



Research Article

Interoperability through Web Services: A Study Using ASP.NET Technology

Dr. Rishi Mathur

Assistant Professor, Department of Computer Science and Applications
S.P.U. P.G. College, Falna, Pali, Rajasthan, India

Corresponding Author: *Dr. Rishi Mathur

DOI: <https://doi.org/10.5281/zenodo.19495969>

Abstract

ASP.NET, developed by Microsoft, is a powerful web application framework that enables developers to build dynamic, scalable, and robust web applications for both desktop and mobile platforms. This research paper focuses on the study of web services and their core components within the ASP.NET environment. Web services are platform-independent software components that facilitate communication and data exchange between heterogeneous applications over a network using standard Internet protocols.

The study highlights the fundamental concepts, architecture, and working mechanism of web services, emphasising their role in enabling interoperability among applications developed in different programming languages such as Java, .NET, and PHP. It explores the key components of web services, namely SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), and UDDI (Universal Description, Discovery, and Integration), which collectively support message transmission, service description, and service discovery.

Furthermore, the paper discusses the features, benefits, and real-world applications of web services, including their loosely coupled nature, XML-based communication, and support for both synchronous and asynchronous operations. A practical implementation using ASP.NET web services (.asmx) is also presented to demonstrate how web methods can be created and consumed for data exchange, such as temperature conversion services.

The findings of this study indicate that web services play a crucial role in modern distributed computing by enabling seamless integration, reusability, and efficient communication across diverse platforms and applications. ASP.NET provides a comprehensive infrastructure for developing and deploying such services effectively.

Manuscript Information

- **ISSN No:** 2583-7397
- **Received:** 13-02-2026
- **Accepted:** 30-03-2026
- **Published:** 10-04-2026
- **IJCRM:5(2); 2026:** 493-498
- **©2026, All Rights Reserved**
- **Plagiarism Checked:** Yes
- **Peer Review Process:** Yes

How to Cite this Article

Mathur R. Interoperability through Web Services: A Study Using ASP.NET Technology. Int J Contemp Res Multidiscip. 2026;5(2):493-498.

Access this Article Online



www.multiarticlesjournal.com

KEYWORDS: Web Service, SOAP, WSDL, UDDI

INTRODUCTION

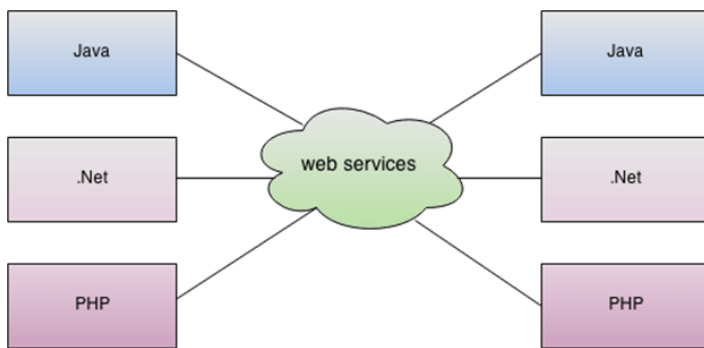
A Web Service can be defined in the following ways:

- It is a client-server application or application component for communication.
- The method of communication between two devices over the network.
- It is a software system for interoperable machine-to-machine communication.
- It is a collection of standards or protocols for exchanging information between two devices or applications.

Basic Means of Web Service: We can use ASP.NET to design websites that include web pages and web services. It is capable of connecting to a database and accessing data on ASP.NET controls. We can set security options on restricted pages. Web Services allow a consumer site (local) to obtain information from a provider site.

For example:

Any local website can display real-time data using web services provided by the Main site(www.Amazon.com), but keep the user on the page of the local site.



In the above diagram, Java, .NET, and PHP applications can communicate with other applications through a web service over the network.

For example, the Java application can interact with Java, .NET, and PHP applications. So, a web service is a language-independent way of communication.

Features of Web Services:

- ASP.NET 2.0 offers a complete web-services solution.
- Web services are a method of making information available that could be accessed by any developer's application over the Web.
- Web services can form a library of information that could be anything, like a mathematical function calculator.
- A web service is not a local web application and does not render as web pages, nor as executable files (.exe); It is just like a user interface.
- The information contained in the web service is wrapped up as an XML document (in other words, plain text).
- Web services communicate using open protocols like SOAP.
- Web services are a self-contained and self-describing mechanism.

- HTTP and XML are the basis for Web services.
- Web services can be published, found, and used on the Web.
- Web services use XML to code and to decode data, and SOAP to transport it (using open protocols).
- Loosely Coupled
- Coarse-Grained
- Ability to be Synchronous or Asynchronous
- Supports Remote Procedure Calls (RPCs)
- Supports Document Exchange

Benefits:

- A web developer can use web services and integrate them into web applications.
- Web services save the time of developers and effort by reducing code duplication.

