WCF

BY KannaBabu

New Batch

4/2/15

Homogeneous Application:-

It is used to create the communication b/w different application that was developed by using same language or same technology.

Java Java.

Heterogeneous Application:-

It is used to create the communication b/w different applications that was developed by using different languages or different technologies.

Java .net

web service is a distributed technology which is used to ~~develop~~ or develop interoparable applications

Q) what is 'interoparability?

It is a process of creating the communication b/w different applications that was developed in same language or different Languages or technologies.

Q) what is SOA?

SOA is "Service oriented Architecture".

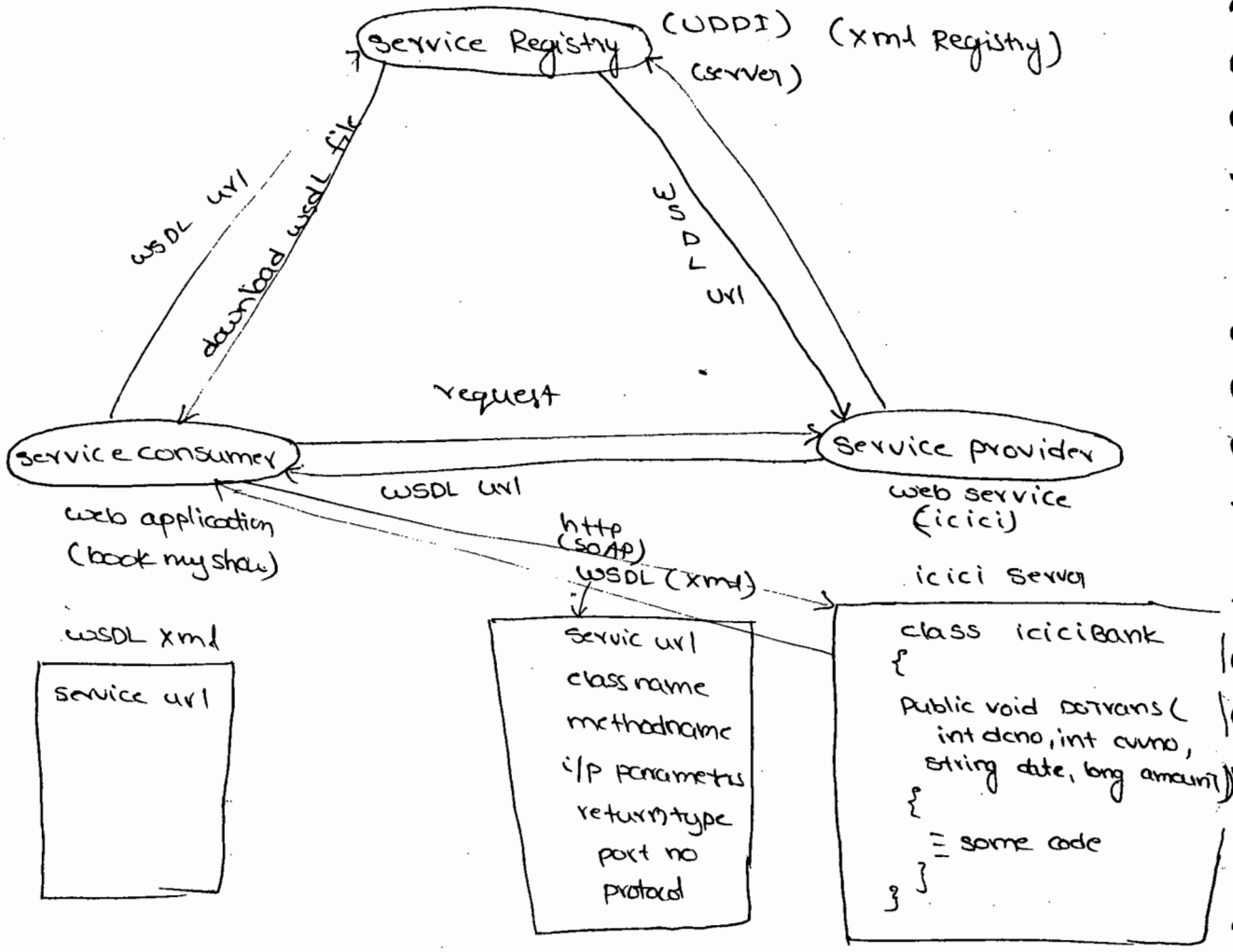
5/2/15

SOA (Service Oriented Architecture) :-

SOA is an approach that was followed by w3c which is used to develop distributed application.

w3c :- world wide web consortium

Architecture of SOA :-



Service provider :-

* Service provider will create the webservice
* web service can be created in any Language or in any technology and can consume in any Language or any technology

* So webservice is Language independent i.e, if we create the service in java we can consume the service in .net vice versa

Service Registry :-

* It is an XML based registry which is used to register all the WSDL file.

