Tutoriel : Accès à un Service Web (GoogleSearch API) avec Visual Basic .Net 2003

Table des matières

Introduction

Nous allons, par l'intermédiaire de ce tutoriel, apprendre à utiliser les WebServices. Nous prendrons comme exemple l'utilisation de l'API Google API. Cette API va vous permettre de réaliser des opérations de recherche sur le Web. Nous apprendrons à utiliser le standard WSDL, qui permet de dialoguer avec des services Web existants en mettant à la disposition du développeur tout le nécessaire (méthodes, paramètres et propriétés utilisables) par l'intermédiaire de messages SOAP qui sont au format XML.

Qu'est-ce qu'un Service Web ???

On pourrait dire grosso modo que c'est un objet sans interface visuel, en fait une fonction qui renvoie un résultat en réponse à une requête de l'application cliente. Dans le cas qui nous intéresse ici, c'est-à-dire Google, ce service va nous renvoyer toutes les pages internet répondant à la saisie des mots clés que l'on aura demandé dans la requête.
C'est à l'application cliente ensuite d'organiser les données reçues suivant la charte graphique de l'application.

Les préliminaires

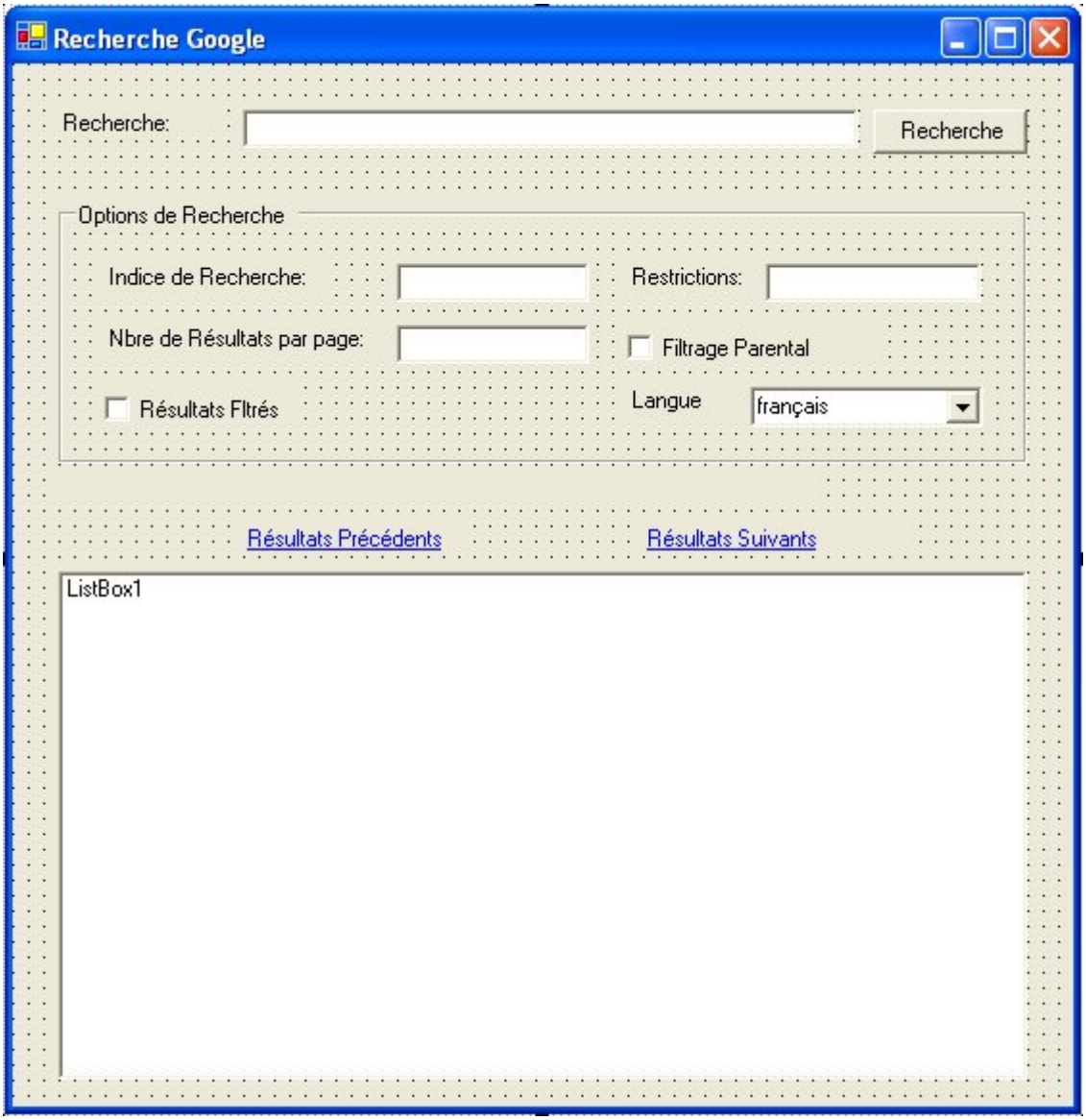
Avant de pouvoir utiliser ce service, il va falloir dans un premier temps télécharger l'API, pour ce faire vous devez vous rendre à l'adresse suivante : http://www.google.fr/apis/

