modifié le **TextBlock** par un **ContentPresenter**, afin de pouvoir mettre n'importe quel contrôle dans notre bouton (comme plus haut avec un calendrier ou une image).

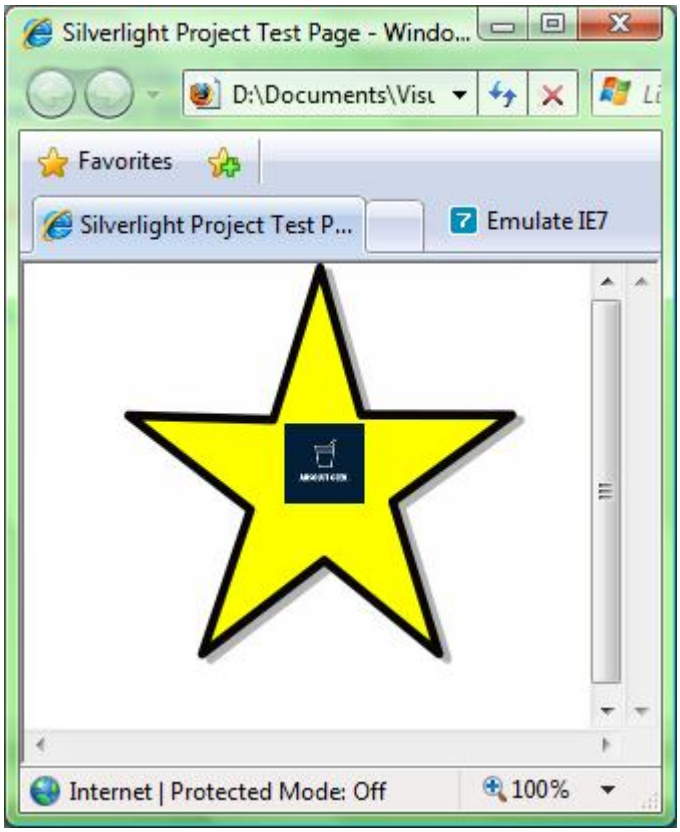
```
<Button Style="{StaticResource StarButton}" Width="200" Height="200" Content="Push Me !" />
```


Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

Ou encore :

```
<Button Style="{StaticResource StarButton}" Width="200" Height="200">
    <Button.Content>
        <Image Source="absolut.jpg" Width="40" Height="40"/>
    </Button.Content>
</Button>
```


Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

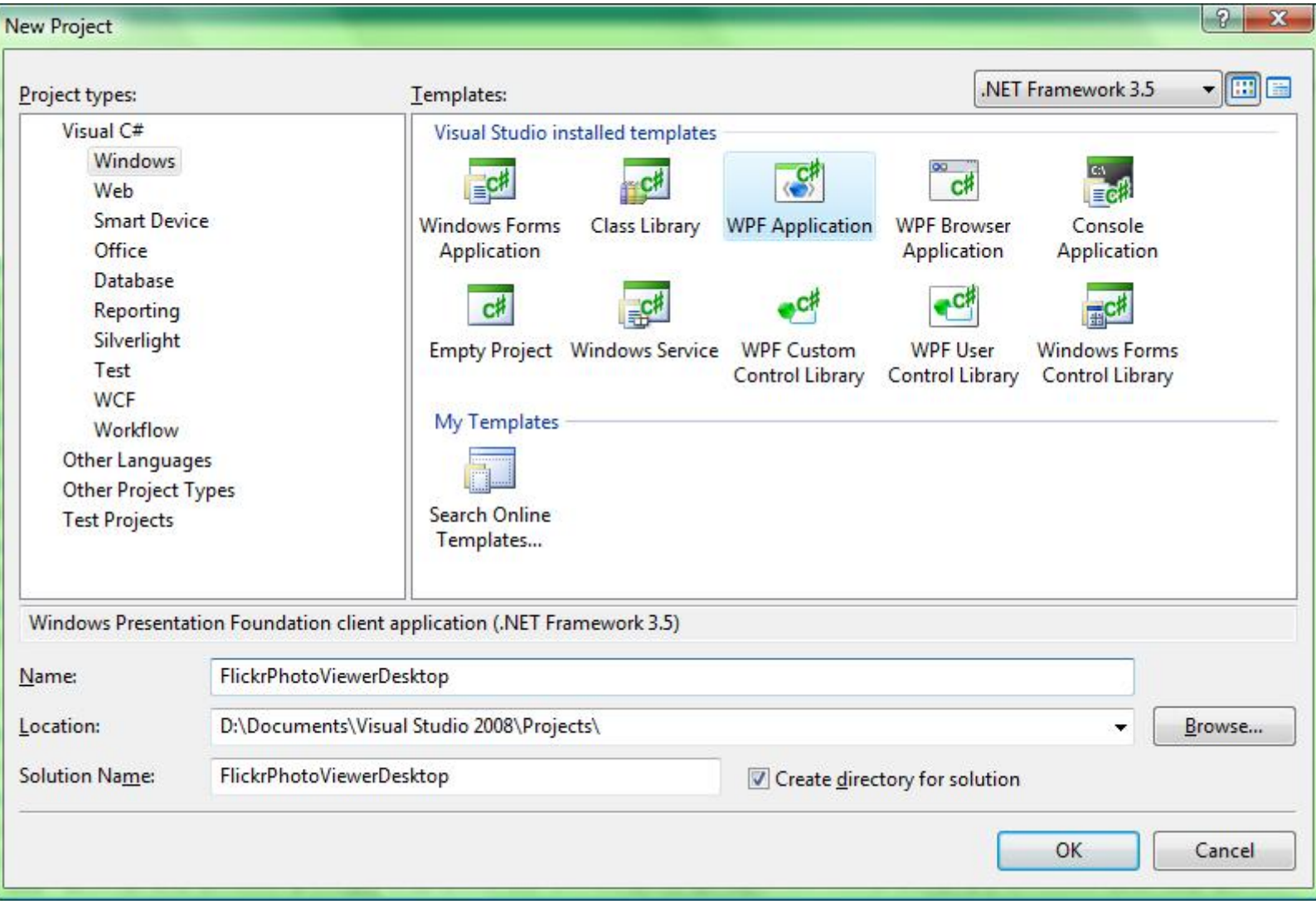


Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

10 - De Silverlight à WPF

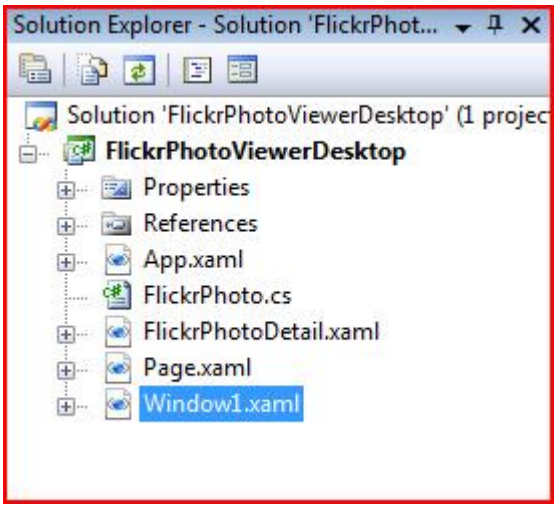
Dans cette partie nous allons voir comment passer de notre application Silverlight donc Web à une application WPF donc client lourd.

Tout d'abord il faut créer une nouvelle application WPF.



Ensuite il suffit simplement de copier nos fichiers de notre projet Silverlight dans notre nouveau projet WPF (à priori plus tard ce sera automatique).

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.



Une fois ceci fait il faut également changer les namespaces de nos fichiers Silverlight (ceci sera amélioré par la suite).

```
<UserControl xmlns:my="clr-namespace:System.Windows.Controls;assembly=System.Windows.Controls.Data"
                                x:Class="FlickrPhotoViewer.Page"
    xmlns="http://schemas.microsoft.com/client/2007"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:Flickr="clr-namespace:FlickrPhotoViewer;assembly=FlickrPhotoViewer">
```

Devient par exemple :

