**Reusablity**

Functions : Within a program

DLL : : Across the pojects

Web Service , HTTP / HTTPS , SOAP Based , Request / Reply pattern

Web API : Rest Style HTTP / HTTPS , MVC

WCF , Supports multiple protocols . HTTP, TCP , Named Pipes , UDP , Communication could be one way , duplex

What is WCF Service

Windows Communication Foundation (WCF) is a framework for building service-oriented applications. Using WCF, you can send data as asynchronous messages from one service endpoint to another. A service endpoint can be part of a continuously available service hosted by IIS, or it can be a service hosted in an application.

WCF is a unified communication framework that integrates all the Communication frameworks .NET REMOTING, DCOM/COM and MSMQ into one logical model.

WCF increases the productivity in the sense that you learn only one programming model rather than learning multiple models as in the other communication frameworks describe above. Each one of them has its own programming model so if developers want to learn or work in any of them, they need to understand each framework programming model separately.

In WCF, you write code in one way and have many ways of communication, like in one case if you want SOAP based communication through MSMQ or in another case Restful based communication through Http etc.

* WCF is used for developing SOAP-based services whereas Web API is used for both SOAP-based and RESTful services.
* WCF does not offer any support for MVC features whereas Web API supports MVC features.
* WCF supports HTTP, UDP, and custom transport protocol whereas Web API supports only HTTP protocol.
* WCF offers Text, MTOM, and Binary Encoding support whereas Web API supports the UTF-8 encoding format.
* WCF supports message queues, message security, duplex communication, transaction whereas Web API doesn’t support.
* WCF stands for Windows Communication Foundation whereas API stands for Application Program Interface.
* WCF is an ideal option if you are planning to use .NET Framework 3.5.
* WCF support multiple protocols such as HTTP, TCP, Named pipe.
* WCF service is useful if you want to build service with standards like Transactions Messaging security and Reliable Message Security.
* You can use Request-Reply, One Way, and Duplex message exchange patterns with WCF.
* Web API provides supports for convention-based CRUD Actions since it works with HTTP verbs ( GET, PUT, POST, and DELETE)
* It is easy to define, expose, and consume in a REST-ful way.
* ASP.Net Web API allows you to showcase your data and services to various devices.
* Web API is open source, so it is an ideal option for creating RESTful services over the .NET platform.
* Allows you scalable, decoupled development of a Web Service which can be shared and used by other developers to access web services.
* It is also helpful to build simple, non-SOAP-based HTTP Services.

| **Parameter** | **WCF** | **Web API** |
| --- | --- | --- |
| Message Exchange Pattern | Request reply, one-way, or duplex. | By default request-reply only. However, additional message pattern can be supported with web sockets integration. |
| Hosting | IIS hosting, Self-hosting & works activation services. | Self-hosting & IIS hosting. |
| Open Source | It is not open source software. | It is shipped with.Net framework.It is also available as an independent download. |
| Support for MVC features | WCF does not provide any support for MVC features like controllers, routing, filter, auction results, etc. | ASP.NET Web API supports MVC features like routing, controllers, results, filter, action, etc. |
| Encoding support | Text, MTOM, and Binary. | Default support for UTF-8 encoding format. |
| Support for | Support for HTTP, UDP, and custom transport protocol. | Supports only HTTP protocol |
| Use of model | User attributed based programming model. | Maps HTTP verbs to methods |
| Configuration | Uses web.config and attributes to configure a HttpConfiguration class but not on the web.config Service. | Web API can be configured using web.config service. |
| RESTful services | It offers supports for RESTFUL service but with limitations. | Ideal for building RESTFUL services. |

**Web Service vs. WCF Service**

Web services use XMLSerializer which does not specify which fields or properties of the type are serialized into XML, while WCF uses DataContratSerializer that shows which fields or properties are serialized into XML. That's why WCF is preferred over web services, as DataContratSerializer performs better than XMLSerializer.

2. Web services support only one protocol- HTTP and HTTPS during communication, but WCF supports more protocols like- HTTP, TCP, and MSMQ that can be extended for a comprehensive solution, reliable session, and transactions. It signifies WCF is more adaptable to work together for a variety of software.

