

Remplir dynamiquement un contrôle JavaScript

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. Il est aussi possible d'effectuer ce remplissage en utilisant du JavaScript côté client.

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 Using JavaScript Code](#)

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. Il est aussi possible d'effectuer ce remplissage en utilisant du JavaScript côté client.

Étapes

Tout d'abord, nous avons besoin d'un web service ASP.NET qui implémente la méthode qui sera appelée par le **DynamicPopulateExtender**. Le web service implémente la méthode getDate() qui requiert un argument de type string, nommée contextKey, puisque le **DynamicPopulate** envoie une information de contexte à chaque appel du service web. Voici le code (dans le fichier **DynamicPopulate.cs.asmx** qui récupère la date dans un de ces trois formats:

```
<%@ 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;
    }
}
```

Ensuite, nous allons créer un nouveau site ASP.NET et qui commence avec le **ScriptManager**:

```
<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" />
```

Puis, incluons un **DynamicPopulateExtender** et renseignons les informations du web service, la cible, mais pas le nom du contrôle qui lancera l'événement de remplissage, cela sera effectué plus tard en utilisant du JavaScript!

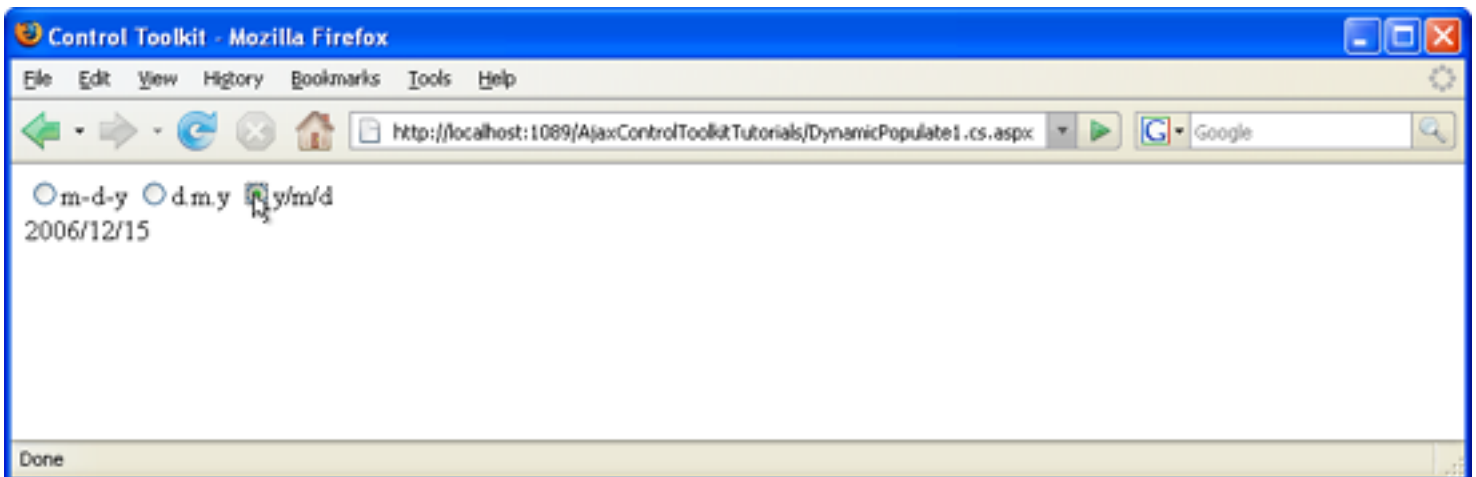
```
<ajaxToolkit:DynamicPopulateExtender ID="dpe1" runat="server"
ClearContentsDuringUpdate="true"
TargetControlID="myDate" ServicePath="DynamicPopulate.cs.aspx"
ServiceMethod="getDate" />
```

Maintenant, la partie JavaScript. La fonction **\$find()**, définie par la bibliothèque ASP.NET AJAX, retourne la référence côté serveur des objets de l'ASP.NET AJAX Control Toolkit tel que le **DynamicPopulateExtender**. Dans ce fichier, **\$find("dpe")** retourne la référence du **DynamicPopulateExtender** contenu dans la page. Il expose la méthode nommée **populate()** qui lancera le processus de remplissage automatique. La méthode **populate()** exige un argument: la clé de contexte qui servira d'argument à la méthode web **getDate()**. Par exemple, **\$find("dpe").populate("format1")** remplira le label avec la date courante au format mois-jour-année.

Dans le but de rendre l'exemple un peu plus flexible, l'utilisateur peut désormais choisir entre différent format de date. Pour chacun d'eux, nous définissons un bouton de type HTML "radio". Dès que l'utilisateur clique sur un bouton, le code JavaScript remplira dynamiquement le label avec la date au format sélectionné. Voici le code de ces boutons de type "radio":

```
<asp:Panel ID="panell1" runat="server">
    <input type="radio" id="rb1" name="format" value="format1" runat="server" onclick="$find('dpe1').populate(this
d-y
    <input type="radio" id="rb2" name="format" value="format2" runat="server" onclick="$find('dpe1').populate(this
    <input type="radio" id="rb3" name="format" value="format3" runat="server" onclick="$find('dpe1').populate(this
m/d
</asp:Panel>
```

Notez qu'au sein de contexte du radio button, l'expression JavaScript **this.value** fait référence à la valeur du bouton courant, qui est exactement la même information que la méthode **getDate()** peut utiliser.



Un click sur le bouton récupère la date du serveur, au format spécifié