Dictionary Web Service

Lecturer Paul POCATILU, PhD
Economic Informatics Department, ASE Bucharest

Using Web services in developing applications has many advantages. In this paper is described an English-Romanian, Romanian English dictionary implemented as a XML Web service, called DService. A simple client for the Web service is also provided.

Keywords*: XML Web Services, ASP .NET, C#*

1 Introduction

Web services are methods for stored objects on the server to accept requests from the clients using XML over the HTTP. Using the description of the Web service from a server, a proxy server is generated. The requests from the clients are received by the proxy, and sent to server using HTTP. The response from the server are received by the proxy as XML over the HTTP and sent to the client as the result of the request. In figure1 are depicted the interactions between the clients, proxy and the server.

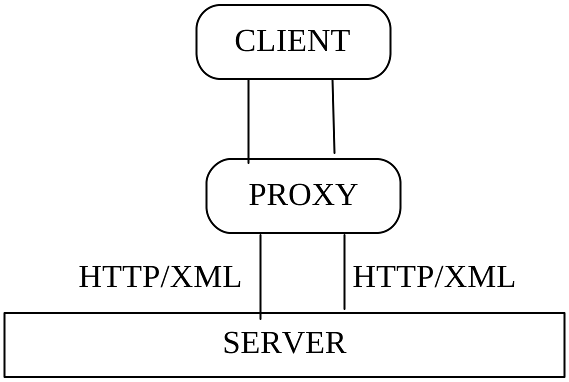


Fig. 1 Accessing Web services

The supported requests types are HTTP GET, HTTP POST and SOAP. Web services are developed using Java or ASP.NET.

2. The architecture of the Dictionary Web Service

The Dictionary Web Service has developed using ASP.NET and it was implemented in a class named DService. The source code of the DService is given in figure 2. It can be seen that the class DService is derived from the System.Web.Services.WebService.

The class has two methods that will be exposed to the Web: *Tradu* and *Translate*. Each method has its description and returns a string that represents the translation.

Inside the class, as a private member is an object of Dict type. The Dict class handles the access to the database, sending the word and getting its translation from the required table.

DService supports all protocols for communication (HTTP GET and POST and SOAP). The supported protocols are obtained through SDL (Service Descriptor Language) that is based on XML.

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Web;
using System.Web.Services;

namespace Dictionary
{ public class DService : System.Web.Services.WebService
  {  Dict d=new Dict();
     public DService()
     {
     }

     [WebMethod(Description="Translate a word from english to romanian")]
     public string Translate(string word)
     {  return d.Find("EN",word);
     }

     [WebMethod(Description="Traduce un cuvint din romana in engleza")]
     public string Tradu(string cuvint)
     {  return d.Find("RO",cuvint);
     }
  }
}
```

Fig. 2. The source code of the Web service DService

In figure 3 is shown the architecture of the Dictionary Web service. Various types of clients could access the service over the Internet.

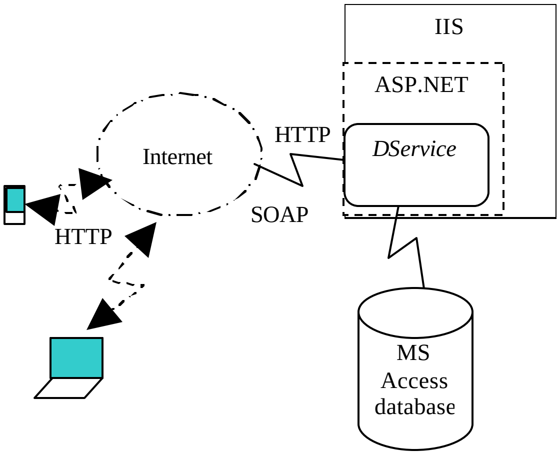


Fig. 3 The architecture of the Dictionary Web Service

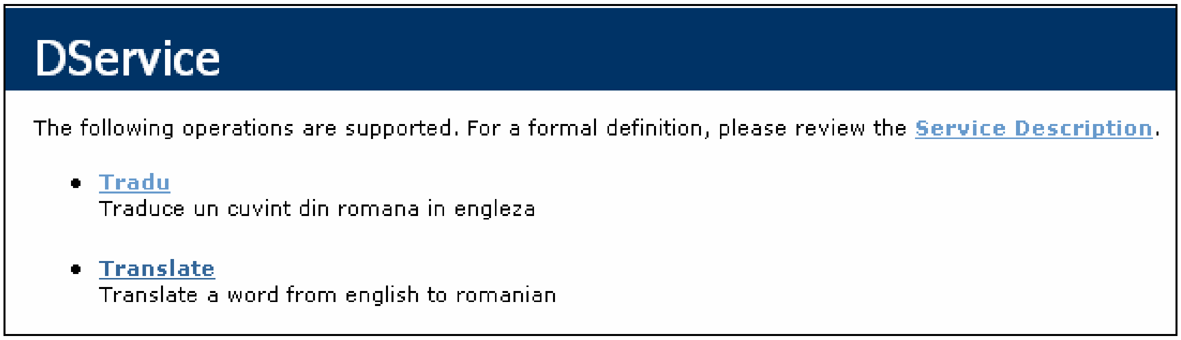
The DService is stored in a directory called Dictionary that is published on the IIS as an application.

3. Using the Web service

In order to create a client for the Web Services, a proxy needs to be generated. There is a tool WSDL (Windows Services Description Language) that creates a proxy having the address of the service. For the Web service Dictionary, the proxy is generated using wsdl.exe from the command line:

wsdl.exe http://pocatilu.ase.ro/Dictionary/DService.asmx?wsdl

The result is a C# source file (DService.cs). Accessing the service through Internet Explorer using its URL http://pocatilu.ase.ro/Dictionary/DService.asmx, the Web browser will display the operations supported by the service, as shown in figure 4.



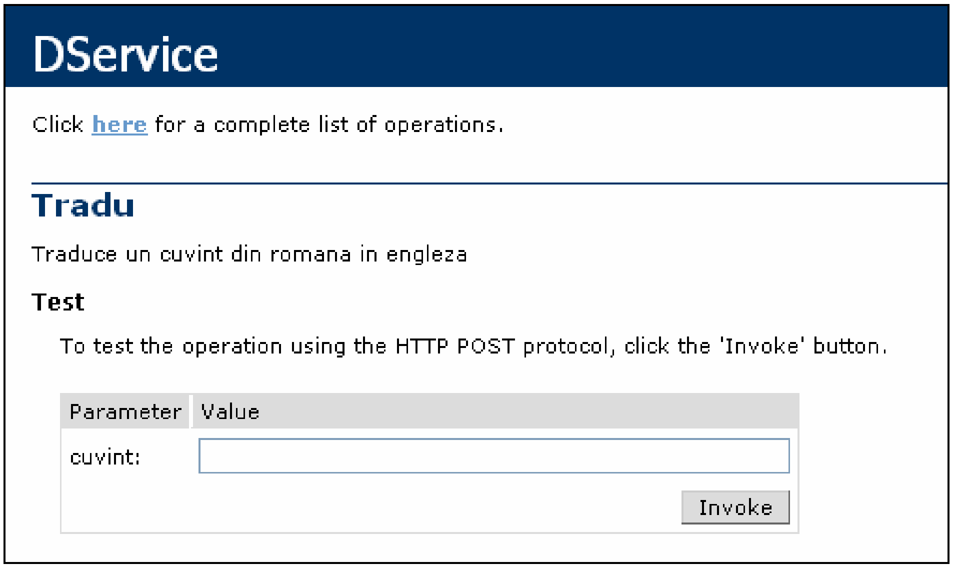
DService

The following operations are supported. For a formal definition, please review the Service Description.

- Tradu
  Traduce un cuvint din romana in engleza
- Translate
  Translate a word from english to romanian

Fig.4. Operations supported by the Dictionary Web Service

When an operation is selected from the local host, like *Tradu*, information about it will be displayed in browser, and also is given the possibility to invoke the operation, after the parameter was passed (figure 5).


DService

Click here for a complete list of operations.

Tradu

Traduce un cuvint din romana in engleza

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
cuvint:	

Fig.5. The description of the *Tradu* operation

A simple client console application for the DService Web Service has the following source code, given in figure 6.