Way of using web services:

They can be used in one of two ways.

1. You can create a web service that is exposed to the web to share with other developers and other applications. Or
2. You can search for a web service that can be added to your own application. (They are similar to plug-ins in that respect.)

Components of Web Services:

There are three major web service components.

1. SOAP
2. WSDL
3. UDDI

Another way we can describe the components of Web services into different categories, like:

- The method of transmission
- The method used to wrap the web service up
- The way the web service is defined.

All have clear W3C standards associated with the technologies involved. And all these standards are based on XML. So they're quick and easy to download, and even easier to use.

1. SOAP

- SOAP is an acronym for Simple Object Access Protocol.
- SOAP is an XML-based protocol for accessing web services.
- SOAP is a W3C recommendation for communication between applications.
- SOAP is XML-based, so it is platform-independent and language-independent. In other words, it can be used with Java, .NET, or PHP language on any platform.

Basic concept: SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages. It is important for application development to allow Internet communication between programs.

Need for SOAP:

Today's applications communicate using Remote Procedure Calls (RPC) between objects like DCOM and CORBA, but HTTP was not designed for this. RPC represents a compatibility and security problem; firewalls and proxy servers will normally block this kind of traffic.

Solution:

SOAP was created to accomplish this in a better way to communicate between applications over HTTP, because HTTP is supported by all Internet browsers and servers.

General Features of SOAP:

- SOAP defines its own security known as WS Security.
- SOAP is a communication protocol
- SOAP is for communication between applications.
- SOAP is a format for sending messages
- SOAP communicates via the Internet.
- SOAP is platform independent.
- SOAP is language independent.
- SOAP is based on XML.
- SOAP is simple and extensible
- SOAP is a W3C recommendation in 24. June 2003

SOAP Building Blocks:

A SOAP message is an ordinary XML document containing the following elements:

Skeleton SOAP Message

```
<? xml version="1.0"?>
```

```
< soap: Envelope
xmlns: soap="http://www.w3.org/2001/12/soapenvelope"
soap: encodingStyle="http://www.w3.org/2001/12/ soap-encoding">
```

```
< soap: Header>
```

```
...
```

```
</soap: Header>
```

```
< soap: Body>
```

```
... ASP.NET Notes ...
```

```
</soap: Body>
```

```
</soap: Envelope>
```

Here:

- An Envelope element: that identifies the XML document as a SOAP message.
- A Header element: that contains header information.
- A Body element: that contains call and response information
- A Fault element: containing errors and status information
- All the elements above are declared in the default namespace for the SOAP envelope.

2. **WSDL:** WSDL stands for Web Services Description Language. It is the standard format for describing a web service. WSDL was developed jointly by Microsoft and IBM. Basic Mean: WSDL is a language for describing web services and how to access them.

Features of WSDL

- WSDL is an XML-based protocol for information exchange in decentralised and distributed environments.
- WSDL definitions describe how to access a web service and what operations it will perform.
- WSDL is a language for describing how to interface with XML-based services.
- WSDL is an integral part of Universal Description, Discovery, and Integration (UDDI), an XML-based worldwide business registry.
- WSDL is the language that UDDI uses.
- WSDL is pronounced as 'wiz-dull' and spelt out as 'W-S-D-L'.
- WSDL is written in XML.
- WSDL became a W3C Recommendation 26. June 2007
- WSDL document is just a simple XML document.
- It contains a set of definitions to describe a web service.

WSDL Usage

WSDL is often used in combination with SOAP and XML Schema to provide web services over the Internet. A client program connecting to a web service can read the WSDL to determine what functions are available on the server. Any special data types used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the functions listed in the WSDL.

WSDL breaks down web services into three specific, identifiable elements that can be combined or reused once defined.

The three major elements of WSDL that can be defined separately are –

- Types
- Operations
- Binding

A WSDL document has various elements, but they are contained within these three main elements, which can be developed as separate documents, and then they can be combined or reused to form complete WSDL files.

WSDL Elements**A WSDL document contains the following elements:**

- **Definition** – It is the root element of all WSDL documents. It defines the name of the web service, declares multiple namespaces used throughout the remainder of the document, and contains all the service elements described here.
- **Data types** – The data types to be used in the messages are in the form of XML schemas.
- **Message** – It is an abstract definition of the data, in the form of a message presented either as an entire document or as arguments to be mapped to a method invocation.

