

Tutoriel : Accès à un Service Web (Google API) avec Delphi 2005

Table des matières

<u>INTRODUCTION</u>	2
<u>QU'EST-CE QU'UN SERVICE WEB ???</u>	2
<u>LES PRELIMINAIRES</u>	2
<u>LE DESIGN DE LA FICHE DE RECHERCHE GOOGLE</u>	2
<u>IMPORTATION DU WEBSERVICE VIA L'IMPORTATEUR WSDL</u>	4
<u>ACCES AU SERVICE - PRELIMINAIRES</u>	5
<u>ACCES AU SERVICE – TRAITEMENT DE LA REQUETE DE L'UTILISATEUR</u>	6
<u>CONCLUSION</u>	8

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

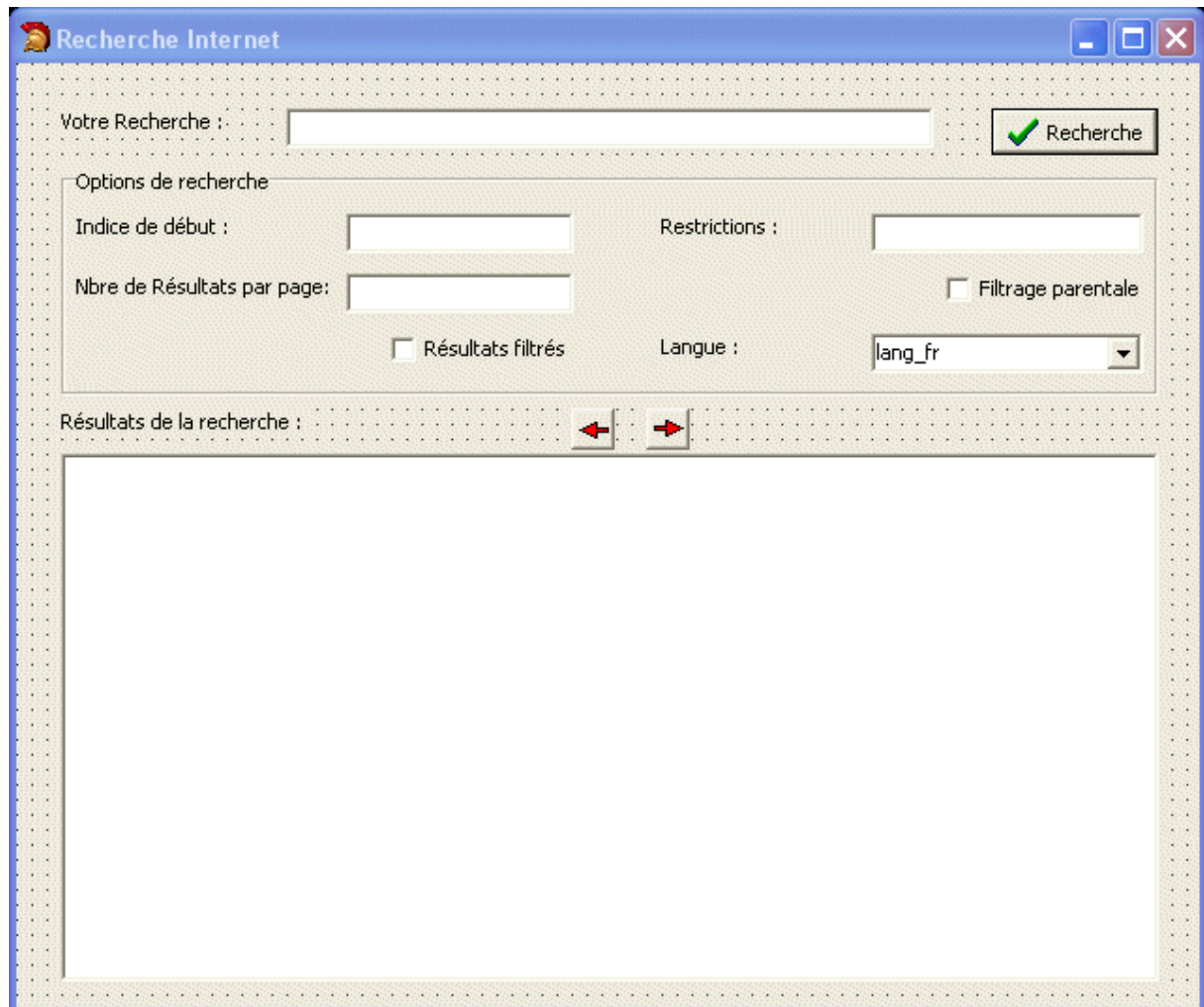
Avant de pouvoir utiliser ce service, il va falloir dans un premier temps télécharger l'API, pour ce faire vous devez vous rendre à l'adresse suivante : <http://www.google.fr/apis/>

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 accéder 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 débiter un nouveau projet basé sur une fiche Win32. Pour ce faire dans le menu Fichier->Nouveau Sélectionner Fiche – Delphi pour Win32.