```
using System;

class DSClient
{
public static void Main(string[] args)
{
  // Creating an instance of DService Web Sevice
  DService myDS = new DService();

  Console.Write("Cuvintul in Romana: ");
  string cuvint=Console.ReadLine();

  //the call of the Tradu method of DService
  string traducere=myDS.Tradu(cuvint);

  // Showing the result to the console
  Console.WriteLine("Traducerea in Engleza: {0}",traducere);
  }
}
```

Fig. 6. The source code of the DService client

In order to create the executable file, the client application has to be compiled together with the proxy:

csc.exe DSClient.cs DService.cs

The client application for mobile devices is developed using Microsoft .NET Compact Framework.

4. Conclusions

Using XML Web Services has many advantages for developers and for users: they use simple protocols and the implementation of the services and clients is easier than other methods. The client application could be based on Windows Forms or a mobile application or Web based. The Dictionary Web Service could be easily improved by adding other methods and new functionalities.

Bibliography

[IVAN01] IVAN, Ion, POCATILU, Paul, TOMA, Cristian, LEAU, Alexandru – *e3-commerce: e-commerce, mobile application.*

Aplicatia e3com, Informatica Economica, Nr. 3(19)/2001, Bucuresti 2001

[IVAN02] IVAN, Ion, POCATILU, Paul, POPA, Marius, TOMA, Cristian, – *Semnatura electronica si securitatea datelor în comertul electronic*, Revista Informatica Economica Nr. 3/2002, Bucuresti 2002

[PLAT01] PLATT, David – *Building reusable Web Components with SOAP and ASP.NET*, in MSDN Magazine, February 2001, Vol. 16, No 2, pp. 100-114.

[POCA03] POCATILU, Paul – *Mobile Applications Testing* în volumul International Workshop IE&SI, editia a II-a, Timisoara, 23-24 Mai 2003, pag 41-44

[POCA03a] POCATILU, Paul, TOMA, Cristian – *Calitatea aplicatiilor mobile*, în volumul Conferintei internationale "Rolul stiintei si învatamântului economic în realizarea reformelor economice din Republica Moldova", Chisinau, 25-26 septembrie 2003, pp. 474-477

[PRES00] PRESSMAN, Roger S. – *Software Engineering – A Practitioner's Approach, European Adaptation Fifth Edition,* McGraw-Hill, 2000

[WEI03] WEI-MENG, Lee – *Developing Mobile Applications Using the Microsoft Mobile Internet Toolkit*, http://www.devx.com/wireless/Article/10148, 2003

[WIGL02] WIGLEY, Andy, ROXBURGH Peter – *Building .NET Applications for Mobile Devices*, Microsoft Press, 2002