

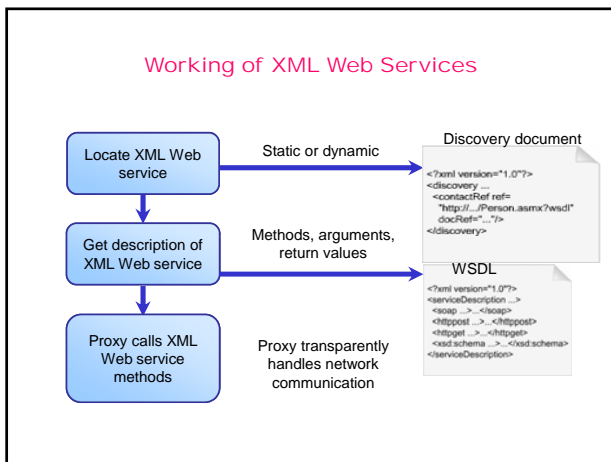
XML Web service is an URL-addressable set of functions that is exposed over a network to serve as a building block for creating distributed applications.

Uses

Internet technologies: HTTP, XML, and SOAP (a lightweight HTTP- and XML-based protocol used for information exchange)

Basic elements

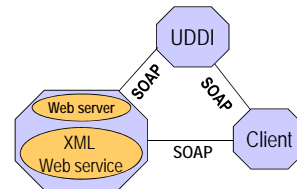
- XML Web service broker - hosts a global registry of available Web services
- XML Web service provider - hosts a Web service
- XML Web service consumer - hosts any client using HTTP (browser, console application, GUI application)



SOAP (Simple Object Access Protocol) is a lightweight XML-based protocol for exchange of information in decentralized, distributed environments.

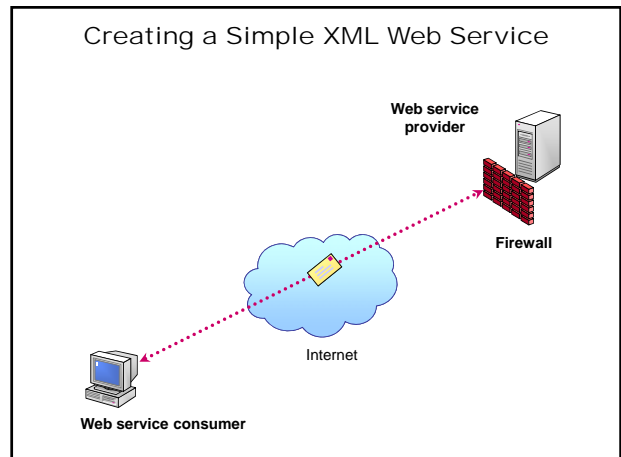
Characteristics

- Pass by reference
- Pass objects, structures, data sets



WSDL (Web Services Description Language) - an XML grammar used for describing a Web service in terms of the messages it accepts and generates

WSDL document - defines the types used in the operations (methods) of a Web service and the documents that are exchanged for each operation



1 Creating a Web service on the local Web server using the ASP.NET Web Service project template

File ⇒ New ⇒ Web Site ⇒
 New Web Site dialog box
 Templates: ⇒ ASP.NET Web Service
 Location: address of the Web server and directory name
 http://localhost/Website
 http://localhost/ by default

2 Add the [WebService] attribute to the Service class with the namespace identified the Web service

[WebService(Namespace = "http://localhost/Website/")]

3 Define methods with the [WebMethod] attribute that exposes methods as the XML Web service

[WebMethod(Description = "This method converts a temperature in " + "degrees Fahrenheit to a temperature in degrees Celsius.")]
 public double FahrenheitToCelsius(double fromValue)

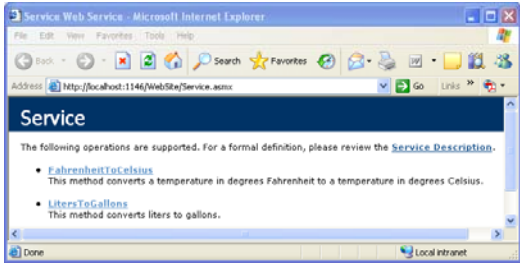
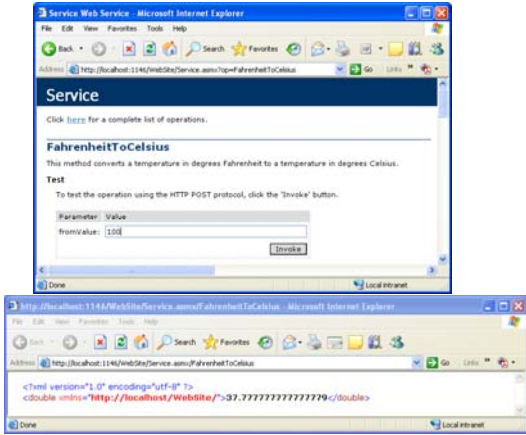
4 Compile the XML Web service