Une fois installer sur votre ordinateur, il vous reste à accomplir une tâche primordiale, si vous voulez utiliser ce service : l'enregistrement. En effet sans cette étape vous ne pourrez pas accédez à ce service. Une fois l'enregistrement effectué, vous allez recevoir une clé d'utilisation, il faut la garder précieusement, vous en aurez besoin pour la suite, pour vous connecter au service. Vous aurez le droit ainsi à 1000 requêtes par jour. Voilà maintenant vous êtes prêt à entrer dans le vif du sujet.

Le design de la fiche de Recherche Google

Nous allons débuter un nouveau projet fondé sur une application Windows. Pour ce faire dans le menu Fichier->Nouveau->Projet->Projets Visual Basic, cliquer sur l'icône Application Windows. La fenêtre du projet s'ouvre avec la fiche Form1 en visualisation directe. Le premier travail consiste à élaborer le design de la fenêtre principale.

Ensuite réaliser l'interface de façon à ce qu'elle ressemble à celle-ci :



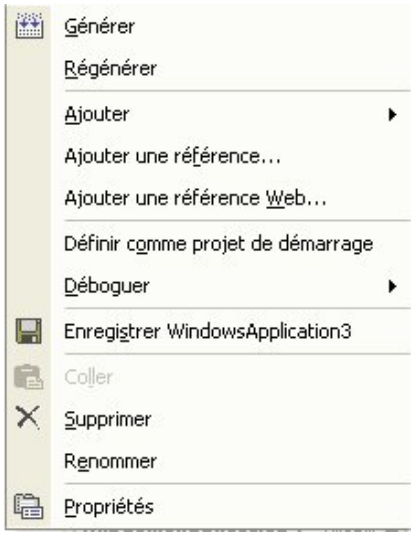
Pour pouvoir aborder le plus facilement possible la suite de ce tutoriel nous allons utiliser les noms suivants pour les composants de cette fiche :

- txtRecherche qui correspond à la zone d'édition (utiliser un composant TextBox) permettant de renseigner la recherche sous forme de mots clé.
- btnRecherche qui va permettre de lancer la recherche (utiliser un composant Button).
- txtIndice qui correspond à la zone de saisie de l'indice de début de recherche (utiliser aussi un composant TextBox).
- txtNbreResu qui correspond au nombre de résultats que l'on veut voir apparaître sur une page. Sachant que celui-ci ne peut pas dépasser la valeur 10 (utiliser de même un composant TextBox).
- cbxResuFiltres qui permet de filtrés les résultats trouvées et d'afficher ou non des doublons (utiliser une CheckBox).
- txtRestriction qui correspond à une option permettant de restreindre le nombre de pages trouvées en fonctions d'un pays (.fr, .eu, .en) ou en fonction de la langue voulue (une TextBox).

