

# SUPPORT WINDEV NUMERO 7.1



21/08/2015

## Consommation d'un Webservice

L'objectif de ce support est de vous faire intégrer et consommer un Webservice existant

# Support WinDev Numéro 7.1

## VERSION 19

### CONSOMMATION D'UN WEBSERVICE

WinDev, WebDev et WinDev Mobile permettent d'importer directement des WebServices dans vos applications.

Un Webservice est un ensemble de points d'entrées mis à la disposition des utilisateurs afin d'effectuer différents traitements. Par exemple, un service d'accès distant met à votre disposition les traitements d'accès aux données. Les échanges d'informations avec un Webservice se font au format XML et utilisent les protocoles SOAP (Simple Object Access Protocol) et HTTP.

A partir de la description au format WSDL (Web Services Description Language) de ce service, WinDev, WebDev ou WinDev Mobile va générer automatiquement des types et des fonctions WLangage correspondants à l'interface de programmation du Webservice.

Pour utiliser le Webservice, il suffit d'utiliser les fonctions générées au moment de l'import.

### PRESENTATION DU TP :

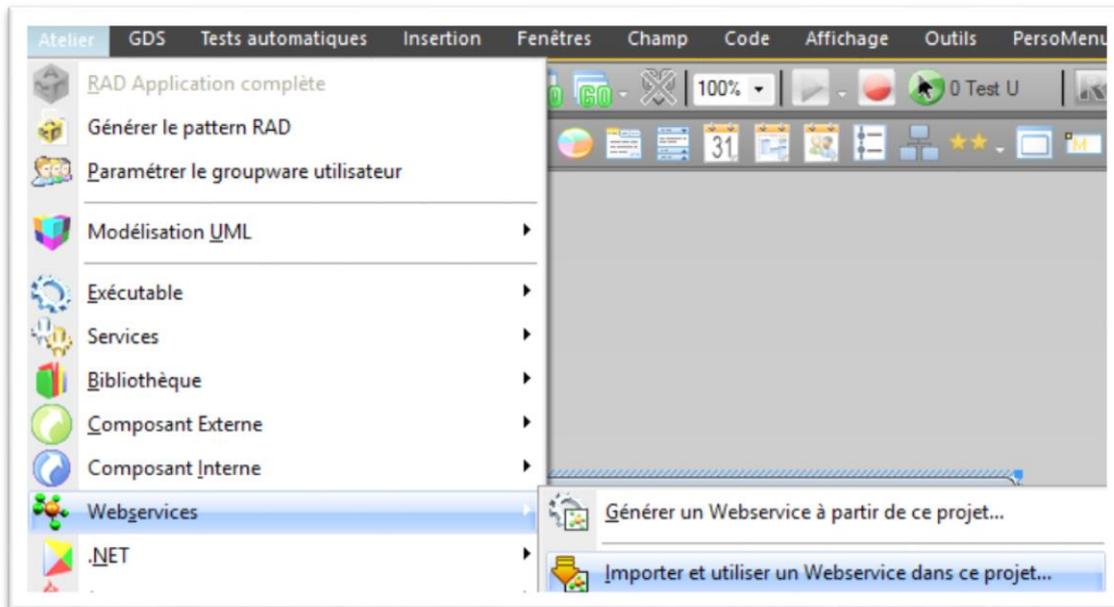
Nous allons utiliser un WebService existant qui nous donne la météo d'une ville. Le Webservice se base sur les informations météorologiques envoyées par les stations météo des aéroports. Donc nous n'aurons que la météo des villes disposant d'un aéroport.

#### Création d'un nouveau projet.

Créez un nouveau projet nommé **WebServiceMeteo**.

Utilisez les choix par défaut et n'associez pas d'analyse à ce projet.

Pour importer le WebService, il nous faut aller dans le menu **Atelier**, puis **WebService**, puis **Importer et utiliser un Webservice dans ce projet...**



L'assistant d'importation se lance, allez à l'étape 2 de l'assistant sur la description WSDL du Webservice. Saisissez les informations comme ci-dessous :

## Description WSDL du Webservice

Localisez la description WSDL du Webservice :

charger la description à l'adresse suivante :

<http://www.webservices.net/globalweather.asmx?WSDL>

Si l'accès à la description du Webservice (WSDL) nécessite une authentification, indiquez le nom d'utilisateur et le mot de passe ci-dessous :

Nom d'utilisateur :

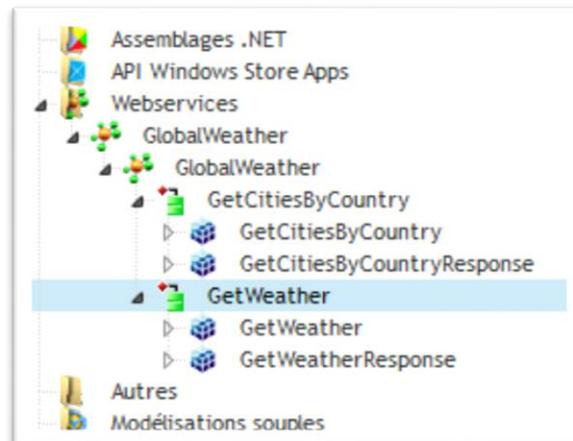
Mot de passe :

le service est décrit dans le fichier suivant :

...

