

MSN LIVE Search, utilisation du moteur dans vos développements .Net

Par laurent Geffroy
(<http://www.laurentgeffroy.com>)
8 mars 2006

Introduction

J'ai appris aujourd'hui la mise en ligne de <http://search.live.com>, le nouveau moteur de recherche de Microsoft. Un peu déçu par la lourdeur du msn actuel et utilisateur régulier (voir accro) de google, telle ne fut pas ma surprise !!

Une interface simple, mais efficace, avec une magnifique innovation sur la méthode de navigation entre les pages. Bref, je suis conquis.

Mais la curiosité du spécialiste .Net que j'essaie d'être a été accentuée par le résultat de la recherche. Il y a de l'AJAX dans l'air ! Et qui dit AJAX, dit Flux XML.

Me voici donc à utiliser Fiddler pour voir ce qu'il se cache derrière ce redoutable engin. Nouvelle surprise ! Les résultats sont retournés via un flux rss !!!!

Mon cerveau a immédiatement compilé, un peu comme le Framework 2.0 et me voici en train de vous faire part de mes petits travaux à savoir :

- Faire un UserControl qui exploite la puissance de MSN Live Search
- Transformer un flux rss avec le Framework 2.0

Le UserControl

Rien de plus simple, un champ de saisir, un bouton, un lien qui donnera l'url du flux rss et un placeholder pour ajouter ma transformation XSLT.

Le plus intéressant, le code derrière le bouton :

```
protected void Button1_Click(object sender, EventArgs e)
{
    // Lance la recherche
    if (txt_search.Text.Length > 0)
    {
        // J'encode ma chaine de recherche
        string v_search = Server.UrlEncode(txt_search.Text);
        // Je complete ma requete http
        string v_url = "http://search.live.com/live/query.aspx?q=" +
v_search + "&source=web&count=20&first=1";

        // En français ou US
        if (cb_fr.Checked)
            v_url += "&mkt=fr-FR&v=3";
        else
            v_url += "&mkt=en-US&v=3";

        // Le chemin de ma feuille de style
        string v_xsl = "http://localhost/livesearch/transform.xsl";

        // Mon controle HTML
        HtmlGenericControl v_out = new HtmlGenericControl();
        // Je mets dans mon controle le résultat de la transformation
        v_out.InnerHtml = Transformation(v_url, v_xsl);

        // J'initialise une image (rss)
        Image img = new Image();
        img.BorderStyle = BorderStyle.None;
        img.ImageUrl = "/livesearch/rss.gif";

        // Je l'ajoute comme lien
        HyperLink1.Controls.Add(img);
        // Je mets l'url du flux rss de résultat de recherche comme
        //lien
        HyperLink1.NavigateUrl = v_url;
        HyperLink1.Target = "_blank";

        // J'ajoute mon controle dans le placeHolder
        ph_result.Controls.Add(v_out);

        // Requete supplémentaire permettant l'avoir un complément de
        // recherche grace au spelling

        //http://search.live.com/live/query.aspx?q=MARECHERCHE&source=s
        //pelling&mkt=en-US&v=3
    }
}
```

Enfin la transformation du flux XML via ma feuille de style dans la méthode Transformation.

```
/// <summary>
/// Transforme les données
/// </summary>
/// <param name="p_xml">Fichier XML</param>
/// <param name="p_xsl">Fichier XSL</param>
/// <returns>String</returns>
private string Transformation(string p_xml, string p_xsl)
{
    // Charge le fichier XML
    XmlDocument v_xml = new XmlDocument();
    v_xml.Load(p_xml);

    // Charge le fichier XSL
    XmlDocument v_xsl = new XmlDocument();
    v_xsl.Load(p_xsl);

    // On créé un lecteur du fichier XML
    XmlNodeReader v_xmlReader = new XmlNodeReader(v_xml);

    // Puis le transformateur XSL
    XslCompiledTransform v_xslt = new XslCompiledTransform();
    v_xslt.Load(v_xsl, null, null);

    // Chaîne utilisée pour le retour
    StringBuilder v_output = new StringBuilder();

    // Construit la méthode d'écriture du flux HTML
    XmlWriterSettings v_settings = new XmlWriterSettings();
    v_settings.ConformanceLevel = ConformanceLevel.Auto;
    v_settings.Indent = true;

    // Fait la transformation
    XmlWriter v_html = XmlWriter.Create(v_output, v_settings);
    v_xslt.Transform(v_xmlReader, null, v_html, new XmlUrlResolver());
}
```

Notez la bidouille : Le flux XML retourne des caractères non reconnus autour des mots clés. En fouillant dans le javascript (search.js), j'ai trouvé qu'il s'agit des balises et .

N'étant pas un pro de l'encodage et de la manipulation de chaînes, ma méthode est quelque peu barbare et mérite d'être améliorée.

```
// Le fichier issu de MSN Live Search comporte des caractères
// non reconnus pour le <b> et le </b>
// Parcours la chaîne et transforme les données
// /uE000 et /uE001 sont les équivalents des balises bold
// Trouvé dans le fichier search.js du moteur de recherche !!!
string cout = v_output.ToString();

int vpb = cout.IndexOf("\uE000");
while (vpb >= 0)
{
    cout = cout.Substring(0, vpb) + "<b>" + cout.Substring(vpb + 1,
cout.Length - (vpb + 1));
    vpb = cout.IndexOf("\uE000");
}

vpb = cout.IndexOf("\uE001");
while (vpb >= 0)
{
    cout = cout.Substring(0, vpb) + "</b>" + cout.Substring(vpb +
1, cout.Length - (vpb + 1));
    vpb = cout.IndexOf("\uE001");
}

// Retourne la valeur
return cout;
}
```

Ma feuille de style

Rien de très complexe, a part :

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:opensearch="http://a9.com/-/spec/opensearchresponse/1.1/"
xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:livesearch="http://www.live.com/search/2005/livesearch">
```

Nécessaire pour attaquer correctement vos nœuds.

Le résultat



Conclusion

Je suis plutôt agréablement surpris par la démarche de Microsoft qui ouvre son API d'une manière très simple.

Reste à gérer les pages suivantes, gérables dans l'argument de l'url.

Bonne découverte à tous

Laurent GEFROY

<http://www.laurentgeffroy.com>

Pour aller plus loin :

Le moteur de recherche

<http://search.live.com>

Fiddler

<http://www.fiddlertool.com/fiddler/>

Transformation XSL / XML avec le framework 2.0

http://www.csharpfr.com/codes/TRANSFORMATION-XSLT-NET_33721.aspx