3. WCF offers much flexibility as web services can only be hosted in IIS, while WCF can be hosted in IIS, windows activation services, self-hosting, and Windows services.

4. WCF is excellent for building real-time applications because it's more reliable, fast and robust as compared to the web services.

5. Web services perform far better than WCF services even when they both execute some actions.

6. ASP.NET web services don't support duplex operations but WCF does.

7. When WCF development services are used, the hash table can be serialized, but not when using web services. Web services serialize only those collections that incorporate IEnumerable and ICollection.

8. Using service behavior classes, WCF supports multi-threading, but web services don't.

9. In [ASP.NET Development services](https://www.brainvire.com/asp.net-development/), SOAP messages are exchanged over HTTP, but WCF services can exchange the message using any format over any transport protocol. Though, SOAP is a default format that WCF uses.

10. WCF services have timeouts by default that can be configured. In WCF services, if the communication channel cannot be opened/ closed and there is no response within the time span, then WCF services get timed-out. Web services don't have any such property.

<https://stackify.com/soap-net-core/>

**SOAP** is a **protocol** which is **used to** interchange data between applications which are built on different programming languages. **SOAP** is built upon the XML specification and works with the HTTP **protocol**. This makes it a perfect for usage within web applications.

**WCF** offers much flexibility as **web services** can only be hosted in IIS, while **WCF** can be hosted in IIS, windows activation **services**, self-hosting, and Windows **services**.

**WCF Service Components**

Each WCF service has three main components

1. Address
2. Binding
3. Contract

**Address**

Address is called end-point also. They represent service location or contain a piece of information about where to send a message. Address specifies protocol (Http) and host (www.example.com) for using service.

**Bindings**

Binding specifies how to send a message and which protocol is used i.e. SOAP, or how a client can communicate with the service. Bindings also specify security constraints and other options. Some of the built-in bindings that WCF provides and their usage scenarios are as follow.

|  |  |
| --- | --- |
| **Binding Types** | **Usage Scenario** |
| WebHttpBinding | For Restful Communication using Http |
| BasicHttpBinding | For SOAP Communication using Http |
| NetTcpBinding | For .Net-To-.Net(Cross Machine) Communication using TCP |

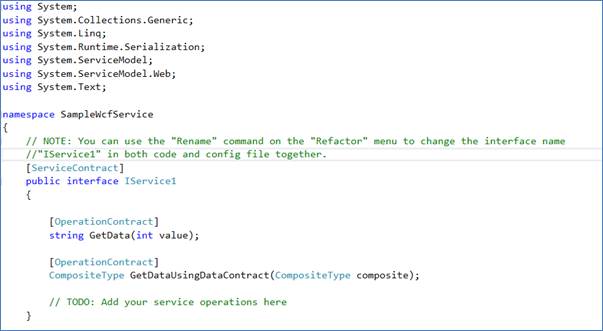
**Files**

* IService – Service Interface File
* Service.SVC – It’s the file where Service code is available and similar to .asmx file of web service
* Web.config – Configuration details where the Endpoint information are stored.

**IService.cs**

This is the file which has all the declarations rahther than definition of properties, here we call it Contract in WCF and this helps for all the operations that happen with the service named “Operation Contract”.

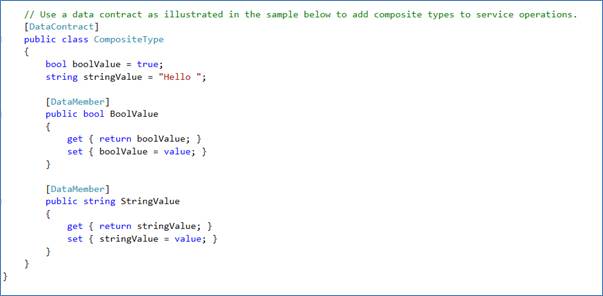
Operation Contract: The method is declared and where the actual implementation is done in .SVC file, each contact has to be decorated with the appropriate Attribute tags as shown below.