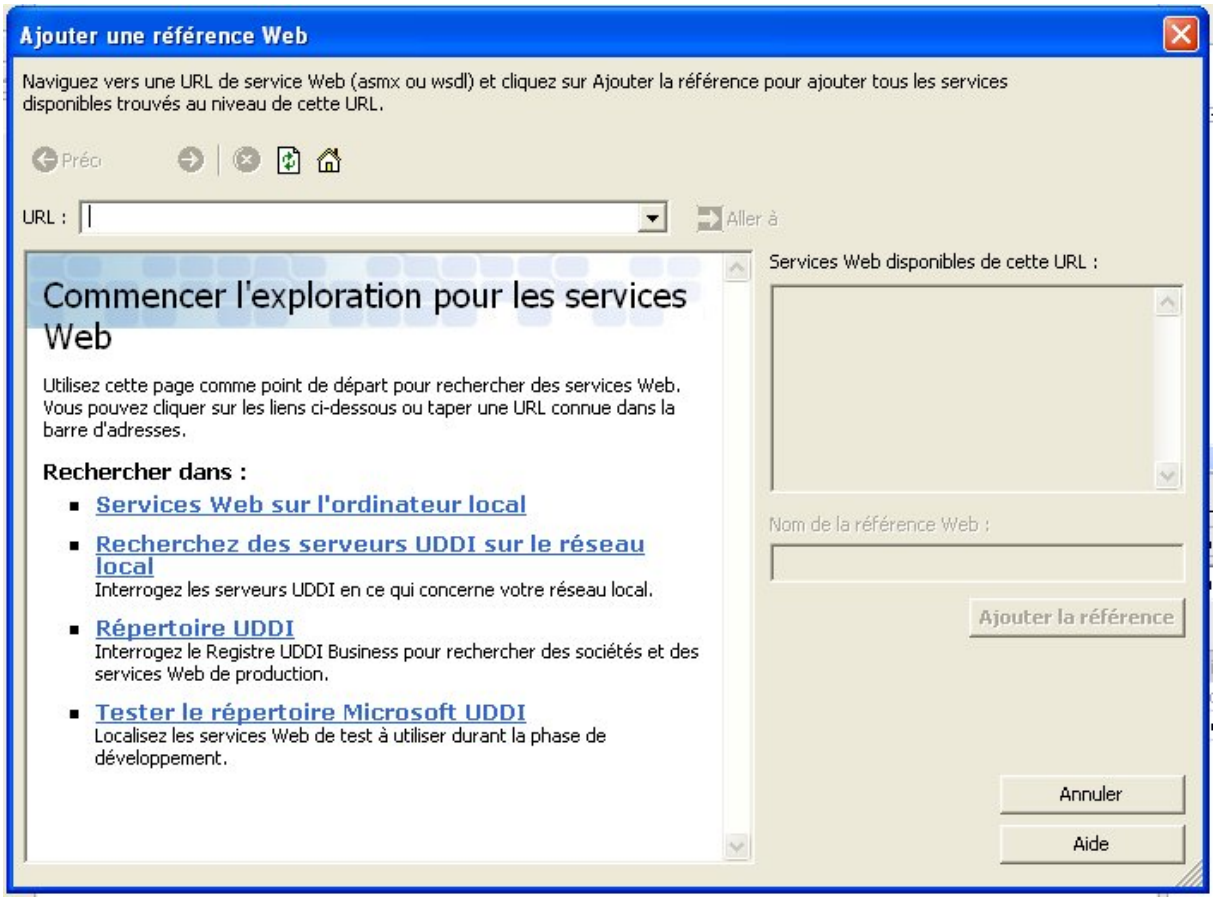
- cbxFiltrageParental : permet de filtrer les pages trouvées et d'afficher ou non des pages comportant des documents destinées aux grandes personnes (utiliser un composant CheckBox).
- cbxLangue qui permet de réaliser un filtrage sur les résultats obtenus afin de n'afficher que les pages écrites dans la langue voulue (utiliser un ComboBox).
- lblResu qui va nous permettre d'afficher le nombre de pages trouvées (utiliser un Label).
- lnkResuPrecedents qui est un bouton qui va permettre de revoir les pages que l'on a vu précédemment (utiliser un linklabel).
- lnkResuSuivants qui est un bouton qui va permettre de voir les pages suivantes (utiliser un linklabel).
- Et pour finir un ListBox qui va contenir les pages résultantes de la requête.

