

Web Services in ASP.NET — Simple Notes

What is a Web Service?

A **Web Service** is a component that allows different applications to communicate over the internet using standard protocols.

It exposes methods that other applications can call.

Web Methods

A **Web Method** is just a function inside a web service that is exposed so clients can call it.

You mark a method with:

```
[WebMethod]  
public int Add(int a, int b)  
{  
    return a + b;  
}
```

The **[WebMethod] attribute** tells ASP.NET to make it available to other applications.

Attributes

Web services use special attributes:

- **[WebService]** → identifies a class as a web service
- **[WebMethod]** → exposes a method
- **[SoapDocumentService] / [SoapRpcService]** → specify SOAP formatting
- **[WebServiceBinding]** → defines communication contracts

Attributes help control how the service behaves and how clients interact with it.

XML, WSDL, and UDDI

XML (Extensible Markup Language)

- Data exchanged between client and server is formatted in XML.
- SOAP messages use XML.

WSDL (Web Services Description Language)

- A WSDL file describes the web service.
- It explains:
 - What methods are available
 - What parameters they accept
 - What the service returns
 - How to call it
- Automatically generated by ASP.NET when you create a .asmx service.

UDDI (Universal Description, Discovery, and Integration)

- Like a “yellow pages” for web services.
- Allows publishing and discovering services.
- Rarely used today but important in theory.

▲ Three Important Aspects of Web Services

1 Creating a Web Service

- Create an .asmx file in ASP.NET.
- Write methods inside the service.
- Mark them using [WebMethod].

2 Adding Service Reference & Creating a Proxy

- To use a web service in another project, you **add a service reference**.
- Visual Studio automatically generates a **proxy class**.
- The proxy knows how to send and receive SOAP messages.

3 Consuming the Web Service

- Create an object of the **proxy class**.
- Call the web methods through the proxy as if you're calling a normal local function.

Example:

```
ServiceReference1.MyServiceSoapClient proxy = new ServiceReference1.MyServiceSoapClient();  
int result = proxy.Add(5, 7);
```

✳ How It Works (In Simple Terms)

Your Code

You call methods on the **proxy object**, just like calling normal C# methods.

Web Service Proxy

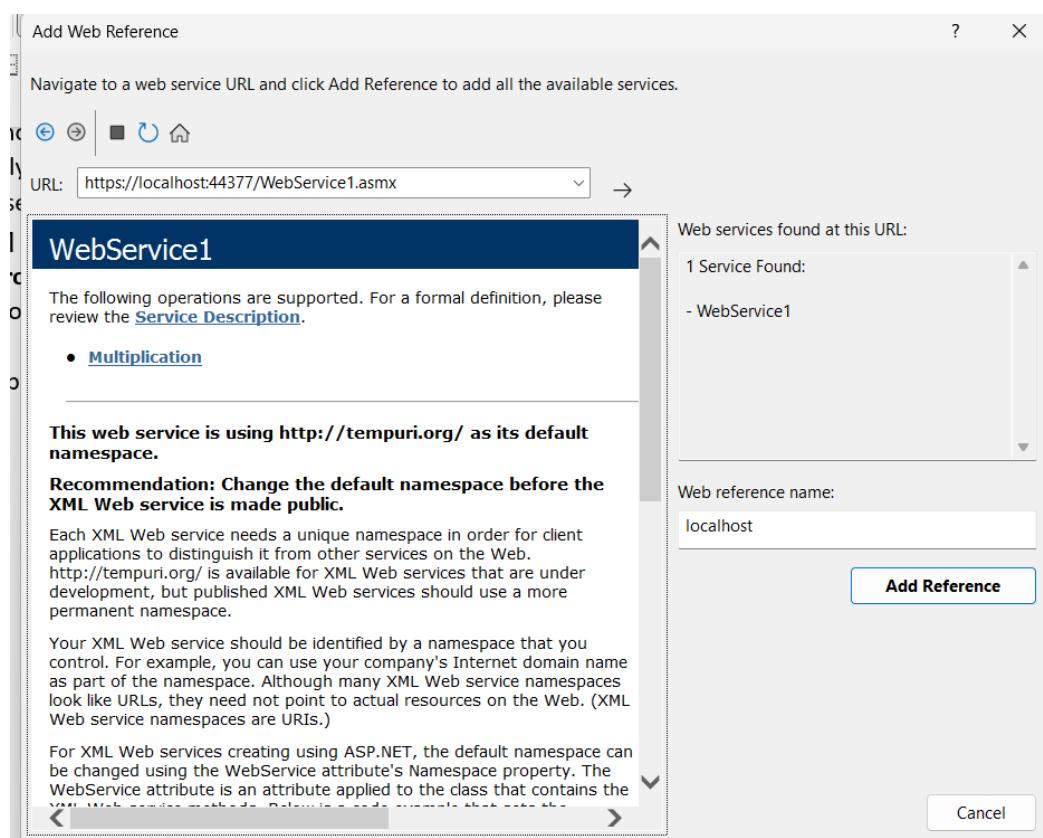
- Created automatically when adding a service reference.
- Takes your method call → turns it into a SOAP request → sends it to the server.

Web Service

- Receives the SOAP message.
- Executes the method.
- Returns the result back in SOAP XML.
- Proxy converts it back to normal data for your program.

✓ In One Line

Your code → Proxy → SOAP XML → Web Service → SOAP Result → Proxy → Your application



```
0 references
protected void Button1_Click(object sender, EventArgs e)
{
    localhost.WebService1 ob1 = new localhost.WebService1();
    int a = Convert.ToInt32(TextBox1.Text);
    int b = Convert.ToInt32(TextBox2.Text);
    int c = ob1.Multiplication(a, b);
    Label1.Text = c.ToString();
}
```

