**What are Web Services?**

Web Services are used to share data between same and other platforms.

Data Sharing Techniques:

* Text / CSV / Excel etc. Format (Download)
  + Disadvantage: Data will not update automatically, user has to re-download the data again and again.
  + If you are integrating this downloaded data in your application, then user has to re-upload the data again and again.
* SQL Server Access
  + Advantage: User can have the updated data all the time.
  + Disadvantage: User has the DB Access
  + Disadvantage: Platform Dependent.
* Web Services
  + User will not have DB Access
  + Application can integrate the updated data all the time.
  + Updated data is available all the time because the data is available in Cloud (Internet)
  + Platform independent

**What is the difference between ASMX, WCF, and ASP.NET Web API?**

Application architecture has evolved from monolithic architecture to SOA architecture in order to make better separation, then to more refined microservice architecture today.

Applications need to communicate among themselves, and to achieve this purpose, Microsoft has developped technologies like ASMX, WCF, and ASP.NET Web API.

Let’s explore them together today.

**1. ASMX**

**ASMX** (ASP.NET Web Services) is the primary web service technology in .NET 1.0 and .NET 2.0.

ASMX provides the ability to build web services that send **SOAP** (Simple Object Access Protocol) messages over **HTTP** protocol.

**1.1. SOAP message**

SOAP message is in **XML** format.

SOAP message request sample:



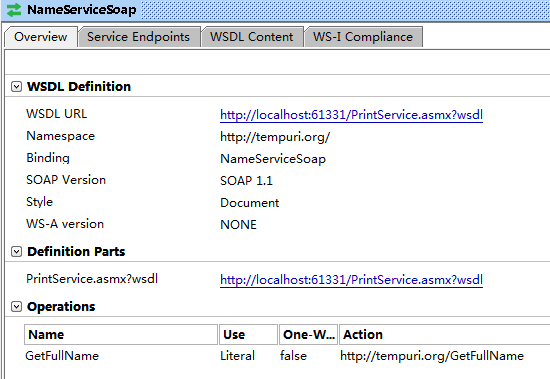
**1.2. ASMX web service code sample**

Text

Description automatically generated

**1.3. WSDL**

**WSDL** (Web Service Description Language) is a language to describe the service.



**1.3.1. WSDL structure**

Graphical user interface, text, application, email

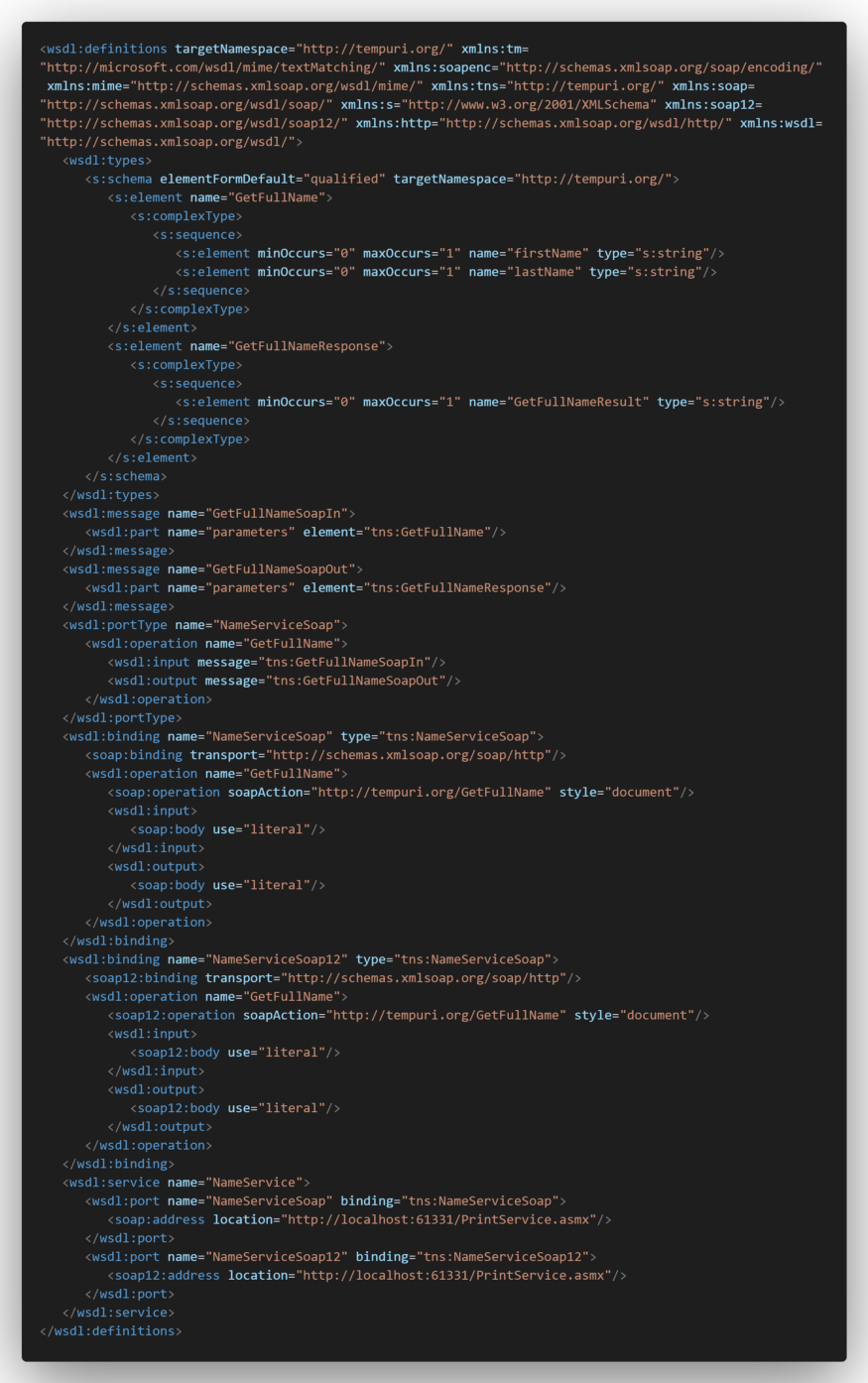
Description automatically generated

**1.3.2. PrintService’s WSDL detail**

Text

Description automatically generated

**1.3.3. PrintService’s WSDL code**



See more information about WSDL: [W3C WSDL Specification](https://www.w3.org/TR/wsdl20/)

**1.3.4. SOAP vs WSDL vs UDDI**

Table

Description automatically generated

**1.4. ASMX service consumption**

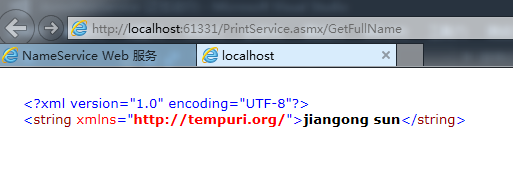
To consume the ASMX service, you can use any web browser or SoapUI.

**1.4.1. Consume ASMX service in Web browser**

**Call:**



**Result:**



**1.4.2. Consume ASMX service in SoapUI**

To consume the WCF service in SoapUI, you must enter a .WSDL suffix after .asmx.

Call and result:

Graphical user interface, text, application

Description automatically generated

**1.5. ASMX project source project**

Download [SOAP ASMX service sample project](https://github.com/hellomrsun/BlogCodeSource/tree/master/src/2020-02-16_ASMX_WCF_WebApi/01_ASMX/AsmxWebService)

## **ASMX Service**

1. It is based on SOAP and return data in XML form.
2. It support only HTTP protocol.
3. It is not open source but can be consumed by any client that understands xml.
4. It can be hosted only on IIS.

**2. WCF**

**WCF** (Windows Communication Foundation) is a framework for building service-oriented applications. WCF is introduced since .NET 3.0.

**2.1. WCF Architecture**

Diagram

Description automatically generated

Source: Microsoft

**2.2. WCF communication binding**

Diagram

Description automatically generated

Source: Microsoft

**2.3. SOAP WCF service sample code**

WCF can transfer **SOAP** messages over **HTTP** protocol by default, just like **ASMX** service.

SOAP WCF service sample code.

Text

Description automatically generated

**2.4. SOAP WCF WSDL**

Graphical user interface, text, application, email

Description automatically generated

**2.4.1. SOAP WCF WSDL detail**

A picture containing text

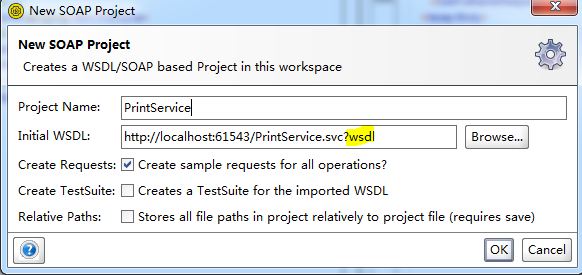
Description automatically generated

**2.4.2. SOAP WCF WSDL code**



**2.5. SOAP WCF service consumption**

**Add WSDL in SoapUI:**



PrintService.svc’s **.svc** means Service.