Validez l'assistant, un message vous indiquera que l'importation est terminée.

Le Webservice est maintenant intégré dans Windev. Pour le visualiser regardez dans la rubrique WebServices de l'explorateur de projet.

**La structure est constituée de :**

- le nom du Webservice (GlobalWeather dans cet exemple).
- le nom de chaque fonction (GetCitiesByCountry et GetWeather dans cet exemple).
- le nom d'une structure contenant soit les paramètres d'appel (GetWeather dans cet exemple), soit les valeurs de retour (GetWeatherResponse dans cet exemple).
- le nom de chaque variable de la structure (CountryName et CityName dans cet exemple).

Dans notre exemple, le Webservice importé se nomme : GlobalWeather. Il contient 2 fonctions :

- GetCitiesByCountry
- GetWeather

La fonction GetCitiesByCountry contient 2 structures :

- GetCityByCountry
- GetCityByCountryResponse

La fonction GetWeather contient 2 structures :

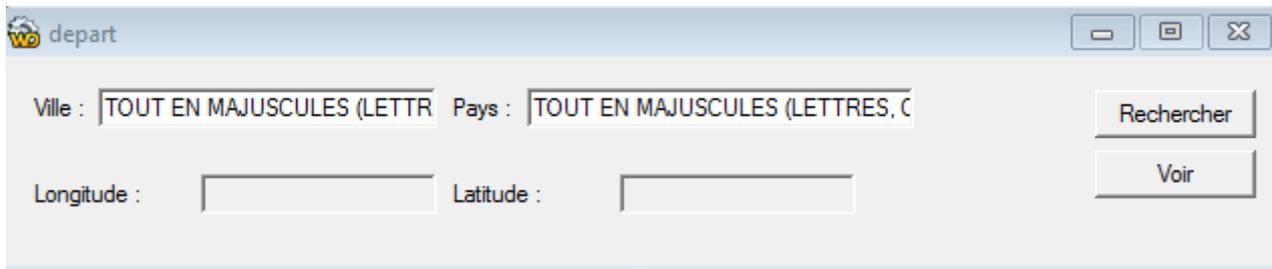
- GetWeather
- GetWeatherResponse

La structure GetWeather contient 2 variables (CityName et CountryName) correspondant aux valeurs à passer en paramètre pour l'appel de la fonction GetWeather.

La structure GetWeatherResponse contient une variable (GetWeatherResult) correspondant à la valeur de retour de la fonction.

## Utilisation du Webservice

Créez une fenêtre nommée **Départ**.



1. Placez 2 champs de saisie en édition Ville et Pays, faite en sorte que les champs soient **Tout en majuscules**.
2. Placez 2 champs de saisie en affichage seulement : Longitude et Latitude
3. Placez enfin 2 boutons tels que présenté ci-dessus.

L'utilisateur saisira la ville souhaitée et son pays, le code du bouton **Rechercher** activera le Webservice et retournera la longitude et la latitude.

### Code du bouton Rechercher

v1 est un GetWeather

v2 est un GetWeatherResponse

meteo est une chaîne

location est une chaîne

n est un entier

longi, lati sont des chaînes

lodegre,ladegre sont des chaînes

// on affecte les valeurs saisies aux variables du Webservice

v1.CountryName = Pays

v1.CityName = Ville

v2 = GetWeather(v1)

// La chaîne météo récupère le résultat en format XML de la requête

```
meteo=v2.GetWeatherResult
```

```
// Extraction du code XML contenu dans la première balise "<Location>"
```

```
location = XMLExtraitChaîne(meteo, "Location")
```

```
n=Position(location,""),1,DepuisDébut)
```

```
longi=Milieu(location,n+2,5)
```

```
lati=Milieu(location,n+9,6)
```

```
lodegre=Remplace(longi,"-","°")
```

```
ladegre=Remplace(lati,"-","°")
```

