

MASH UP LiveSearch / GoogleSearch avec ASP.Net 2.0 et le Framework Ajax 1.0

Par Laurent Geffroy
<http://www.laurentgeffroy.com>

[Version PDF à télécharger](#)
[Code Source](#)

Droit de diffusion et copyrights :

Ce tutoriel ne peut être diffusé sur d'autres sites Web ou tout autre support sans l'autorisation préalable de son auteur. L'utilisation des sources est totalement libre de droits.
Toutes les marques citées appartiennent à leurs auteurs.

Avant propos :

Il y a quelques semaines, j'ai intégré GoogleSearch dans mon site grâce à son API. Le rendu est plutôt sympa et efficace, tout du moins pour les pages qui sont référencées par Google. Microsoft propose lui aussi son API pour le moteur de recherche Live.com. De mon esprit curieux, sort un mash-up plutôt inattendu avec les deux frères ennemis, d'où l'envie de marier les deux plutôt de que faire 2 zones de recherches.

Ce mini tutoriel vous présentera la manière dont ce mash-up a été réalisé sur mon site internet avec :

- Les pré requis
- La création du Webservice d'interrogation de MSN Live Search
- Le code behind nécessaire
- L'implémentation de GoogleSearch et de LiveSearch en Javascript

Les sources, libres de droits sont disponibles en téléchargement sur
<http://www.laurentgeffroy.com/specific/datas/tutoriels/LiveGoogleSearch/MashUpLiveGoogleSearch.zip>

Les pré requis

1. Pour démarrer, vous devez référencer le Webservice de recherche MSN sur <http://soap.search.msn.com/webservices.asmx?wsdl>
2. Votre Web.Config, doit contenir les éléments nécessaires à l'utilisation d'ASP.Net AJAX 1.0 que vous trouverez sur http://ajax.asp.net/documentation/Migration_Guide_RC_to_RTM.aspx
3. Quatre variables ont été mises dans le Web.Config à savoir :
 - Votre AppKey MSN, que vous pourrez générer sur <http://search.msn.com/developer>
 - Votre API Key Google, générée sur <http://code.google.com/apis/ajaxsearch/signup.html>
 - Le domaine du site, permettant ainsi de limiter les recherches
 - Le nom de votre site, pour personnaliser la recherche Google
4. Référencer la feuille de style de Google dans la page ASPX :
`<link href="http://www.google.com/uds/css/gsearch.css" type="text/css" rel="stylesheet"/>`
5. votre page ASPX doit contenir un ScriptManager :

```
<asp:ScriptManager ID="ScriptManager1" runat="server" />
```

6. Votre fichier CSS doit absolument avoir le code suivant pour éviter d'avoir 2 zones de formulaires Google :

```
.gsc-control form.gsc-search-box {display : none;}
```

Le Webservice d'interrogation de MSN Live Search

Le webservice est localisé dans le fichier App_Code/WebService.cs. Pour que mon Webservice soit accessible via mon javascript client, ne pas oublier de mettre l'attribut System.Web.Script.Services.ScriptService

```
[System.Web.Script.Services.ScriptService]
[WebService(Namespace = "http://www.laurentgeffroy.com/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class Webservice : System.Web.Services.Webservice {
```

Pour la WebMethod, il faut également affecter l'attribut System.Web.Script.Services.ScriptMethod. L'utilisation est plutôt simple. On crée une nouvelle instance du Webservice MSNSearchService auquel on affecte une requête. La requête est initialisée avec l'objet SearchRequest et les propriétés Query. La Query est modifiée pour limiter les recherches sur un domaine particulier. C'est également dans SearchRequest que vous affecterez la AppKey fournie.

Les résultats sont placés dans mes propres classes. Il est je pense possible d'utiliser directement le sourceResponse.Results comme valeur de retour. Mon but était de limiter la quantité de données à retourner au script coté client.