**SOAP WCF service consumption:**

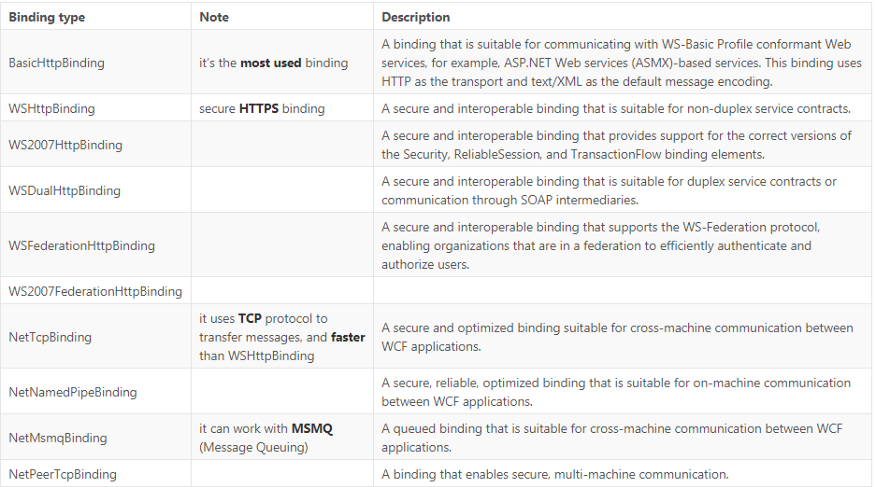
Graphical user interface, text, application

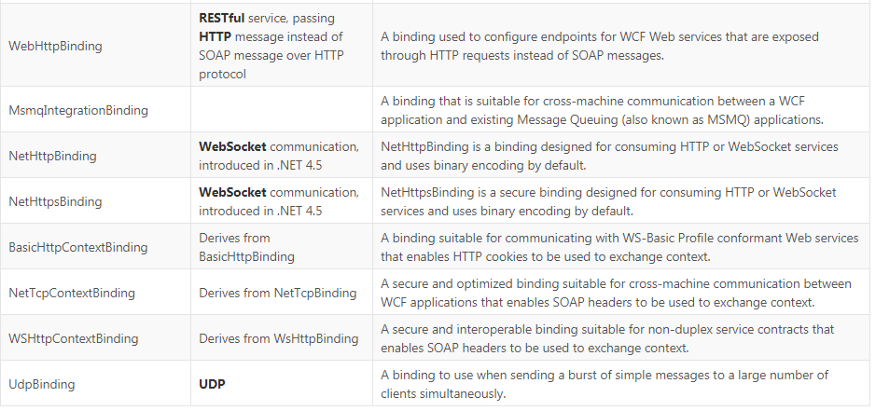
Description automatically generated

**2.6. WCF service bindings**

In addition, WCF provides much more features to build secure, complex WCF services. WCF can work with other protocols like **TCP**, **HTTPS**, and **UDP**.

Here are all the WCF Bindings:





**2.7. REST WCF service sample code**

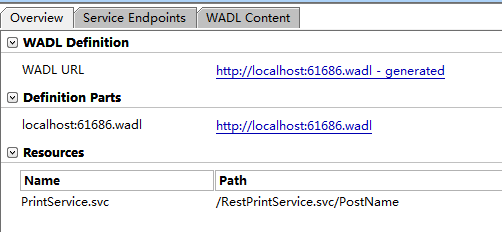
WCF also can act as **RESTful** service with WebHttpBinding.

REST WCF service sample:

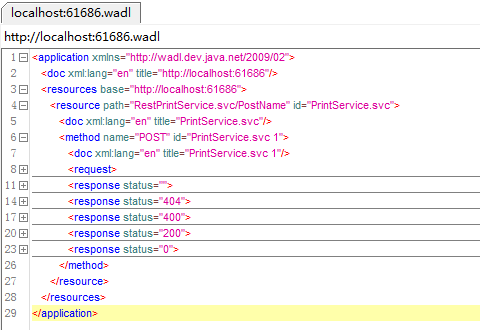


**2.8. REST WCF WADL**

**WADL** (Web Application Description Language) is a machine-readable XML description of HTTP-based web services.



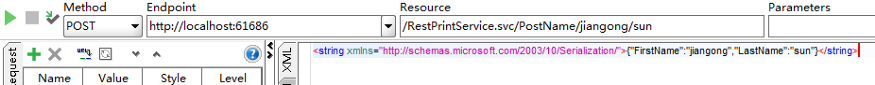
**2.8.1. REST WCF WADL structure**



**2.8.2. REST WCF WADL code**



**2.9. REST WCF service consumption**



**2.10. SOAP and REST WCF services source projects**

Download [SOAP and REST WCF service sample project](https://github.com/hellomrsun/BlogCodeSource/tree/master/src/2020-02-16_ASMX_WCF_WebApi/02_WCF/WcfWebService)

