Web Services in C#

Kip Irvine

Updated: 3/19/2012
Contents

- Introducing XML Web Services
- Creating the HelloService
- BookService Web Service
11.1 Introducing XML Web Services

- **XML Web Services** is a technology that permits computers to share data and methods across networks, particularly across the Internet.
  - also known as: *Web Services*
  - applications can be built that collect data and perform useful operations from a wide variety of sources.
  - Web services are strongly supported by the .NET Framework.
Web Services

• Stored on a Web server
• No visual interface
• Permits clients to call methods on a server
  – functions are called service methods
• A client application *consumes* a Web service
Web Service Technology

• eXtensible Markup Language (XML)
• Simple Object Access Protocol (SOAP)
• Web Services Description Language (WSDL)
• Universal Description, Discovery, and Integration (UDDI)
Windows Communication Foundation

• Also known as WCF
• Microsoft technology that enhances and extends Web service technology
  − extra security
  − more flexibility than standard Web Services
  − lots of support from the .NET Framework
Attribute Classes

• WCF classes that identify elements of a WCF service
• ServiceContractAttribute
  – identifies the Interface that contains service methods
• OperationContractAttribute
  – identifies a single service method
•DataContractAttribute
  – identifies a class that defines a custom data type
•DataMemberAttribute
  – identifies a property in a custom data class
WCF Application Files

• Service host file (Service.svc)
  - identifies the host language, service name, and the name of the codebehind (service implementation) file

• Service contract file (IService.vb)
  - defines the interface (class) containing service methods and data contracts

• Service implementation file (Service.vb)
  - implements the interface that was defined in the service contract file

• Web configuration file (Web.config)
  - service configuration information
### Creating a Hello Service

#### Add New Web Site

<table>
<thead>
<tr>
<th>Installed Templates</th>
<th>Visual C#</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASP.NET Web Site</td>
<td>Visual C#</td>
</tr>
<tr>
<td>ASP.NET Empty Web Site</td>
<td>Visual C#</td>
</tr>
<tr>
<td>ASP.NET Dynamic Data Entities Web Site</td>
<td>Visual C#</td>
</tr>
<tr>
<td>ASP.NET Dynamic Data Linq to SQL Web Site</td>
<td>Visual C#</td>
</tr>
<tr>
<td>WCF Service</td>
<td>Visual C#</td>
</tr>
<tr>
<td>ASP.NET Reports Web Site</td>
<td>Visual C#</td>
</tr>
<tr>
<td>ASP.NET Crystal Reports Web Site</td>
<td>Visual C#</td>
</tr>
</tbody>
</table>

**Type:** Visual C#

A Web site for creating WCF services

- **Web location:** File System
- **Web location:** `\bear-ed.cs.fiu.edu\homes\Desktop\HelloService`
HelloService Example

```csharp
[ServiceContract]
public interface IHelloService
{
    [OperationContract]
    string GetMessage(int value);

    [OperationContract]
    CompositeType GetDataUsingDataContract(CompositeType composite);
}

public class HelloService : IHelloService
{
    public string GetMessage(int value)
    {
        if (value == 0)
            return "Hello there!";
        else
            return "Bonjour!";
    }
}
```
Service Definition (.svc)

```
<%@ ServiceHost Language="C#" Debug="true" Service="HelloService" CodeBehind="~/App_Code/Service.cs" %>
```
Consuming a Web Service

• Also known as a *client application*
• Can be a Windows Forms or ASP.NET Web site
• Web reference makes the Web service known to the client
  - can be within the same solution container
  - running on the same machine, under IIS
  - on a local network
  - on a remote server (Internet)
Consuming the Hello Service
Add a Service Reference

Server Explorer Window:

Find the service

Add a Service Reference

To see a list of available services on a specific server, enter a service URL and click Go. To browse for available services, click Discover.

Address:
http://localhost:65633/HelloService/HelloService.svc

Namespace:
localhost

Select a service contract to view its operations.

1 service(s) found in the solution.
protected void btnGo_Click(object sender, EventArgs e)
{
    localhost.HelloServiceClient client = new localhost.HelloServiceClient();
    lblResponse.Text = client.GetMessage(0);
}

protected void btnFrench_Click(object sender, EventArgs e)
{
    localhost.HelloServiceClient client = new localhost.HelloServiceClient();
    lblResponse.Text = client.GetMessage(1);
}
BookService Web Service

- Service methods that do the following:
  - return a list of books
  - search for a single book, using its ISBN number
- Client application's user interface:

![BookService Client Application](image)
Service Description

• Web Service Description (wsdl) File
  - describes the service methods as messages:

  - `<wsdl:message name="IBookService_GetBookList_InputMessage">
      <wsdl:part name="parameters" element="tns:GetBookList" />
  </wsdl:message>

  - `<wsdl:message name="IBookService_GetBookList_OutputMessage">
      <wsdl:part name="parameters" element="tns:GetBookListResponse" />
  </wsdl:message>

  - `<wsdl:message name="IBookService_FindBook_InputMessage">
      <wsdl:part name="parameters" element="tns:FindBook" />
  </wsdl:message>

  - `<wsdl:message name="IBookService_FindBook_OutputMessage">
      <wsdl:part name="parameters" element="tns:FindBookResponse" />
  </wsdl:message>`
Tutorial 11-2

• Consuming BookService from a Web Application
• Schema Description (XSD) File:
Tutorial 11-3

• Consuming BookService from a Windows Forms Application
Displaying Service References

![Solution Explorer with Service References](image)

- Solution 'Chapt_11' (3 projects)
  - BookService WinForms Client
    - My Project
    - Service References
    - LocalServices
    - App.config
    - Form1.vb
  - D:\...\Book Service\Client
  - D:\...\BookService Web Client
- App_WebReferences.lhagmddw
  - BookService WinForms Client
    - {} BookService_WinForms_Client
      - {} BookService_WinForms_Client.LocalServices
      - Book
      - BookServiceClient
        - IBookService
        - IBookServiceChannel
      - {} BookService_WinForms_Client.My
      - {} BookService_WinForms_Client.My.Resources
ObjectDataSource

- You can connect an ObjectDataSource directly to a Web service
  - the Web service is represented by a local proxy class
- Steps:
  1. Select Configure Data Source from the Tasks window.
  2. Select localhost.ServiceName
  3. Select the Web service method name that will represent the Select query
  4. Optional: select other Web service methods for Update, Insert, and Delete operations
The End