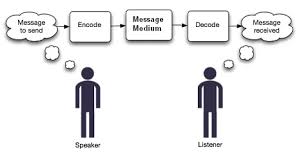
**Message Contract**

****

**Rameshkartik.RS**

**Table of Contents**

[Message Contract 3](#_Toc388006312)

[Why Message Contract? 3](#_Toc388006313)

[Source Code Explanation 5](#_Toc388006315)

[Attachment 7](#_Toc388006319)

[Summary 7](#_Toc388006320)

**WHAT’s IN THIS ARTicle?**

* Message Contract
* Why Message Contract?
* Source Code Explanation
* Attachment
* Summary

# Message Contract

As I told you all in the last article, Data Contract is all about the structure of the data being exchanged between the client and server, where as Message contract is all about the structure of the message being exchanged, By default WCF takes care of creating SOAP messages according to the data and service contracts,But Sometimes we need a complete control over the content of SOAP messages (Header/Body) to provide an additional layer of security. In this case message contract plays a vital role. Ok Lets discuss about the message contract benefits with a source code example.

# Why Message Contract?

In the following instances Message contracts can be used

Session Information

Sometimes your project requires the session information has to be exchanged between the client and server to provide an extra layer of security, so it can be passed in SOAP headers rather than adding additional parameters to operations or adding this information as fields in the data itself.

Security

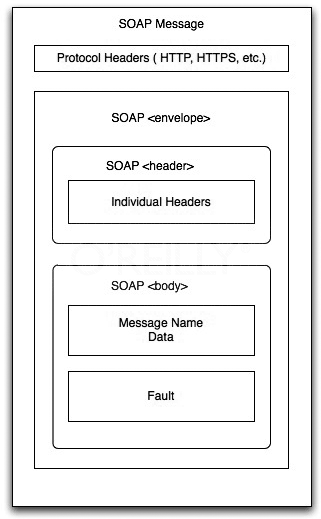
# Sometimes you like to have a custom security protocol (by passing WS-Security) and pass credentials or tokens in custom SOAP headers. Also, when you like to encrypt the header information message contract can be used. The downside with this technique is that the client and the service must manually add and retrieve the information from the SOAP header, rather than having the serialization classes associated with data and operation contracts do it for you

Entity Changes often

In case of MessageContracts in case of modification in data structure just need to change the message contract and no need for the regenerating proxies. In case of change in DataContracts you need to regenerate the proxy. So MessageContract is the good choice if your entity changes often.

# Source Code Explanation

# For your quick reference here am attaching the structure of the SOAP message



If you want more information on the SOAP topic, please refer my article Service Oriented Apporach(SOA). Ok.. Lets discuss How Message contract is implemented?

[MessageContract(IsWrapped=false)]

public class OrderRequest

{

private int iOrderID;

[MessageHeader]

public int OrderID

{

get { return iOrderID; }

set { iOrderID = value; }

}

}

# As mentioned above, Message Contract attribute has to be added on the top of class as we do for the Data Contract. OrderRequest is the MessageContract attributed class communicating back and forth with a SOAP message. OrderID is attributed as SOAP Message header which holds the order id of a customer.

public OrderResponse GetOrderHistory(OrderRequest OrderReq)

{

if (OrderReq.OrderID.Equals(OrderID))

{

OrderResponse response = new OrderResponse();

OrderInfo OI = new OrderInfo();

OI.OrderId = OrderID;

OI.OrderDate = DateTime.Now;

OI.PaidByCard = true;

OI.PurchasedProductName = "Barbie Doll";

OI.Quality = 5;

response.OrderInfoObject = OI;

return response;

}

else

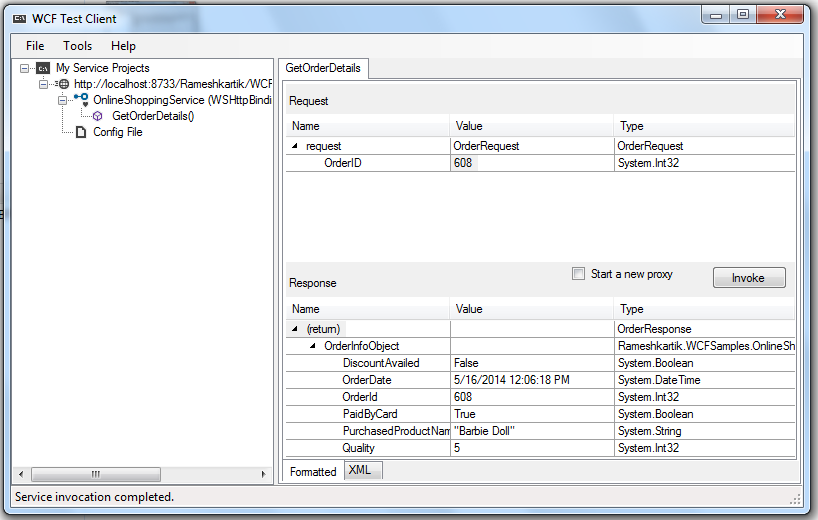
{

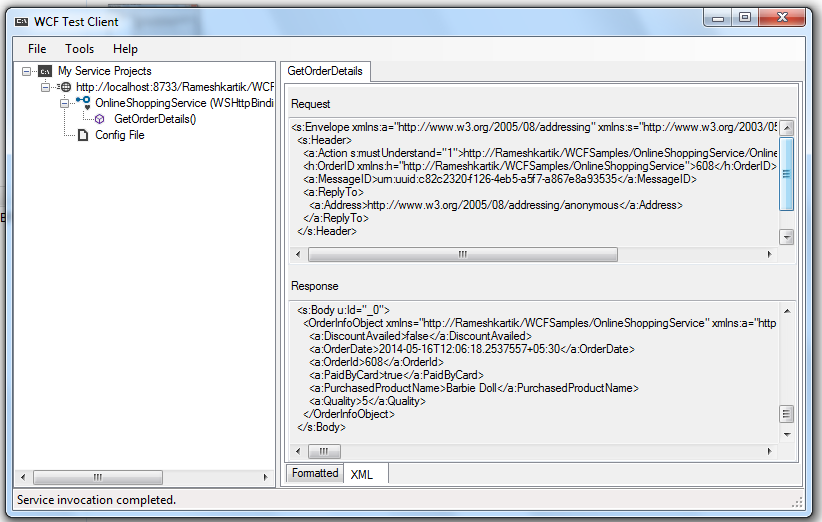
throw new FaultException<string>("Order Detail Not Found",new FaultReason("Order Detail Not Found"));

}

}

# In our case OrderRequest class is used for passing the input parameters to a service method GetOrderHistory. The OrderResponse is also a MessageContract attributed class which returns the SOAP response. The following picture shows you the SOAP request and the response





# Attachment

Refer the code attachment for further details

# Summary

Session related implementations can use message contract to get a complete control on soap messages,Also if your entity is keeps changing you can make use of MESSAGE CONTRACT