5 Test the XML Web service

Internet Explorer:
 http://localhost/Website/Service.aspx

Example: Create a simple Web service with two Web methods:

- Converts a temperature in degrees Fahrenheit to a temperature in degrees Celsius
- Converts volumes of liters to gallons

```
using System;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Data.SqlClient;
using System.Data;

[WebService(Namespace = "http://localhost/Website/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]

public class Service : System.Web.Services.WebService
{
    public Service()
    {
        //Uncomment the following line if using designed components
        //InitializeComponent();
    }
}
```

```
[WebMethod(Description = "This method converts a temperature in " +
"degrees Fahrenheit to a temperature in degrees Celsius.")]

public double FahrenheitToCelsius(double fromValue)
{
    return ((fromValue - 32) * 5) / 9;
}

[WebMethod(Description = "This method converts liters to gallons.")]

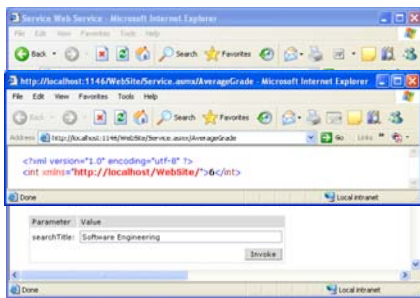
public double LitersToGallons(double fromValue)
{
    return fromValue / 3.7854;
}
}
```

Example: Create a Web service that calculates the number of students enrolled in a course with a given course number.



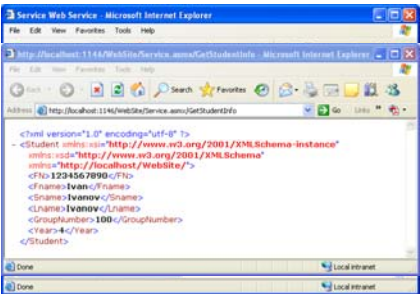
```
[WebMethod(Description = "This method calculates the number of students " +
"enrolled in a course with a given course number.")
public int Count (int newNumber)
{
    int count = 0;
    string sql = "select Count(FN) from enroll where (CourseNumber=@CourseNumber)";
    string connectionString = "Data Source=Mania;" +
    "Initial Catalog=ManagementEnroll;Integrated Security=True";
    using (SqlConnection conn = new SqlConnection(connectionString))
    {
        SqlCommand cmd = new SqlCommand(sql, conn);
        cmd.Parameters.Add("@CourseNumber", SqlDbType.Int);
        cmd.Parameters["@CourseNumber"].Value = newNumber;
        try
        {
            conn.Open();
            count = (int)cmd.ExecuteScalar();
            conn.Close();
        }
        catch (Exception e)
        {
            throw new Exception("Error!" + e.Message);
        }
    }
    return count;
}
```

Example: Create a Web service that calculates the average grade for a given course title.



```
[WebMethod(Description = "This method calculates the average grade " +
"for a given course title.")
public int AverageGrade (string searchTitle)
{
    int average = 0;
    string sql = "select AVG(grade) from enroll where CourseNumber in " +
    "(select CourseNumber from Courses where (title = @title))";
    string connectionString = "Data Source=Mania;" +
    "Initial Catalog=ManagementEnroll;Integrated Security=True";
    using (SqlConnection conn = new SqlConnection(connectionString))
    {
        SqlCommand cmd = new SqlCommand(sql, conn);
        cmd.Parameters.Add("@title", SqlDbType.NVarChar);
        cmd.Parameters["@title"].Value = searchTitle;
        try
        {
            conn.Open();
            average = (int)cmd.ExecuteScalar();
            conn.Close();
        }
        catch (Exception e)
        {
            throw new Exception("Error!" + e.Message);
        }
    }
    return average;
}
```