Importation du WebService via l'importateur WSDL

Ensuite il faut importer le service Web de Google vers Visual basic .Net 2003, pour pouvoir l'utiliser. Pour ce faire dans la fenêtre explorateur de solutions (situé à droite de la fenêtre de design et de code), faites un clic droit sur le nom du projet. Un menu contextuel apparaît :

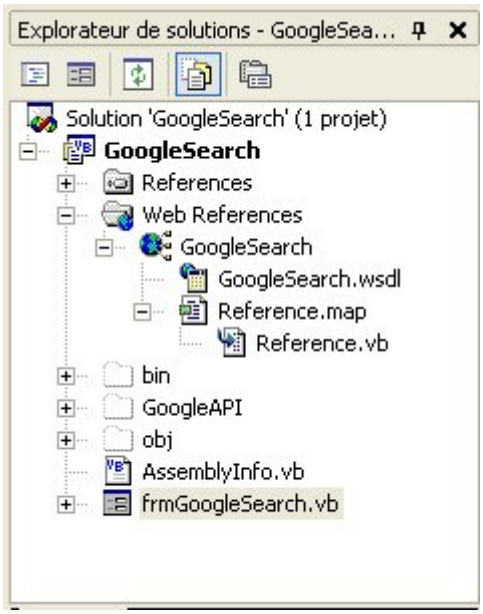


Vous devez choisir la commande « Ajouter une référence Web… ». Une fois sélectionnée une autre fenêtre s'affiche à l'écran :



Dans la zone de texte correspondant à l'URL, vous devez renseigner le répertoire dans lequel vous avez installé les fichiers de l'API (pour moi il s'agit de C:\GoogleAPI\GoogleSearch.wsdl). Ensuite cliquer sur le bouton « Aller à » et vous verrez apparaître dans la fenêtre ceci:

Voici les fonctions mises à disposition par Google que l'on va pouvoir implémenter dans notre application. En ce qui concerne ce tutoriel nous allons nous pencher plus précisément sur la fonction doGoogleSearch().
Il ne reste plus qu'à valider l'utilisation de ce service Web, en cliquant sur le bouton « Ajouter la référence ». Une fois ceci effectué, vous devez voir apparaître des changements au niveau de la fenêtre d'exploration de la solution :



Maintenant, que nous avons importé notre WebService, nous allons pouvoir entrer dans le vif du sujet, c'est-à-dire le codage de notre fiche de recherche Google afin que l'on puisse récupérer des données via la Google Search API.

Accès au service - Préliminaires

Nous avons tout le nécessaire pour pouvoir entrer en contact avec le WebService. Pour commencer vous devez mettre la référence GoogleSearch dans la clause imports de votre fiche principale :

```
Imports WindowsApplication2.GoogleSearch.GoogleSearchResult
```

Ensuite vérifier que vous vous êtes bien enregistrer auprès de Google, afin de pouvoir utiliser le WebService, vous devriez avoir reçu une clé en retour, vous allez déclarez une constante clé de la façon suivante :

```
Public Const Cle As String = "votre clé"
```

Une fois ceci effectué nous allons pouvoir initialisé l'accès proprement dit au Service, pour ce faire vous allez avoir besoin de réaliser les déclarations suivantes :

```
Dim diSearchGoogleService As New GoogleSearch.GoogleSearchService
Dim ResultatRecherche As New GoogleSearch.GoogleSearchResult
Dim ElementsRecherche As New GoogleSearch.ResultElement
```

La première déclaration permet d'initialiser une interface qui va nous être utile afin de dialoguer avec le Webservice Google.
La deuxième déclaration va nous permettre d'initialiser un objet qui va contenir les différentes propriétés de retour comme le nombre de pages estimmées, l'indice de début, l'indice de fin de recherche.