WSDL - Web Service Description Language

UDDI :- Universal Description descovery Integration

UDDI is platform-independent, Extensible markup Language (XML) based registry by which bussiness world wide can list themselves on the internet, and a mechanism to register and locate web service application

Service Consumer :-

It is a web application which will consume the webService

The web application can be developed in any language or in any technology.

Steps to understand the SOA Architecture :-

1. Service provider will create the web service
2. Service provider will host the service on Server
3. Service provider will create WSDL document

WSDL - web service Description Language.

→ WSDL is a document which was developed by using XML which is used to describe the webservice.

i.e, with in the WSDL file the entire information of the webservice will be available.

Like; service url, class name, method name, input parameters, returntype, port no, protocol information etc.

4. Service provider will host the WSDL file in UDDI

   UDDI is a place where all the WSDL files are registered. UDDI will give the WSDL url to service provider

5. Service consumer will give request ~~for service provider~~ to consume service. then service provider will give WSDL url to service consumer.

6. Service consumer will connect to UDDI by using WSDL url and download the WSDL file.

7. Service consumer must able to understand WSDL file in order to consume the service. ~~to~~ So he must know XML

8. By seeing the service url in WSDL file service consumer will give the request to the server and access webservice.

9. The communication blw Service consumer and webService is due to http protocol.

   So http protocol is called as communication protocol and it is stateless protocol.

10. Here the message transfer blw service consumer and server is by using SOAP protocol.

SOAP - "Simple object Access protocol"

SOAP is message transfer protocol.

Note :-

* In web application the client is browser but in webservice the client is web application.
* web application can be consumed by user but webservic must be consumed by developer.
* The person who will create the service and who will consume the service is developer.

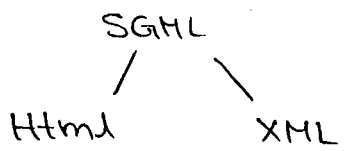
6/2/15

XML (Extensable Markup Language) :

* Initially IBM has introduced a language called GML (Generalised markup Language) for internal communication.
* Later GML was accepted by w3c (worldwide web consortium
* w3c has added some standards and released SGML standard.

  SGML - Standard Generalised Markup Language

* Under SGML languages was introduced



Html	XML
1. It is used to display the data	1. It is used to describe the data.
2. Html supports both container tags & non container tags	2. XML supports only container tags.
3. All Html tags are pre-defined	3. All XML tags are user defined.
4. Html is case insensitive Language	4. XML is case sensitive language
5. Html must be saved with .htm (or) .html	5. XML must be saved with .XML.
6. Html is markup language	6. XML is meta markup language.

Note :-

1. Both Html and XML are tag oriented languages.
2. the output of html & XML will be on browser.

Q). what is a tag?

Any command i.e, enclosed with in angular brases is called as tag.

Ex:- < cmd > .

Tags are of two types

1. container tag : the tag which contains both starting element and ending element are called as container tags.

   Ex:- <html> </html> — predefined (html)

   <Employee> </Employee> — userdefined (XML)

2. Non-container tag :- The tag which contains only starting element but not ending elements are called as Non-container tag.

* XML does not support non-container tags

  Ex:
, <hr>

* XML is a universal markup language which is used to transfer the data.

* XML can be understable by any language or any technology or any database. So XML is platform independent and language independent.

XML documents are of two types

1. well formed XML
2. Valid XML (DTD, XSD).

1. well formed XML :- The XML document with correct syntax is called as wellformed XML.

2. valid XML :- the wellformed XML must validate with DTD or XSD is called as valid XML.

Note :-

Every wellformed XML may not be validated but every valid XML must be welformed.

