

Déploiement de contrôles web personnalisés pour SharePoint.

ASP.Net, et SharePoint, sont livrés en standard avec un certain nombre de contrôles.

Cependant, il peut arriver que vous souhaitiez développer votre propre contrôle, et le déployer au sein de votre ferme SharePoint.

Dans cet article, nous allons voir comment déployer nos propres contrôles au sein de SharePoint.

Présentation des types de contrôles personnalisés.

En asp.net, on va principalement avoir 2 types de contrôles ; les custom controls et les user controls.

La seule différence qui subsiste entre ces types de contrôles est la manière de les concevoir ; un custom controls sera conçu directement avec du code .net, tandis qu'un user controls sera généré de manière visuelle, ou en éditant du code un peu à la (x)html, et sera en fait un contrôle fait à partir de contrôles déjà préexistants.

Le choix de l'un ou l'autre des contrôles dépend essentiellement de votre besoin ; si vous avez besoin de rajouter un contrôle complètement différent de ce qui existe déjà, le custom control sera peut-être celui que vous choisirez ; si par contre, vous souhaitez mettre en œuvre une fonctionnalité qui pourrait être mise par plusieurs contrôles déjà existant, peut-être devriez vous utiliser les user controls.

SharePoint rajoute un 3^{ème} type de contrôles, qui sont les delegate controls. Un delegate control est un contrôle conçu pour fournir une certaine fonctionnalité, et qui pourra être facilement remplacé par un autre contrôle portant le même identifiant et défini comme contrôle prioritaire.

Nous verrons les delegate controls dans un prochain article.

Création et déploiement d'un custom control.

Pour cela, nous allons commencer par créer notre contrôle.

Création du contrôle.

Créer un nouveau projet de type Class Library et ajoutez y le code suivant.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Web.UI;
using System.Web.UI.WebControls;
using Microsoft.SharePoint;
namespace Coforcert.SharePoint.Controls.CustomControls {
    public class SiteLists : WebControl {
        public override void RenderControl(HtmlTextWriter writer) {
            SPWeb currentWeb = SPContext.Current.Web;
            writer.Write("<-- Listes du Site --><br/>");
        }
    }
}
```

```

        foreach (SPList list in currentWeb.Lists)
            writer.Write(string.Format("Liste : {0}<br/>Modèle de Base
: {1}<br/>Description : {2}<br/><br/>", list.Title,
list.BaseType.ToString(), list.Description));
    }
}

```

Ce code va nous permettre de récupérer toutes les listes du site ou le contrôle sera déployé.

Déploiement du contrôle

Ensuite, compilez votre assembly et ajoutez-la, soit dans le GAC (nécessite de signer l'assembly), soit dans le répertoire bin de votre application web.

Enregistrement du contrôle

Afin de pouvoir utiliser vos contrôles comme vous le faites avec vos webparts, il va falloir les marquer comme sur.

Pour cela, ouvrez le web.config de l'application concerné, qui se situe par défaut dans le répertoire « C:\inetpub\wwwroot\wss\VirtualDirectories\n°dePort », et ajoutez-y la ligne suivante, sous réserve que vous ayez utilisé le même namespace et que votre assembly s'appelle CustomControl.dll. A défaut, remplacez par vos valeurs.

```

<SafeControls>
<!-- Les autres Safe Control sont déjà présents -->
  <SafeControl Assembly="CustomControl"
Namespace="Coforcert.SharePoint.Controls.CustomControls" TypeName="*"
Safe="True"/>
</SafeControls>

```

Intégration du contrôle

Une fois le contrôle déployé, il suffit maintenant de l'intégrer au sein d'une page quelconque, ou alors dans toutes les pages via les master pages.

Dans notre cas, nous allons le déployer au sein d'une seule page.

Pour cela, nous allons utiliser SharePoint Designer, pour éditer cette page.

Une fois la page ouverte, il nous suffit de déclarer notre contrôle dans l'une des zones de contenu à notre disposition.

Dans notre exemple, nous déployons notre contrôle dans le PlaceholderMain.

```

<%@ Page Language="C#" masterpagefile="~/masterurl/default.master" title="Page de Test des
Composants" inherits="Microsoft.SharePoint.WebPartPages.WebPartPage, Microsoft.SharePoint,
Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c"
meta:webpartpageexpansion="full" meta:progid="SharePoint.WebPartPage.Document" %>
<%@ Register Assembly="CustomControl"
Namespace="Coforcert.SharePoint.Controls.CustomControls" TagPrefix="MySPControls" %>
<asp:Content id="Content1" runat="server" contentplaceholderid="PlaceholderMain">
  <MySPControls:SiteLists id="sl01" runat="server"/>
</asp:Content>

```

Et voici le résultat.



Comme vous avez pu le constater, il est extrêmement facile de créer des pages en y déployant des contrôles entièrement custom.

Nous allons à présent voir le 2^{ème} type de contrôle asp.net, les user controls.

Création et déploiement d'un user control

Comme précédemment, nous allons commencer par créer notre contrôle.

Pour cela, nous allons créer un projet de site web, et y ajouter un nouvel item de type web user control (fichier .ascx).

Ici, nous allons créer un contrôle qui nous permet de créer des nouvelles doclibs sur notre site.

Création du contrôle

Cette étape va se dérouler en 2 phases.

Durant la 1^{ère} phase, nous allons créer visuellement notre contrôle.

Pour cela, ajoutez à l'interface 2 labels, 2 textbox et 1 bouton ; configurer l'une des textbox pour un affichage multiligne.

Vous devriez obtenir un rendu comme ci-dessous.

The screenshot shows a web control interface. It features a label 'Nom :' followed by a single-line text input field. Below it is a label 'Description :' followed by a multi-line text input field with a vertical scrollbar on the right. At the bottom left of the form is a button labeled 'Ajouter'.

Une fois cette étape réalisé, nous allons passer en affichage html, afin de rajouter le code exécuté lors du click sur le bouton.

Nous allons également importer la dll Microsoft.SharePoint.dll et le namespace using Microsoft.SharePoint, étant donné que nous allons utiliser le modèle objet de SharePoint.

Voici le code html de notre contrôle.