Example: Create a Web service that gets a student information for a given faculty number.



```
WebService => Add New Item ... => Class

using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public class Student
{
    private string fn;
    private string fname;
    private string sname;
    private string lname;
    private int groupNumber;
    private int year;
}
```

```

public string FN
{
    get { return fn; }
    set { fn = value; }
}
public string FName
{
    get { return fname; }
    set { fname = value; }
}
public string Sname
{
    get { return sname; }
    set { sname = value; }
}
public string Lname
{
    get { return lname; }
    set { lname = value; }
}
public int GroupNumber
{
    get { return groupNumber; }
    set { groupNumber = value; }
}
public int Year
{
    get { return year; }
    set { year = value; }
}
public Student() {}
}
    
```

```

[WebMethod(Description = "This method gets student information " +
    "for a given faculty number.")]
public Student GetStudentInfo (string studentFN)
{
    Student student = new Student();
    try
    {
        string connectionString = "Data Source=Mania;" +
            "Initial Catalog=ManagementEnroll;Integrated Security=True";
        SqlConnection conn = new SqlConnection(connectionString);
        SqlCommand cmd = new SqlCommand();
        cmd.CommandText =
            "SELECT * FROM Students WHERE FN='"+studentFN+"'";
        cmd.Connection = conn;
        conn.Open();
        SqlDataReader r = cmd.ExecuteReader();
    }
}
    
```

```

if (r.Read())
{
    student.FN = r.GetString(0);
    student.Fname = r.GetString(1);
    student.Sname = r.GetString(2);
    student.Lname = r.GetString(3);
    student.GroupNumber = r.GetInt32(4);
    student.Year = r.GetInt32(5);
}
else
{
    throw new Exception
        ("Student with a faculty number " + studentFN + " is not found");
}
r.Close();
conn.Close();
return student;
}
catch(Exception e)
{
    throw new Exception ("Error in searching a student: "+ e.Message);
}
}
    
```

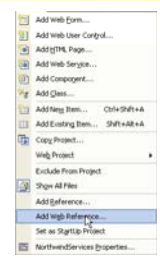
Read advances the SqlDataReader to the next record

GetX returns the value of the specified column

Creating a XML Web Services Client

- 1 Creating a client application
- 2 Generate the proxy

- Add Web reference
Project ⇒ Add Web Reference



OR

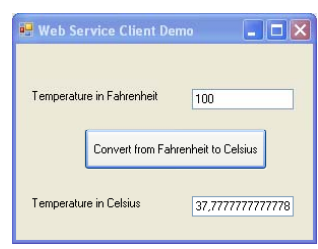
- Run **WSDL.exe**

- 3 Create an instance of the Web service


```
public localhost.Service convert = new localhost.Service();
```
- 4 Call XML Web service methods


```
double temp = convert.FahrenheitToCelsius(
                double.Parse(FahrenheitTextBox.Text));
```
- 5 Handle exceptions thrown by the XML Web service

Example: Calling the XML Web service from Windows Forms.



```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

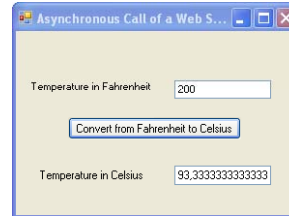
namespace WebServiceClient
{
    public partial class Form1 : Form
    {
        public localhost.Service convert = new localhost.Service();

        public Form1()
        {
            InitializeComponent();
        }

        private void convertButton_Click(object sender, EventArgs e)
        {
            double temp =
                convert.FahrenheitToCelsius(double.Parse(FahrenheitTextBox.Text));
            CelsiusTextBox.Text = temp.ToString();
        }
    }
}

```

Example: Calling the XML Web service asynchronously from Windows Forms.



The proxy class for every Web method
 public double FahrenheitToCelsius(double fromValue)
 automatically generates asynchronous method
 public void FahrenheitToCelsiusAsync(double fromValue)
 and a corresponding event for every Web method.

```

public event FahrenheitToCelsiusCompletedEventHandler
    FahrenheitToCelsiusCompleted;
public partial class FahrenheitToCelsiusCompletedEventArgs :
    System.ComponentModel.AsyncCompletedEventArgs

```

When the asynchronous method is called, it
 executes on another thread and raises its
 corresponding event when it returns.

The user can execute code when an asynchronous
 method returns by creating a handler for its
 corresponding event.

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace WebServiceClientAsync
{
    public partial class Form1 : Form
    {
        private localhost.Service convert = new localhost.Service();
        private double temp = 0;

        public Form1()
        {
            InitializeComponent();
            convert.FahrenheitToCelsiusCompleted += new
                localhost.FahrenheitToCelsiusCompletedEventHandler
                (convert_FahrenheitToCelsiusCompleted);
        }
    }
}

```

```

void convert_FahrenheitToCelsiusCompleted(object sender,
    localhost.FahrenheitToCelsiusCompletedEventArgs e)
{
    temp = e.Result;
    CelsiusTextBox.Text = temp.ToString();
}

private void button1_Click(object sender, EventArgs e)
{
    CelsiusTextBox.Text = String.Empty;
    convert.FahrenheitToCelsiusAsync
        (double.Parse(FahrenheitTextBox.Text));
}
}

```