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



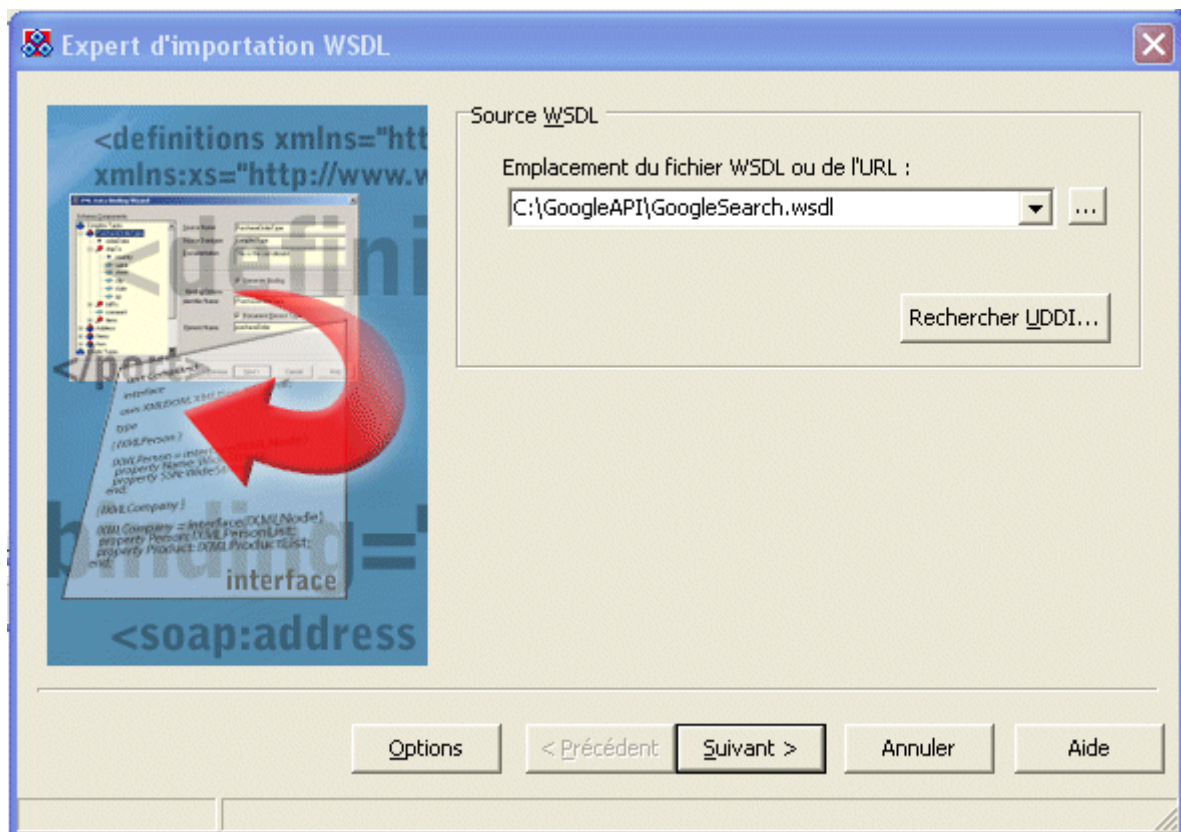
Pour pouvoir aborder le plus facilement possible la suite de ce tutoriel nous allons utiliser les noms suivants pour les composants de cette fiche :

- txtRecherche qui correspond à la zone d'édition (utiliser un composant TEdit) permettant de renseigner la recherche sous forme de mots clé.
- btnRecherche qui va permettre de lancer la recherche (utiliser un TBitBtn).
- txtIndex qui correspond à la zone de saisie de l'indice de début de recherche (utiliser aussi un composant TEdit).
- 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 TEdit).
- cbxResuFiltres qui permet de filtrés les résultats trouvées et d'afficher ou non des doublons (utiliser une TCheckBox).
- 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.
- cbxControleParental : permet de filtrer les pages trouvées et d'afficher ou non des pages comportant des documents destinées aux grandes personnes (utiliser un composant TCheckBox).
- cbxChoixLangue qui permet de réaliser un filtrage sur les résultats obtenus afin de n'afficher que les pages écrites dans la langue voulue (utiliser un TComboBox).

- lblResultats qui va nous permettre d'afficher le nombre de pages trouvées (utiliser un TLabel).
- btnPagePrecedente qui est un bouton qui va permettre de revoir les pages que l'on a vu précédemment (utiliser un TSpeedButton).
- btnPageSuivante qui est un bouton qui va permettre de voir les pages suivantes.
- Et pour finir un TListBox qui va contenir les pages résultantes de la requête.