## **WCF**

1. It is also based on SOAP and return data in XML form.
2. It is the evolution of the web service(ASMX) and support various protocols like TCP, HTTP, HTTPS, Named Pipes, MSMQ.
3. The main issue with WCF is, its tedious and extensive configuration.
4. It is not open source but can be consumed by any client that understands xml.
5. It can be hosted with in the applicaion or on IIS or using window service.

## **WCF Rest**

1. To use WCF as WCF Rest service you have to enable webHttpBindings.
2. It support HTTP GET and POST verbs by [WebGet] and [WebInvoke] attributes respectively.
3. To enable other HTTP verbs you have to do some configuration in IIS to accept request of that particular verb on .svc files
4. Passing data through parameters using a WebGet needs configuration. The UriTemplate must be specified
5. It support XML, JSON and ATOM data format.

**3. ASP.NET Web API**

ASP.NET Web API pass messages over HTTP or HTTPS protocol.

Modern ASP.NET Web APIs are RESTful.

**3.1. REST**

**REST** (RepreSentational State Transfer)

REST Web API has the following characteristics:

* Client-server architecture

The principle behind the client-server constraints is the separation of concerns.

* Statelessness

The client-server communication is constrained by no client context being stored on the server between requests. Each request from any client contains all the information necessary to service the request, and the session state is held in the client. The session state can be transferred by the server to another service such as a database to maintain a persistent state for a period and allow authentication.

* Cacheability

Well-managed caching partially or completely eliminates some client-server interactions, further improving scalability and performance.

* Layered system
* Code on demand (optional)

Servers can temporarily extend or customize the functionality of a client by transferring executable code

* Uniform interface

The uniform interface constraint is fundamental to the design of any RESTful system.

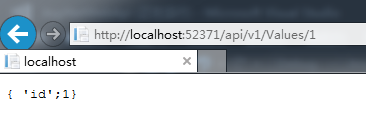
The four constraints for this uniform interface are:

1. Resource identification in requests
2. Resource manipulation through representations
3. Self-descriptive messages
4. Hypermedia as the engine of application state (HATEOAS)

**3.2. Web API sample code**



**3.3. Web API consumption**



**3.4. Web API source project**

Download [ASP.NET Web API sample project](https://github.com/hellomrsun/BlogCodeSource/tree/master/src/2020-02-16_ASMX_WCF_WebApi/03_WebApi/AspNetWebApi)

## **Web API**

1. This is the new framework for building HTTP services with easy and simple way.
2. Web API is open source an ideal platform for building REST-ful services over the .NET Framework.
3. Unlike WCF Rest service, it use the full featues of HTTP (like URIs, request/response headers, caching, versioning, various content formats)
4. It also supports the MVC features such as routing, controllers, action results, filter, model binders, IOC container or dependency injection, unit testing that makes it more simple and robust.
5. It can be hosted with in the application or on IIS.
6. It is light weight architecture and good for devices which have limited bandwidth like smart phones.
7. Responses are formatted by Web API’s MediaTypeFormatter into JSON, XML or whatever format you want to add as a MediaTypeFormatter.

Creating Web Service 🡪 Publish 🡪 Consume Web Service

How to Create ASMX Web Service?

* Create an empty ASP.Net Web Application
* Add .ASMX File
* Write Web Methods
* Publish so can consume

How to Consume ASMX Web Service?

* Web Service must be published and must be up and running
* Must know the configuration of the web service like Published URL, Web Methods etc.
* Create any Application like Windows or Web
* Add Web Service Reference in your project.
* Now web service can be consumed.

Enable Session State in ASMX Web Services:

Use the below Directive with the Web Method.

[WebMthod(EnableSession=true)]

Public <WEBMETHOD>

{

}

WebMethod Overloading

Usually Web Services does not support Method Overloading, but we can specify a name to a web method.

Calling ASMX Web Services using ASP.Net built-in Java Script AJAX Control.

* Java Script AJAX is used to Re-load a partial (any part) page.
* It requires Input Object and returns Output with Success and Failure Objects.