- **Operation** – It is the abstract definition of the operation for a message, such as naming a method, message queue, or business process, that will accept and process the message.
- **Port type** – It is an abstract set of operations mapped to one or more endpoints, defining the collection of operations for a binding; the collection of operations, as it is abstract, can be mapped to multiple transports through various bindings.
- **Binding** – It is the concrete protocol and data formats for the operations and messages defined for a particular port type.
- **Port** – It is a combination of a binding and a network address, providing the target address of the service communication.
- **Service** – It is a collection of related endpoints encompassing the service definitions in the file; the services

map the binding to the port and include any extensibility definitions.

- **Documentation** – This element is used to provide human-readable documentation and can be included inside any other WSDL element.
- **Import** – This element is used to import other WSDL documents or XML Schemas.

NOTE – WSDL parts are usually generated automatically using web services-aware tools.

WSDL Document Structure

A WSDL document can also contain other elements, like extension elements and a service element that makes it possible to group together the definitions of several web services in one single WSDL document.

```

<definitions>
  <types>
    definition of types.....
  </types>

  <message>
    definition of a message....
  </message>

  <port Type>
    <operation>
      definition of an operation.....
    </operation>
  </port Type>

  <binding>
    definition of a binding....
  </binding>

  <service>
    definition of a service....
  </service>
</definitions>

```

A WSDL document can also contain other elements, like extension elements and a service element that makes it possible to group together the definitions of several web services in one single WSDL document.

3. UDDI

UDDI is an acronym for Universal Description, Discovery and Integration. UDDI is an XML-based framework for describing, discovering and integrating web services.

UDDI is a directory of web service interfaces described by WSDL, containing information about web services.

A Web Service Example

In the following example, we will use ASP.NET to create a simple Web Service that converts the temperature from Fahrenheit to Celsius, and vice versa:

```

<%@ WebService Language="VBScript"
Class="TempConvert" %> Imports System
Imports System.Web.Services
Public Class TempConvert: Inherits WebService
<WebMethod()> Public Function FahrenheitToCelsius(ByVal
Fahrenheit As String) As String
if fahr="" or IsNumeric(fahr)=false then return "Error"
return (((fahr) - 32) / 9) * 5)
end function
<WebMethod()> Public Function CelsiusToFahrenheit(ByVal
Celsius As String) As String dim cel

```

```
cel=trim(replace(Celsius,",", "."))
if cel="" or IsNumeric(cel)=false then return "Error" return
(((cel) * 9) / 5) + 32
end function end class. This document is saved as an .asmx file.
This is the ASP.NET file extension for XML Web Services.
The next steps are basic VB programming. This application has
two functions. One to convert from Fahrenheit to Celsius, and
one to convert from Celsius to Fahrenheit.
The only difference from a normal application is that this
function is defined as a "WebMethod ()".
```

Use "WebMethod()" to convert the functions in your application into web services:

```
<WebMethod(> Public Function FahrenheitToCelsius(ByVal
Fahrenheit As String) As String
dim fahr fahr=trim(replace(Fahrenheit,",", "."))
if fahr="" or IsNumeric(fahr)=false then return "Error"
return (((fahr) - 32) / 9) * 5)
end function
<WebMethod(> Public Function CelsiusToFahrenheit(ByVal
Celsius As String) As String dim cel
cel=trim (replace(Celsius,",", "."))
if cel="" or IsNumeric(cel)=false then return "Error" return
(((cel) * 9) / 5) + 32)
end function
```

Here is the code to add the Web Service to a web page:

```
<form action='tempconvert.asmx/FahrenheitToCelsius'
method="post" target="_blank">
<table>
<tr>
<td>Fahrenheit to Celsius:</td>
<td>
<input class="frmInput" type="text" size="30"
name="Fahrenheit">
</td>
</tr>
<tr>
<td></td>
<td align="right">
<input type="submit" value="Submit" class="button">
</td>
</tr>
</table>
</form>
```

```
<form action='tempconvert.asmx/CelsiusToFahrenheit'
method="post" target="_blank">
<table>
<tr>
<td>Celsius to Fahrenheit:</td>
<td>
<input class="frmInput" type="text" size="30"
name="Celsius">
</td>
</tr>
<tr>
<td></td>
<td align="right">
```

```
<input type="submit" value="Submit" class="button">
</td>
</tr>
</table>
</form>
```

CONCLUSION

You can create a web service that is exposed to the web to share with other developers and other applications. A web service is not a local web application and does not render as web pages, nor as executable files (.exe); It is just like a user interface. Web services can form a library of information that could be anything like a mathematical function calculator. The information contained in the web service is wrapped up as an XML document (in other words, plain text). Web services communicate using open protocols like SOAP. Web services are self-contained and self-describing mechanism. HTTP and XML is the basis for Web services. Web services can be published, found, and used on the Web. Web services use XML to code and to decode data, and SOAP to transport it (using open protocols). Web service is a combination of three components like SOAP, WSDL, UDDI.

SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages. SOAP was created to accomplish this for better way to communicate between applications over HTTP, because HTTP is supported by all Internet browsers and servers.

WSDL is a language for describing web services and how to access them. WSDL is an XML-based protocol for information exchange in decentralized and distributed environments. WSDL is often used in combination with SOAP and XML Schema to provide web services over the Internet. A client program connecting to a web service can read the WSDL to determine what functions are available on the server. Any special data types used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the functions listed in the WSDL.

Universal Description, Discovery, and Integration (UDDI) specification defines a way to publish and discover information about Web services. In this matter we specify the example of XML web service for converts the temperature from Fahrenheit to Celsius, and vice versa.

Web service is capable to connect database and access data on ASP.NET controls. We can set security options on restricted pages. Web Services allow a consumer site (local) to obtain information from a provider site.

REFERENCES

1. Javatpoint. What is web service [Internet]. Available from: <https://www.javatpoint.com/what-is-web-service>
2. Javatpoint. Web services tutorial [Internet]. Available from: <https://www.javatpoint.com/web-services-tutorial>
3. DotNetSpider. Web service example discussion forum [Internet]. Available from: <https://www.dotnetspider.com/forum/185123-Webservice-Example>

4. TechAcademyPro. Computer and IT tutorials platform [Internet]. Available from: <https://www.techacademypro.in/>
5. Computer Hindi Notes. Computer learning notes in Hindi [Internet]. Available from: <https://computerhindinotes.com/>
6. eHindiStudy. Online study material in Hindi [Internet]. . Available from: <https://ehindistudy.com/>
7. FreeCodeSpot. Programming tutorials and resources [Internet]. Available from: <https://www.freecodespot.com/>
8. Wikipedia. ASP.NET [Internet]. Available from: <https://en.wikipedia.org/wiki/ASP.NET>
9. TutorialsPoint. SOAP tutorial [Internet]. Available from: <https://www.tutorialspoint.com/soap/>
10. W3Schools. Web services example [Internet]. Available from: https://w3schools.sinsixx.com/webservices/ws_example.asp
11. Microsoft. ASP.NET Web services overview [Internet]. Available from: [https://learn.microsoft.com/en-us/previous-versions/aspnet/8wbhsy70\(v=vs.100\)](https://learn.microsoft.com/en-us/previous-versions/aspnet/8wbhsy70(v=vs.100))
12. CodeProject. Simple web service program for temperature conversion [Internet]. Available from: <https://www.codeproject.com/Questions/309022/A-Simple-Web-Services-program-that-Converts-Temper>
13. W3Schools. XML web services [Internet]. Available from: https://www.w3schools.com/xml/xml_services.asp
14. Microsoft MSDN Forums. XML web service temperature conversion discussion [Internet]. Available from: <https://social.msdn.microsoft.com/Forums/en-US/93839f67-dc4b-4ffb-b336-782a8c853fb1/xml-web-service-temperature-conversion>
15. TutorialsPoint. UDDI overview [Internet]. Available from: https://www.tutorialspoint.com/uddi/uddi_overview.htm
16. Javatpoint. Web service components [Internet]. Available from: <https://www.javatpoint.com/web-service-components>
17. Wikipedia. Web services discovery [Internet]. Available from: https://en.wikipedia.org/wiki/Web_Services_Discovery
18. TutorialsPoint. WSDL introduction [Internet]. Available from: https://www.tutorialspoint.com/wsdl/wsdl_introduction.htm
19. Wikipedia. Web Services Description Language (WSDL) [Internet]. Available from: https://en.wikipedia.org/wiki/Web_Services_Description_Language
20. W3Schools. XML WSDL tutorial [Internet]. Available from: https://www.w3schools.com/xml/xml_wsdl.asp
21. Javatpoint. SOAPUI working with WSDL file [Internet]. Available from: <https://www.javatpoint.com/soapui-working-with-wsdl-file>

Creative Commons (CC) License

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution–Non-Commercial–No Derivatives 4.0 International (CC BY-NC-ND 4.0) license. This license permits sharing and redistribution of the article in any medium or format for non-commercial purposes only, provided that appropriate credit is given to the original author(s) and source. No modifications, adaptations, or derivative works are permitted under this license.

About the corresponding author



Dr. Rishi Mathur is an experienced Assistant Professor in Computer Science with over 15 years of teaching experience at the UG and PG levels. He specialises in programming, web technologies, and emerging IT concepts, with multiple research publications in reputed international journals. He is known for his effective teaching methods, strong academic guidance, and active contribution to research, seminars, and professional development programs.