Importation du Webservice via l'importateur WSDL

Ensuite il faut utiliser l'importateur WSDL fournit avec Delphi, pour pouvoir importer le Webservice. Pour ce faire Cliquer sur Fichier->Nouveau->Autre vous allez voir apparaître la fenêtre suivante :



Dans Emplacement de Fichier WSDL ou de l'URL vous devez renseigner le répertoire dans lequel vous avez installé les fichiers de l'API. Ensuite cliquer sur le bouton Suivant et vous verrez apparaître dans la fenêtre de code ceci:

```
1 // ***** //
2 // Les types dclars dans ce fichier ont t gnrs partir de donnes lues dans le
3 // fichier WSDL dcrit ci-dessous :
4 // WSDL : C:\GoogleAPI\GoogleSearch.wsdl
5 // Version : 1.0
6 // (08/08/2005 15:16:37 - 1.33.2.5)
7 // ***** //
8
9 unit GoogleSearch;
10
11 interface
12
13 uses InvokeRegistry, SOAPHTTPClient, Types, XSBuiltIns;
14
15 type
16
17 // *****
18 // Les types suivants mentionns dans le document WSDL ne sont pas reprsents
19 // dans ce fichier. Ce sont soit des alias[@] de types reprsents ou alors il
20 // rfrens mais jamais[!] dclars dans le document. Les types de la dernire c
21 // sont en principe mapps des types Borland ou XML prdfinis/connus. Toutefc
22 // signaler des documents WSDL incorrects n'ayant pas russi dclarer ou impc
23 // *****
```

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 uses de votre fiche principale :

```
Uses GoogleSearch ;
```

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

```
Const cle = 'votre clé' ;
```

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

```
Var _di_SearchGooglePort : SearchGooglePort ;
Var ResultatRecherche : GoogleSearchResult ;
```

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.

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 :

```
procedure TfrmRecherche.Recherche;
var
  i : integer;

begin
  _di_GoogleSearchPort := GetGoogleSearchPort;
  ResultatRecherche := GoogleSearchResult.Create;

  ResultatRecherche := _di_GoogleSearchPort.doGoogleSearch(Cle,
    txtRecherche.Text,
    StrToInt(txtIndex.Text)+(n*MaxRes),
    StrToInt(txtNbreResu.Text),
    cbxResuFitres.Checked,
    txtRestriction.Text,
    cbxFiltrageParental.Checked,
    cbxChoixLangue.Text,
    'dummy',
    'dummy'
  );
  lblResultats.Caption := 'Résultat de la Recherche : ' + IntToStr(ResultatRecherche.estimatedTotalResultsCount)
  + ' pages trouvées';
  iNbreResu := ResultatRecherche.estimatedTotalResultsCount;

  for i := ResultatRecherche.startIndex- indexdebut to ResultatRecherche.endIndex - indexdebut do
  begin

    ElementsRecherche := ResultatRecherche.ResultElements[i-1];
    ListBox1.Items.Add(IntToStr(i+(n*MaxRes)));
    if ElementsRecherche.title <> "" then
      ListBox1.Items.Add(ElementsRecherche.title);
    if ElementsRecherche.snippet <> "" then
      ListBox1.Items.add(ElementsRecherche.snippet);

    ListBox1.Items.Add(ElementsRecherche.URL);
    ListBox1.Items.Add(ElementsRecherche.hostName);
    ListBox1.Items.Add("");
  end;

  if iNbreResu > MaxRes then
  begin
```

```
btnPagePrecedente.Visible := True;  
btnPageSuivante.Visible := True;  
end;  
end;
```

Cette fonction peut être découpée 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 GoogleSearchPort. Cette fonction prend en paramètre un certain nombre d'options qui sont les suivantes :

- Le premier élément est 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 (ElementsRecherche.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 toutes les informations possibles pour pouvoir effectuer des recherches avec mots clés.

Il nous reste à voir le fonctionnement des deux boutons Page Suivante et Page Précédente, dont voici le code :

```

procedure TfrmRecherche.btnPageSuivanteClick(Sender: TObject);
begin
if (indexdebut +MaxRes) < inbreResu then
begin
indexdebut := indexdebut + MaxRes;
inc(n);
end;
ListBox1.Items.Clear;
Recherche;
end;

procedure TfrmRecherche.btnPagePrecedenteClick(Sender: TObject);
begin
if (indexdebut - MaxRes) > 0 then
begin
indexdebut := indexdebut - MaxRes;
dec(n);
end;
ListBox1.Items.Clear;
Recherche;
end;

```

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 étudié en détail précédemment.

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

```

procedure TfrmRecherche.FormCreate(Sender: TObject);
begin
txtIndex.Text := '0';
txtNbreResu.Text := '10';
indexdebut := StrToInt(txtIndex.Text);
n := 0;
MaxRes := StrToInt(txtNbreResu.Text);
btnPageSuivante.Visible := False;
btnPagePrecedente.Visible := False;

end;

```

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 Delphi nous facilite grandement la tâche grâce à son Importateur WSDL. On peut dire aussi que les services 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.