La troisième déclaration va nous permettre de pointer sur le contenu d'une page résultante de la rehcerche, on va pouvoir en extraire des propriétés intéressantes comme l'adresse (l'url), le résumé, le titre, le domaine, etc…

Maintenant que nous avons vu les préliminaires, nous allons pouvoir passer à l'accès au WebService et au traitement des résultats retournés.

Accès au Service – Traitement de la requête de l'utilisateur

Pour effectuer la recherche nous allons créer une procédure qui va nous permettre de réaliser cette opération :

```
Private Sub Recherche()
        Dim i As Integer


        ListBox1.Items.Clear()
        Requete = txtRequete.Text
        If cbxLangue.Text = "français" Then
            sChoixLangue = "lang_fr"
        Else
            sChoixLangue = "lang_en"
        End If


        ResultatRecherche = diSearchGoogleService.doGoogleSearch(Cle,
Requete, Int(txtIndice.Text) + (n * iMaxResu), Int(txtNbreResu.Text),
cbxResuFiltres.Checked, txtRestrictions.Text, cbxFiltrageParental.Checked,
sChoixLangue, "", "")
        iNbreResu = ResultatRecherche.estimatedTotalResultsCount
        lblResu.Text = "Résultat de la Recherche : " +
Str(ResultatRecherche.estimatedTotalResultsCount) + " pages trouvées"
        For i = ResultatRecherche.startIndex - iIndiceDebut To
ResultatRecherche.endIndex - iIndiceDebut
            ElementsRecherche = ResultatRecherche.resultElements(i - 1)
            ListBox1.Items.Add(ElementsRecherche.title)
            ListBox1.Items.Add(ElementsRecherche.snippet)
            ListBox1.Items.Add(ElementsRecherche.URL)
            ListBox1.Items.Add(ElementsRecherche.hostName)
            ListBox1.Items.Add("")


        Next
        If iNbreResu > iMaxResu Then

            lnkResuPrecedents.Visible = True
            lnkResuSuivants.Visible = True

        End If

    End Sub
```

Cette fonction peut être découpé en deux parties bien distinctes l'envoi de la requête et la réception des résultats en retour, et puis bien sûr le traitement de ceux-ci.

L'envoi de la requête et la réception des données retournées s'effectuent par l'intermédiaire de la fonction doGoogleSearch de l'interface GoogleSearchService. Cette fonction prend en paramètre un certain nombre d'options qui sont les suivantes :

- Le premier élément es indispensable et correspond à la clef personnelle d'accès au service. Il faut donc passer en paramètre la constante cle que nous venons de définir.
- Le deuxième élément constitue la demande de l'utilisateur proprement dite. C'est le même principe d'utilisation que sur le site.
- Le troisième élément est l'indice de l'élément recherché à partir duquel on va afficher les résultats. Il faut savoir que le nombre de résultats affichables sur une page (par requête) est limité à 10. ce qui veut dire que si la recherche retourne plus de 10 éléments, on va pouvoir sélectionner les éléments de 1 à 10 par exemple, ou de 91 à 100. Chaque nouvelle recherche fait décompter le nombre de requête qui je vous le rappelle est limitée à 1000/jour.
- Le quatrième élément correspond au nombre d'éléments affichables sur la page (celui est limité à 10).
- Le cinquième élément permet de demander au service de filtrer ou non les éléments.
- Le sixième élément permet de définir une option de restriction dans la recherche, ceci peut être une langue ou un nom de domaine par exemple.
- Le septième élément permet d'interdire ou non les sites pour adultes, indispensable en cas d'utilisation des enfants.
- Le paramètre suivant permet de filtrer les résultats en fonction de la langue des pages. Pour renseigner ce paramètre il faut utiliser, par exemple lang_fr pour afficher les pages écrites en français, ou encore lang_en pour celles écrites en anglais.
- Les deux autres paramètres ne sont plus utilisés et ne présentent pas d'intérêt. Vous pouvez remplacer ces deux paramètres par 'dummy' . En effet les options d'encodage se font maintenant par l'intermédiaire de la norme UtF-8.