```
<%@ Control Language="C#" %>
<%@ Assembly Name="Microsoft.SharePoint, Version=12.0.0.0,
Culture=neutral, PublicKeyToken=71e9bce11e9429c" %>
<%@ Import Namespace="Microsoft.SharePoint" %>

<script runat="server">
    protected void Button1_Click(object sender, EventArgs e)
    {
        SPContext.Current.Web.Lists.Add(this.TextBox1.Text,
this.TextBox2.Text, SPListTemplateType.DocumentLibrary);
    }
</script>

<asp:Label ID="Label1" runat="server" Text="Label">Nom : </asp:Label>
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox><br />
<asp:Label ID="Label2" runat="server" Text="Label">Description :
</asp:Label>
<asp:TextBox ID="TextBox2" runat="server"
TextMode="MultiLine"></asp:TextBox><br />
<asp:Button ID="Button1" runat="server" Text="Ajouter"
onclick="Button1_Click" />
```

Ceci étant fait, nous allons à présent déployer notre contrôle.

Déploiement du contrôle

Le code de notre contrôle étant embarqué dans notre fichier ascx, nous n'avons pas d'assembly à déployer dans le GAC ou dans le bin.

Pour déployer notre contrôle, il nous suffit de copier le fichier .ascx dans un répertoire prévu à l'effet, le répertoire ControlTemplates : « C:\Program Files\Fichiers communs\Microsoft Shared\web server extensions\12\TEMPLATE\CONTROLTEMPLATES ».

Ceci étant fait, passons à l'étape suivante.

Enregistrement de notre contrôle

L'étape suivante devrait donc être l'enregistrement de notre contrôle comme sur dans le web.config ; cependant, comme je vous l'ai dit précédemment, le répertoire ControlTemplate est prévu à cet effet, à tel point qu'il est déjà marqué comme sur dans le web.config.

Si vous ouvrez le fichier web.config, vous y trouverez la ligne

```
<SafeControl Src="~/_controltemplates/*" IncludeSubFolders="True"
Safe="True" AllowRemoteDesigner="True" />
```

_controltemplates correspond au répertoire virtuel qui est mappé sur le répertoire physique ControlTemplate. Donc cette ligne approuve tous les ascx stockés dans les sous répertoires de ce dossiers.

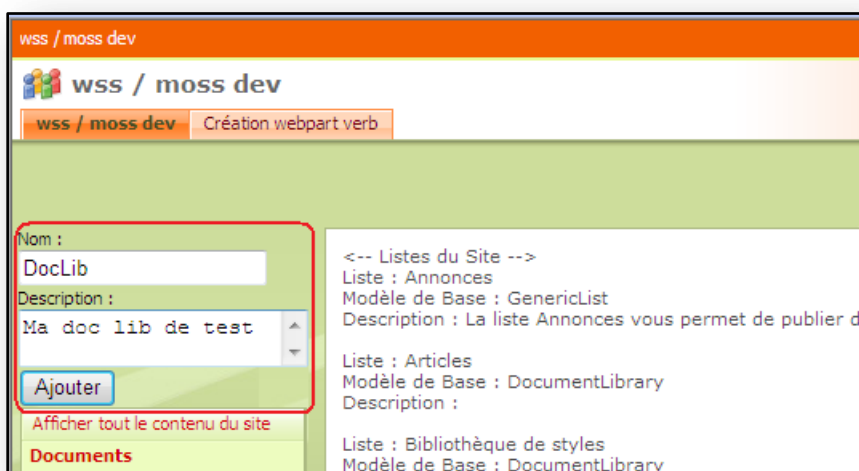
Intégration de notre contrôle

Nous allons à nouveau éditer une page avec SharePoint Designer afin d'intégrer notre contrôle dans une page SharePoint.

Une fois la page .aspx ouverte avec SD, il nous suffit de rajouter les lignes ci-dessous pour intégrer notre contrôle dans la zone de contenu PlaceholderLeftNavBarTop.

```
<%@ Page Language="C#" masterpagefile="~/masterurl/default.master" title="Page de Test des
Composants" inherits="Microsoft.SharePoint.WebPartPages.WebPartPage, Microsoft.SharePoint,
Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c"
meta:webpartpageexpansion="full" meta:progid="SharePoint.WebPartPage.Document" %>
<%@ Register TagPrefix="MySPControls" TagName="AddDocLib"
src="~/_controltemplates/AddDocLib.ascx" %>
<asp:Content id="Content2" runat="server" contentplaceholderid="PlaceholderLeftNavBarTop">
    <MySPControls:AddDocLib id="adl01" runat="server"/>
</asp:Content>
```

Et voici le résultat.



Conclusion

Comme vous avez pu le constater au travers de cet article, il est très simple de customiser entièrement l'interface de SharePoint en y rajouter nos propres contrôles. Ainsi, nous ne sommes pas uniquement limités à l'ajout de webparts dans des webparts zones, mais pouvons customiser à souhait nos pages.