**Data Contract**

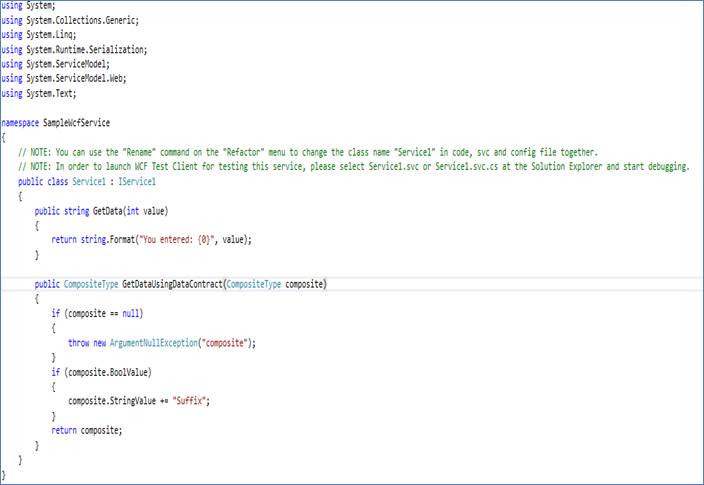
Here the Data are to be transferred and processed within service and they store the values, so in the WCF terminology they are called “Data Contract”.

Where each member of the Class; i.e., the Data contract is called “Data Member” and they are also to be decorated with the Attributes.



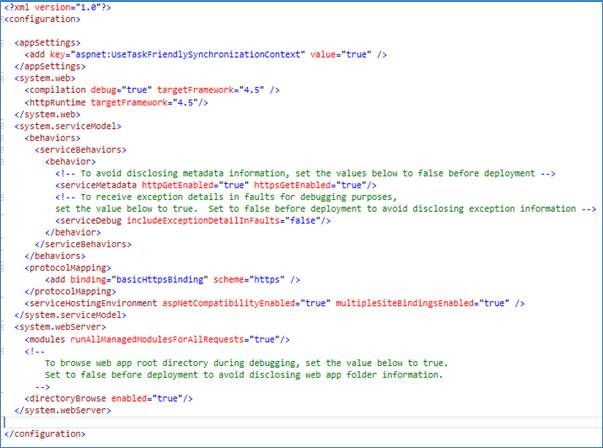
**Service.SVC**

This is the main file for any of the WCF services where this file inherits the “IService” interface and implements all the methods of the operation contract methods.



**Web.Config**

In a WCF application Web.config files play another important role, as the application will have various set of “ABC”- Address, Binding and Contract and all those are defined in the web.config files.



**Execute the Application using WCF Test Client**

The WCF Test Client is one of the best tools for developers to test the WCF application.

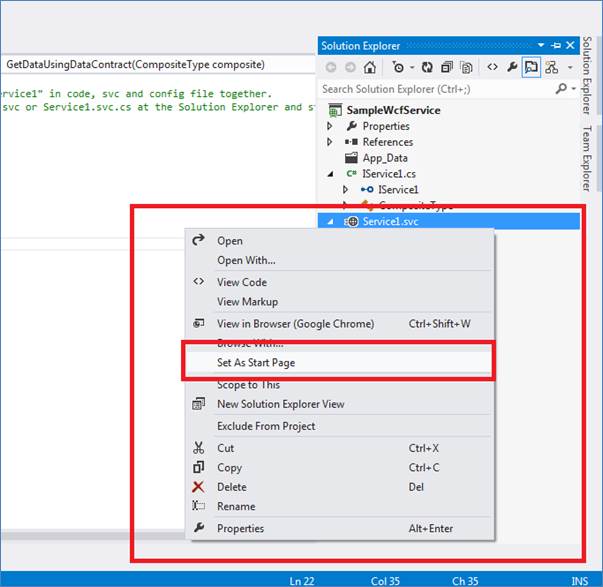
Windows Communication Foundation (WCF) Test Client (WcfTestClient.exe) is a GUI tool that enables users to input test parameters, submit that input to the service, and view the response that the service sends back. It provides a seamless service testing experience when combined with WCF Service Host.