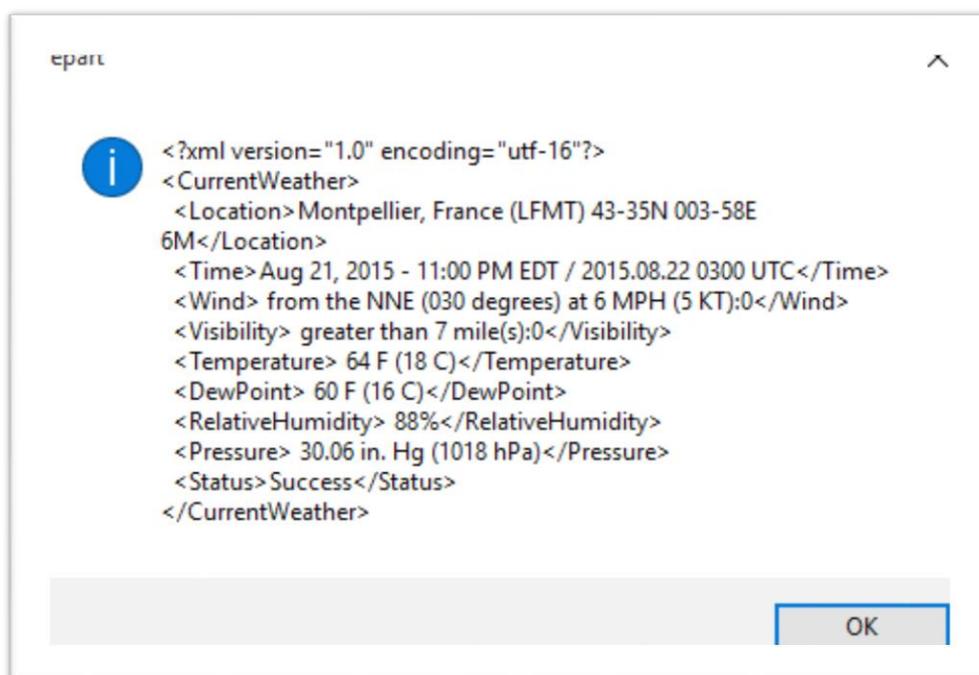
```
Longitude=lodegre
```

```
Latitude=ladegre
```

### Explication du code :

#### **meteo=v2.GetWeatherResult**

Le résultat de la requête au WebService va être envoyé dans la variable **météo**. Si nous avons fait un test du type « Info(meteo) » sur la ville de Montpellier en France vous liriez ceci :



Comme vous le voyez, plusieurs informations sont transmises :

- La date de la requête,
- La force du vent,
- La visibilité,
- La température,
- ...

```
// Extraction du code XML contenu dans la première balise "<Location>"
```

```
location = XMLExtraitChaîne(meteo, "Location")
```

Dans la variable Location, nous avons le contenu de la rubrique Location du fichier XML retourné.

Nous avons besoin des données de la latitude et de la longitude de la ville de Montpellier.

Pour pouvoir extraire les 2 informations, il nous faut les extraire de cette chaîne. C'est le but du code suivant :

```
n=Position(location,")",1,DepuisDébut)
```

```
longi=Milieu(location,n+2,5)
```

```
lati=Milieu(location,n+9,6)
```

On recherche la position de la parenthèse fermante du mot (**LFMT**) car on va trouver les coordonnées voulues juste après.

C'est le rôle de la fonction **Position**.

La fonction **Milieu** va nous permettre de « découper » un bout de texte dans cette chaîne. Pour la longitude les coordonnées débutent **2 caractères** après la parenthèse et sont écrites sur **5 caractères**.

**Longi** va donc recevoir **43-35**.

```
lodegre=Remplace(longi,"-","°")
```

```
ladegre=Remplace(lati,"-","°")
```

Il nous faut formater la longitude et la latitude en degrés. Il faut donc remplacer le caractère «-» par « ° »

C'est le rôle de la fonction **Remplace**.

## Test du code.

Lancez le projet et remplissez comme ceci :

Cliquez sur **Recherchez** :

Voilà nous avons les coordonnées Gps de l'aéroport de **Rodez** dans le magnifique département de l'**Aveyron** !

Comme vous ignorez sûrement où se situe Rodez il serait judicieux d'envoyer les coordonnées trouvée dans Google Maps. C'est le but du bouton Voir.

## Code du bouton Voir

LienHTTP est une chaîne

```
LienHTTP="http://maps.google.com/maps?q="
```

```
LienHTTP+= Longitude +"%2C"+Latitude
```

```
LanceAppliAssociée(LienHTTP)
```

## Explications :

Nous formatons une chaîne LienHTTP comme une adresse url de google maps contenant les coordonnées Gps de Rodez.

Ceci fait nous demandons à Windev de lancer l'application associée aux Url.

Windev va donc lancer le navigateur par défaut avec l'url fabriquée juste avant.

Maintenant sous vos yeux émerveillés vous découvrez l'Aveyron et l'aéroport de Rodez !