**Service Contract**

****

**Rameshkartik.RS**

**Table of Contents**

[Introduction to Service Contract 3](#_Toc392775928)

[How do we need to define Service Contract? 4](#_Toc392775929)

[How should we need to implement it? 5](#_Toc392775930)

[Extracting Service MetaData 7](#_Toc392775934)

[Attachment 9](#_Toc392775943)

[Summary 9](#_Toc392775944)

**WHAT’s IN THIS ARTicle?**

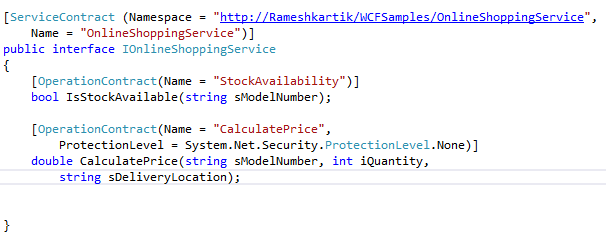
* Introduction to ServiceContract
* Steps to Create and Implement Service Contract
* Attachment
* Summary

# Introduction to Service Contract

Normally we write functionalities(logical implementation) and that should be accessed by the other method to get the benefit, Yes, what if the functionalities to be exposed to the outside world? Client may be a Non- .Net client Say Java or anyone. In the SOA world,the messages need to be exchanged in a SOAP format. Until we attribute the class as [ServiceContract] and the method as [OperationContract] the functionalities will not be exposed to all different types of clients. As per the operations defined in the service contract the SOAP messages with the schemas are exchanged between the client and server. Schemas are nothing but a pattern, it organizes the categories of information with the references. Ok, Let us see the steps involved in creating the service contract

# How do we need to define Service Contract?

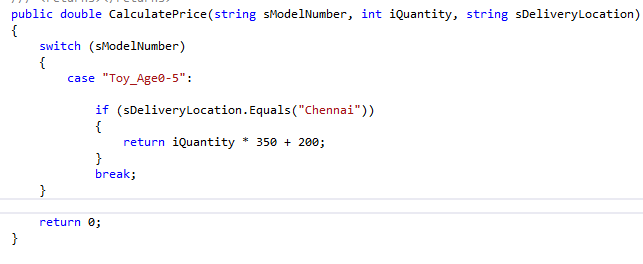
As a first step we need to create a service interface through the WCF service library, You can find the same in New->Project->WCF->WCF Service Library. Once you create the project add an assembly reference System.ServiceModel then create an interface as like the below snapshot

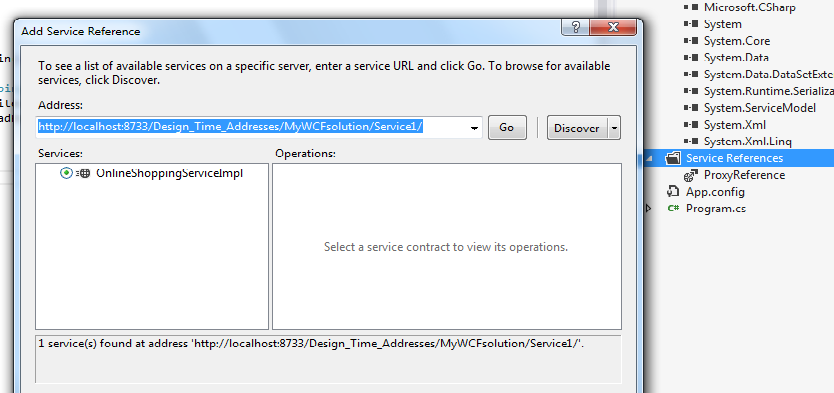


As I said earlier just put the [ServiceContract] attribute over the interface and the [Operation Contract] over the service method, This will include this service operation in the WSDL document. If we didn’t mention the above attributes it will not be included in the WSDLdocument. This can also be called as opt in model. We should not combine the service interface and the implementation altogether, if we do so, we are breaking the rule of decoupling. Decoupling is none other than abstracting the service definition from the concrete implementation . Now we have done our part in defining the service,next we need to implement the defined service interface.

# How should we need to implement it?

Create an another project and implement the service interface what we created ,then you can do your own implementation for each and every interface method. Following is the snapshot of implementation

 Once you have done your implementation, it is the time to think about Hosting the implemented service. Hosting is the bigger concept am not going to explain here, also it is irrelevant to this topic, you can follow my future articles in WCF, I will give you all a detailed view on this. So far we have a defined a service interface and implemented it. Now we will see how to utilize it on the client side. The first thing you need to do here is open the service reference and enter the service address to discover it. Make sure your host is running to get discovered.



# Based on the service reference added in the project, app. config will be

# created automatically with the address, binding and contract information. As a next step

# ceate a proxy object in the client to communicate the server

# clientSnapshot.png

# Extracting Service MetaData

# As a told you earlier WSDL file has to be generated for all types of clients to make use of

# the service. WSDL document contains method definitions and service operations. To get

# the WSDL document of a service you have to use command SVCUtil.exe from the visual

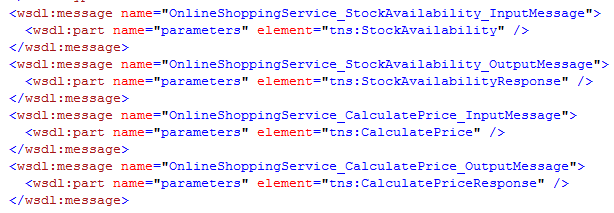
# studio command prompt. Following is the syntax has to be used to generate the WSDL

# document

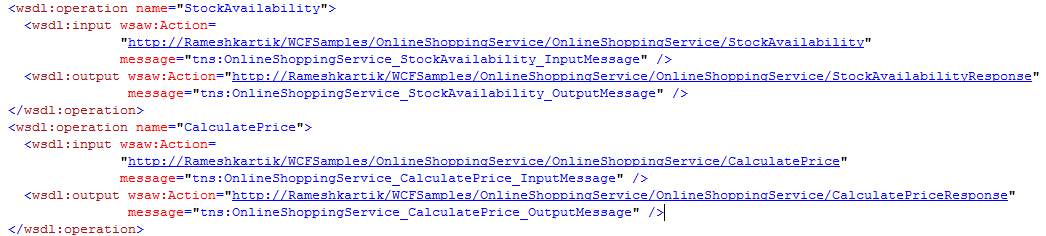
# SVCUtil.exe OnlineShoppingService.DLL [DLL Name of the WCF Service Library]

# In the generated file, you can find the service method definitions,Following snapshot is

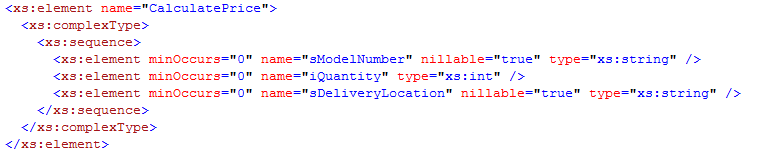
# part of WSDL document showing the method definitions



Another part of the document showing the operations available in the service.



Apart from that, XSD schemas also be generated for the method attributes and the .NET Types. Refer the below attachment you can find the name of data types with the sequence



# Attachment

Refer the code attachment and wsdl document,xsd schemas for further details

# Summary

[Service Contract] and the [Operation Contract] attributes are the foremost attributes

to make the service method as accessible to the outside world