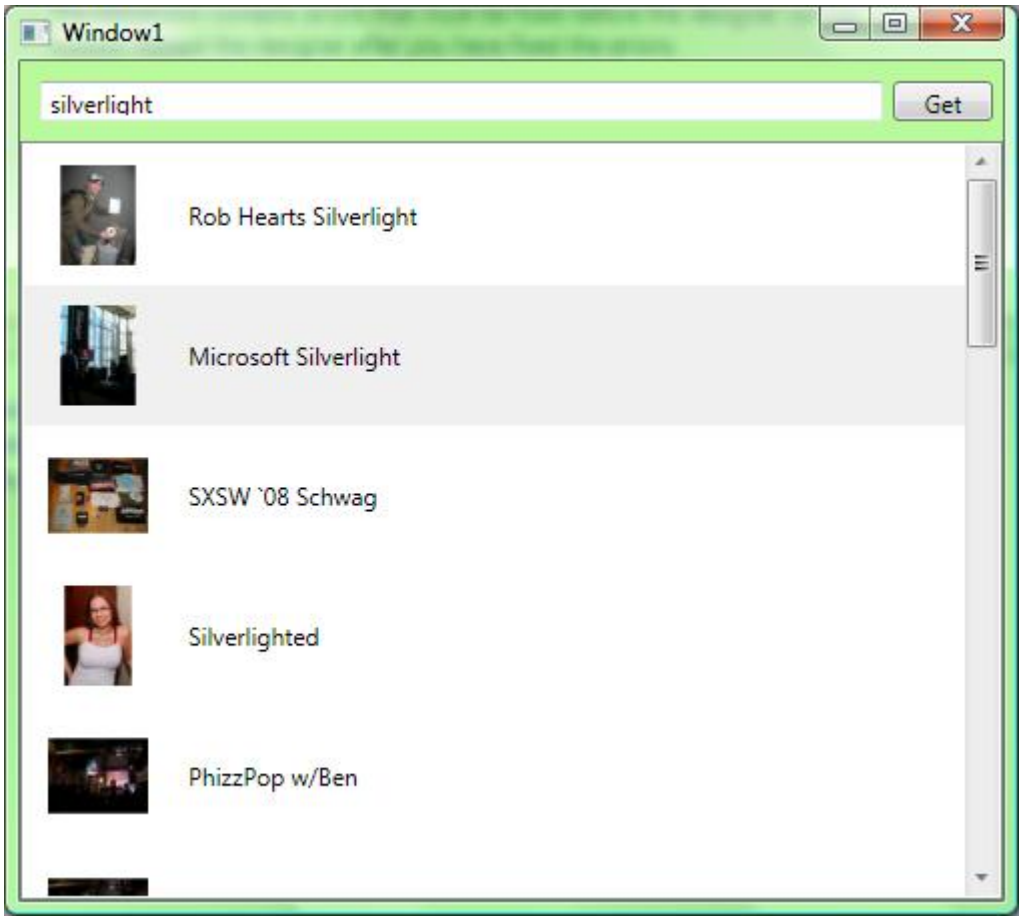
```
<UserControl x:Class="FlickrPhotoViewerDesktop.Page"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:Flickr="clr-namespace:FlickrPhotoViewerDesktop">
```

Ensuite on ouvre le fichier **Window1.xaml** et on rajoute ceci :

```
<Window x:Class="FlickrPhotoViewerDesktop.Window1"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:Flickr="clr-namespace:FlickrPhotoViewerDesktop"
    Title="Window1" Height="300" Width="300">
    <Grid>
        <Flickr:Page />
    </Grid>
</Window>
```

Et voilà nous avons maintenant une application en WPF et ce sans avoir modifié énormément de choses.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.



Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

11 - L'application en ligne

Et voici sans plus attendre le rendu final : **FlickrPhotoViewer**

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

12 - Conclusion

Comme vous avez pu le voir au fil de cet article, Silverlight 2 n'a plus rien à voir avec Silverlight 1.1. Il se rapproche de plus en plus de WPF et c'est tant mieux. J'espère que les ingénieurs de chez Microsoft nous réserverons encore de bonnes surprises pour la bêta 2 ou même la release.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.

13 - Remerciements

Merci à **The_badger_man** et **Cardi** pour leurs corrections, ainsi qu'à toute l'équipe .NET.

Copyright 2008 - Benjamin Roux. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérets. Droits de diffusion permanents accordés à developpez LLC.