Si l'opération s'est bien déroulée, le service nous renvoie alors un tableau associatif (grâce à NUSoap). C'est à nous ensuite de lire les informations contenues dans celui-ci.
Pour accéder à ce tableau on va utiliser la variable RechercheResultat que nous avons défini précédemment, et l'objet ElementsRecherche de type ResultElement.

Il est donc ainsi très aisé de pouvoir accéder à des propriétés comme le titre de la page (ElementsRecherche.title), le résumé de cette page (ElementsRecherhe.snippet), le nom de domaine (ElementsRecherche.hostName), ou encore l'adresse de la page (ElementsRecherche.URL). Pour pouvoir traiter les résultats un par un on va utiliser une boucle classique for … to … do. On utilise les propriétés startIndex et endIndex pour pouvoir définir l'indice du résultat à traiter. Voilà maintenant vous avez toues les informations possibles pour pouvoir effectuer des recherches avec mots clés.

Il nous reste à voir le fonctionnement des deux liens Résultats Suivants et résultats Précédents(voir page suivante) .

Comme vous pouvez le constater le code correspondant à l'événement OnClick de ces deux objets est très simple, on va tester que la recherche est possible en testant si on va dépasser le nombre total de résultats ou obtenir un nombre négatif. Ici n correspond au nombre de fois. Bien sûr avant d'effectuer une nouvelle recherche, on va réinitialiser notre ListBox1 en effaçant son contenu et on va rappeler la fonction que l'on a étudier en détail précédemment.

```
Private Sub lnkResuPrecedents_LinkClicked(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
lnkResuPrecedents.LinkClicked
        If ((iIndiceDebut - iMaxResu) < iNbreResu) And ((iIndiceDebut -
iMaxResu) >= 0) Then

            iIndiceDebut = iIndiceDebut - iMaxResu
            n = n - 1
        End If
        ListBox1.Items.Clear()
        Recherche()


    End Sub

    Private Sub lnkResuSuivants_LinkClicked(ByVal sender As Object, ByVal e
As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
lnkResuSuivants.LinkClicked
        If (iIndiceDebut + iMaxResu) < iNbreResu Then

            iIndiceDebut = iIndiceDebut + iMaxResu
            n = n + 1
        End If
        ListBox1.Items.Clear()
        Recherche()
    End Sub
```

Pour info je vous mets le contenu de la procédure Load associée à la fiche :

```
Private Sub frmRecherche_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        txtIndice.Text = "0"
        txtRestrictions.Text = ""
        txtNbreResu.Text = "10"
        iMaxResu = Int(txtNbreResu.Text)
        lnkResuPrecedents.Visible = False
        lnkResuSuivants.Visible = False

    End Sub
```

Donc comme vous pouvez le constater on va initialiser les variables que l'on va utiliser par la suite dans la procédure de Recherche ensuite.

Conclusion

Vous avez pu apprendre par l'intermédiaire de ce tutoriel comment on communique avec WebService. Même si l'on a pris pour exemple, l'API de Google, vous allez pouvoir utiliser l'importateur WSDL pour n'importe quel Service qui est compatible avec cette norme bien entendu. Vous avez pu constater que l'environnement de programmation Visual Studio .Net 2003 nous facilite grandement la tâche grâce à son Imporateur WSDL. On peut dire aussi que les sevices Web même s'ils fournissent des outils de hautes qualités souffrent d'un défaut majeur, il faut avoir une connexion permanente avec Internet lors de la requête. Comme vous avez pu le constater, ce tutoriel ne vous a permis que d'implémenter la fonction doGoogleSearch(), je vous laisse le soin de peaufiner ceette application afin de pouvoir implémenter les deux autres fonctions mises à votre disposition par l'API Google Search.