This file will be available in the following location,

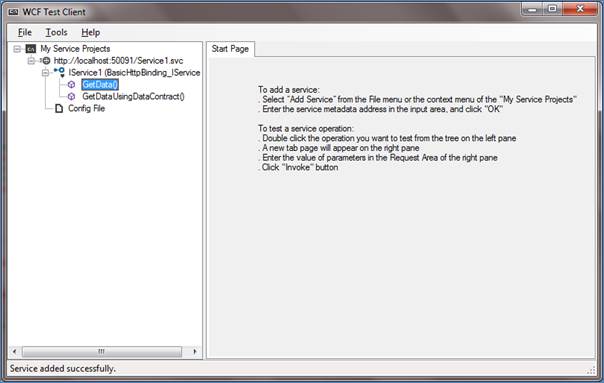
*C:\Program Files (x86)\Microsoft Visual Studio (Your Version Here)\Common7\IDE*

**Step 3**

Set the Service.SVC file as the “Start up page” and execute the application, so automatically the WCF Test client window will be displayed.

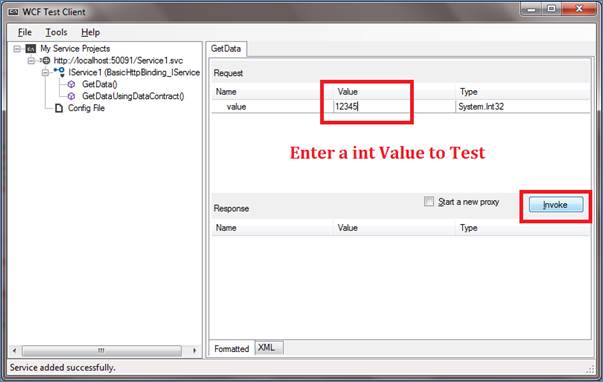


**Step 4 - WCF Test Client Execution**



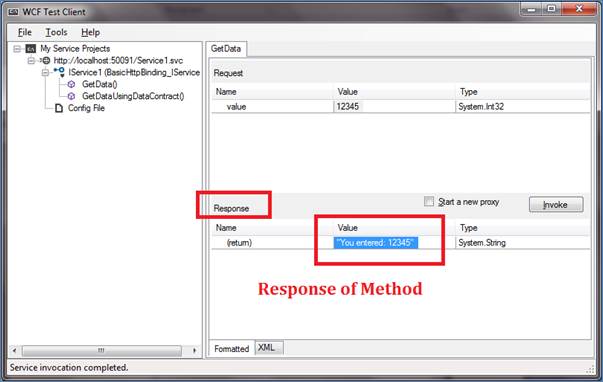
Here all the Methods will be displayed and also the appropriate method details will be displayed on the right side pane when you select the methods, which are readily testable.

**Step 5**



“Invoke” method will execute the method with the input value and gives us the response from the service and you can view the response in the application output section of WCF test client.

**Step 6**



Now you have successfully created the WCF application and also tested using a “WCF TEST CLIENT”, to make it real on the other part just think how many lines of code you have typed for this application's execution.

**WCF - Basic Difference Between Service Application and Service Library**

1. In the WCF Service Application we have a service contract i.e. IService1 for the service implementation and Service1 as a Web.config file.  
     
   In the **Service Library**we also have a service contract i.e. IService1 for the service implementation and Service1 as an App.config file for the configuration (instead of web.config as in the Service Application).
2. The major difference is that the **WCF Service Application** has a .svc file, whereas the WCF Service Library does not have a .svc file. Suppose we want to host this service application in IIS, then you need to specify for IIS the execution runtime environment requirements.  
     
   In a web application we set an .aspx, similarly for a Service Application we need to set a .svc.
3. In **WCF Service Application**the Service1.svc is hosted by default by the ASP.NET Development server where as:  
     
   A **WCF Service Library** is not hosted by the ASP.Net development server. It is hosted by the Test Client, since a Service Library does not have .svc file.
4. If you want to host your service in IIS then you should select the **WCF Service Application** template and if you want to host it as a Windows Service or set a reference of your library then you should select the **WCF Service Library** template.