Votre Webservice est construit ! Le voici en détail :

```
[WebMethod(Description = "Récupère le résultat de la recherche sur le site actuel")]
[System.Web.Script.Services.ScriptMethod]
public LiveSearch.Resultats MsnLiveSearch(string p_search)
{
    // Variable de retour
    LiveSearch.Resultats v_result = new LiveSearch.Resultats();

    // Nouvelle instance du Webservice
    MSNSearchService _search = new MSNSearchService();

    SearchRequest _searchRequest = new SearchRequest();
    SearchResponse _searchResponse;
    Result[] sourceResults;

    try
    {
        SourceRequest[] _sr = new SourceRequest[1];

        // Recherche uniquement sur le Web
        _sr[0] = new SourceRequest();
        _sr[0].Source = SourceType.Web;

        // Construction de la recherche
        _searchRequest.Query = "site:" + ConfigurationManager.AppSettings["SiteWeb"] + " " + p_search;
        _searchRequest.Requests = _sr;

        // Fourniture du LiveAppID et de la culture
        _searchRequest.AppID = ConfigurationManager.AppSettings["LiveAppID"];
        _searchRequest.CultureInfo = "FR-fr";

        // Invocation et récupération du résultat
        _searchResponse = _search.Search(_searchRequest);

        foreach (SourceResponse sourceResponse in _searchResponse.Responses)
        {
            sourceResults = sourceResponse.Results;

            foreach (Result sourceResult in sourceResults)
            {
                // Met les résultat dans mon propre objet
                LiveSearch.Resultat _resultat = new LiveSearch.Resultat();

                if ((sourceResult.Title != null) && (sourceResult.Title != String.Empty))
                    _resultat.Titre = sourceResult.Title;

                if ((sourceResult.Description != null) && (sourceResult.Description != String.Empty))
                    _resultat.Description = sourceResult.Description;

                if ((sourceResult.Url != null) && (sourceResult.Url != String.Empty))
                    _resultat.Url = sourceResult.Url;
            }
        }
    }
}
```

```

        v_result.Add(_resultat);
    }
}
}
catch (Exception et)
{
}
finally
{
    sourceResults = null;
    _searchResponse = null;
    _searchRequest = null;
    _search = null;
}
return v_result;
}
}

```

Le code-behind de votre page

Il est nécessaire de référencer le script Javascript de Google qui contient l'API Ajax. Il est bien entendu possible de le mettre en dur dans le code de la manière suivante :

```

<asp:ScriptManager ID="ScriptManager1" runat="server">
  <Scripts>
    <asp:ScriptReference Path="http://www.google.com/uds/api?file=uds.js&v=1.0&key=maclé" />
  </Scripts>
</asp:ScriptManager>

```

La même chose, en code managé :

```

ScriptReference SRef = new ScriptReference();
SRef.Path = "http://www.google.com/uds/api?file=uds.js&v=1.0&key=" +
ConfigurationManager.AppSettings["GoogleApiKey"];
ScriptManager1.Scripts.Add(SRef);

```

On fournira également deux variables pour fournir au code Javascript le domaine de filtrage et le nom du site Web.

```

// Filtrage du site Web au niveau de la recherche
_restrictedDomain = ConfigurationManager.AppSettings["SiteWeb"];
// Nom du site Web
_siteName = ConfigurationManager.AppSettings["SiteNom"];

```

Le Javascript client

Avant de regarder en détail le javascript, attardons nous sur les div nécessaires pour la construction et la restitution des données :

```

<div class="titre">Rechercher sur ce site</div><br />
<div id="container_search_controls">
  <div id="google_search_form"></div>
  <div id="google_search_control"></div>
  <div id="live_search_container">
    <div id="live_search_query"></div>
    <div id="live_search_control"></div>
  </div>
</div>

```

- google_search_for disposera de l'input de saisie et du bouton
- google_search_control accueillera les résultats de la recherche
- live_search_query sera initialisé avec le résumé de la recherche sur live.com
- live_search_control comprendra les résultats obtenus sur live.com

On initialise deux variables issues du Code behind avec le domaine de recherche et le nom du site. Ma fonction LoadGoogleSearch crée une nouvelle instance de mon MyGoogleSearch qui construit l'interface. LoadGoogleSearch sera lancée plus loin.

```

var _searchDomain = '<%= _restrictedDomain %>';

