

Remplir dynamiquement un contrôle

par Christian Wenz Matt-K (Traducteur) ([Page d'accueil](#))


Date de publication : 04 mai 2009

Dernière mise à jour :

Le contrôle DynamicPopulate de l'ASP.NET AJAX Control Toolkit fait appel à un web service (ou un méthode de la page) et remplit le contrôle ciblé de la page avec les valeurs récupérées, sans rafraîchir la page.

Traduction.....	3
Présentation.....	3
Étapes.....	3

Traduction

Cet article est la traduction la plus fidèle possible de l'article original de Christian Wenz:  **Dynamically Populating a Control**

Présentation

Le contrôle **DynamicPopulate** de l'ASP.NET AJAX Control Toolkit fait appel à un web service (ou un méthode de la page) et remplit le contrôle ciblé de la page avec les valeurs récupérées, sans rafraîchir la page. Ce tutoriel montre comment mettre en place ceci.

Étapes

Tout d'abord, nous avons besoin d'un web service ASP.NET qui implémente la méthode qui sera appelée par le **DynamicPopulate**. Le classe du web service requière l'attribut **ScriptService** qui est définie dans **Microsoft.Web.Script.Services**; sans cela ASP.NET AJAX ne peut créer le proxy JavaScript côté client pour le web service requis par le **DynamicPopulate**.

Cette méthode web attendra en argument un string, nommé contextKey, puisque le contrôle **DynamicPopulate** envoie une information de contexte à chaque appel du web service. Le web service suivant retourne la date courante au format représenté par l'argument **contextKey**:

```
<%@ WebService Language="C#" Class="DynamicPopulate" %>
using System; using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Web.Script.Services;

[ScriptService] public class DynamicPopulate : System.Web.Services.WebService
{
    [WebMethod]
    public string getDate(string contextKey)
    {
        string myDate = "";
        switch (contextKey)
        {
            case "format1":
                myDate = String.Format("{0:MM}-{0:dd}-{0:yyyy}", DateTime.Now);
                break;
            case "format2":
                myDate = String.Format("{0:dd}.{0:MM}.{0:yyyy}", DateTime.Now);
                break;
            case "format3":
                myDate = String.Format("{0:yyyy}/{0:MM}/{0:dd}", DateTime.Now);
                break;
        }
        return myDate;
    }
}
```

Le web service est enregistré dans le fichier **DynamicPopulate.cs.asmx**. Une alternative, aurait été d'implémenter une méthode **getDate()** au sein de la page ASP.NET contenant le **DynamicPopulate**.

Dans l'étape suivante, nous allons créer un nouveau fichier ASP.NET. Comme toujours, la première étape est d'inclure le **ScriptManager** dans la page courante pour charger la bibliothèque ASP.NET AJAX et faire fonctionner le Control Toolkit:

```
<asp:ScriptManager ID="asm" runat="server" />
```

Ensuite, ajoutons un label (par exemple en utilisant le tag HTML du même nom, ou le contrôle web `<asp:Label />` qui affichera plus tard le résultat de l'appel au web service.

```
<label id="myDate" runat="server" />
```

Un bouton HTML (comme le contrôle HTML, vu que nous avons pas besoin d'un postback vers le serveur) qui sera utilisé pour lancer l'évènement qui remplira dynamiquement notre label:

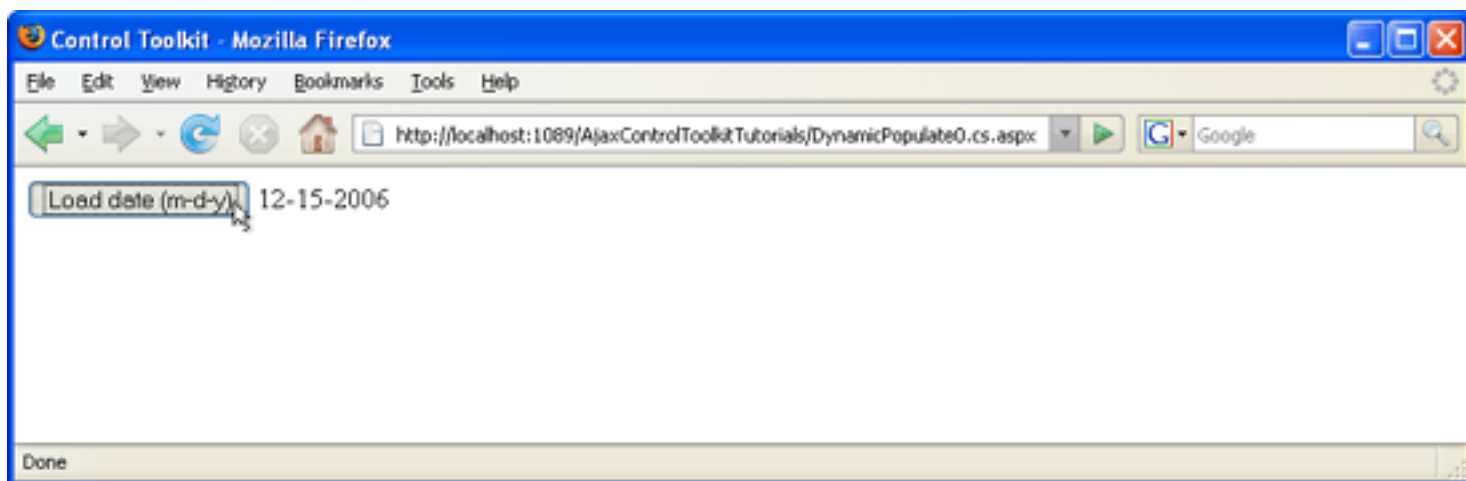
```
<input type="button" id="Button1" runat="server" value="Load date (m-d-y)" />
```

Enfin, nous avons besoin du contrôle **DynamicPopulateExtender** pour connecter les choses entre elles. Les attributs suivants doivent être définis (avec bien sûr ID et `runat="server"`):

- **TargetControlID** : la cible où sera affiché le résultat de l'appel au web service
- **ServicePath** : chemin vers le web service (à ne pas préciser si vous utilisez une méthode dans la page)
- **ServiceMethod** : nom de la méthode web, ou méthode de la page
- **ContextKey** : information de contexte qui doit être envoyée au web service
- **PopulateTriggerControlID** : élément qui exécutera l'appel au web service
- **ClearContentsDuringUpdate** : si l'on doit vider l'élément au cours de l'appel au web service

Comme vous pouvez le voir, le contrôle requiert certaines informations, mais assez simple à mettre en place. Voici le balisage du **DynamicPopulateExtender** pour notre scénario:

```
<ajaxToolkit:DynamicPopulateExtender ID="dpel" runat="server"
ClearContentsDuringUpdate="true"
TargetControlID="myDate" ServicePath="DynamicPopulate.cs.aspx"
ServiceMethod="getDate"
ContextKey="format1"
PopulateTriggerControlID="Button1" />
```



Un click sur le bouton récupère la date du serveur