```

```

var _siteName = '<%= _siteName %>';

function LoadGoogleSearch()
{
    new MyGoogleSearch();
}

function MyGoogleSearch()
{
    // Formulaire de saisie
    var sFormDiv = document.getElementById("google_search_form");
    // Zone de résultat
    var resultDiv = document.getElementById("google_search_control");

    this.searchControl = new GSearchControl();
    this.searchForm = new GSearchForm(true, sFormDiv);

    // Sur le submit
    this.searchForm.setOnSubmitCallback(this, MyGoogleSearch.prototype.onSubmit);
    this.searchForm.setOnClearCallback(this, MyGoogleSearch.prototype.onClear);

    var searcher = new GwebSearch();
    var options = new GsearcherOptions();

    // Limite la recherche au domaine
    searcher.setSiteRestriction(_searchDomain);

    // Personnalise le label de résultat
    searcher.setUserDefinedLabel(_siteName);

    // Ouvre le résultat
    options.setExpandMode(GSearchControl.EXPAND_MODE_OPEN);
    // Résultat long
    this.searchControl.setResultSetSize(GSearch.LARGE_RESULTSET);
    // Liens vers une nouvelle page
    this.searchControl.setLinkTarget(GSearch.LINK_TARGET_BLANK);
    // Ajoute le Searcher
    this.searchControl.addSearcher(searcher, options);

    // Construit l'interface
    this.searchControl.draw(resultDiv);
}

```

La fonction interceptant le submit de la recherche Google lance donc ma requête sur Google et sur Live.com grâce à la fonction LiveSearchBuilder en prenant soin de récupérer la chaîne saisie. Le rendu des réponses de Google est fait automatiquement par l'API Ajax de ce dernier.

```

MyGoogleSearch.prototype.onSubmit = function(form) {
    var query = form.input.value;
    if (query && query != "")
    {
        // Exécute la recherche
        this.searchControl.execute(query);
        // Lance la recherche Live
        LiveSearchBuilder(query);
    }
    return false;
}

```

Ma fonction LiveSearchBuilder() nettoie d'abord un éventuel résultat précédent. On utilisera ensuite le Framework Ajax pour consommer notre Webservice qui récupère la réponse dans OnMsnLiveSucceeded().

```

function LiveSearchBuilder(p_query)
{
    var _objC = document.getElementById("live_search_query");
}

```

```

_objC.style.visibility = "visible";

var _obj = document.getElementById("live_search_control");

if (_obj != null)
{
    // Efface le résultat précédent
    if (_obj.childNodes.length > 0)
    {
        for(x=0; x < _obj.childNodes.length; x++)
        {
            _obj.removeChild(_obj.childNodes[x]);
        }
    }
}

// Invoque le Webservice
Sys.Net.WebServiceProxy.invoke("webservice.asmx",
    "MsnLiveSearch",
    false,
    { p_search : p_query},
    OnMsnLiveSucceeded,
    OnMsnLiveFailed,
    null);
}

```

La fonction OnMsnLiveSucceeded() récupère la réponse du Webservice dans result et fait le rendu dans l'interface.

```

// Si la requete à fonctionné
function OnMsnLiveSucceeded(result, eventArgs)
{
    // Affiche les résultats
    var out = document.getElementById("live_search_control");

    var i = 0;
    if (result.length > 0)
    {
        var _output = document.createElement("div");

        if (result.length > 1)
            document.getElementById("live_search_query").innerHTML = result.length + " résultats sur LIVE.COM";
        else
            document.getElementById("live_search_query").innerHTML = result.length + " résultat sur LIVE.COM";

        for(i=0; i < result.length; i++)
        {
            var _link = document.createElement("a");
            _link.href = result[i].Url;
            _link.target = "_blank";
            _link.innerHTML = result[i].Titre;

            _output.appendChild(_link);

            var _desc = document.createElement("div");
            _desc.className = "live_description";
            _desc.innerHTML = result[i].Description;

            _output.appendChild(_desc);
        }
        out.appendChild(_output);
    }
    else
    {
        document.getElementById("live_search_query").innerHTML = "Aucun résultat sur LIVE.COM";
    }
}
}

```

Enfin, ce morceau de code construit la zone de recherche :

```
GSearch.setOnLoadCallback(LoadGoogleSearch);
```

Ne reste plus qu'a lancer votre page default.aspx...

Pour en savoir plus

- Les sources du tutorial : <http://www.laurentgeffroy.com/specific/datas/tutoriels/LiveGoogleSearch/MashUpLiveGoogleSearch.zip>
- L'API Google Search : <http://code.google.com/apis/ajaxsearch/>
- Le SDK Live.com : <http://msdn2.microsoft.com/en-us/library/bb251794.aspx>
- Gestion des AppKeys MSN : <http://search.msn.com/developer>
- La documentation du Framework Ajax 1.0 pour ASP.Net 2.0 : <http://ajax.asp.net/docs/>
- Me contacter : lgeffroy at gmail.com