Rules to create the welformed XML document :-

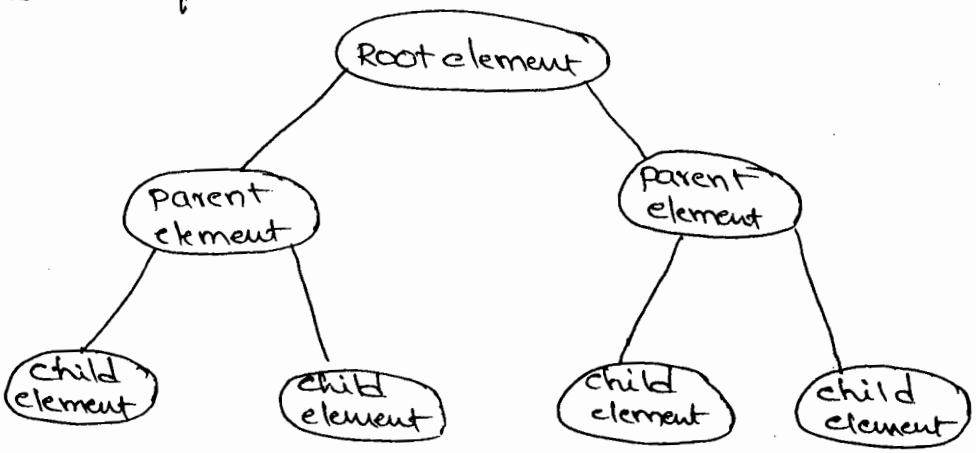
1. Every XML document must have root element.
2. Every Starting element must have ending element.
3. XML tags are case sensitive
4. XML tags must properly nested. i.e, in which order will open the tags in the same order we have to close the tags.
5. XML attribute value must be quoted in double quotes.

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

* If any XML document was created by using the above 5 rules then that document is called as welformed XML.

7/2/15

Structure of XML document :



* We can create XML document either in notepad or in visual studio editor

Ex -

```
<?xml version="1.0" encoding="utf-8" ?>
                                  ↳ unit text format.
<RootElement>
  <ParentElement>
    <ChildElement1> some data </ChildElement1>
    <ChildElement2> some data </ChildElement2>
  </ParentElement>
  <ParentElement>
    <ChildElement1> some data </ChildElement1>
    <ChildElement2> some data </ChildElement2>
  </ParentElement>
</RootElement>.
```

Employee XML document

```
<?xml version="1.0" encoding="utf-8" ?>

<Employees>
  <Employee>
    <Eno> 101 </Eno>
    <Ename> Nani </Ename>
    <Salary> 20000 </Salary>
  </Employee>
  <Employee>
    <Eno> 102 </Eno>
    <Ename> Balu </Ename>
    <Salary> 23000 </Salary>
  </Employee>
</Employee>
```

XML parser :-

XML parser is a program which is used to execute the xml document.

XML parsers are of two types

1. Non validating parser
2. validating parser.

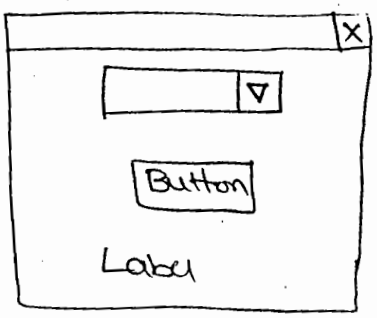
1. **Non-validating XML parser** :- It is used to execute wellformed XML document.

2. **Validating XML parser** : It is used to execute valid XML document

program to display the XML data with in the gridview Control

```
using system.XML;
using system.Data;
protected void page_Load()
{
    Dataset ds = new Dataset()
    ds.ReadXML (Server.Map path ("Employee.xml"));
    Gridview1.DataSource = ds;
    ds.Gridview1.DataBind();
}
```

Binding the XML file to dropdown list :-



```
using System.xml;
using System.Data;
protected void page_Load()
{
 if(IspostBack==false)
 {
    DataSet ds=new DataSet();
    ds.ReadXml(Server.MapPath("Employee.xml"));
    Dropdown List1.DataSource=ds;
    Dropdown List1.DataTextField="Ename";
    Dropdown List1.DataValueField="Eno";
  } Dropdown List1.DataBind();
}
protected void button1_click()
{
  Label1.Text="Emp no is" + DropdownList1.SelectedValue.
                                          ToString();
 }.
```

9/2/15

Valid XML :-

It means wellformed XML + DTD or XSD

DTD :- Document type Definition

DTD & XSD both are used to describe the XML document. i.e, the root element name, parent element name, child element names and what type of data i.e, available with in the XML document i.e, it will display the description of XML document.

DTD is of two types

1. Internal DTD
2. External DTD

1. Internal DTD:-

It is used to create the DTD file and XML document in a single file

Ex:-

```
<?xml version="1.0" encoding="utf-8"
<!DOCTYPE Employees[
  <!ELEMENT Employees (Employee)*>
   <!ELEMENT Employee (Eno, Ename, Salary)>
   <!ELEMENT Eno (#PCDATA)>
   <!ELEMENT Ename (#PCDATA)>
    <!ELEMENT Salary (#PCDATA)>
]>
<Employees>
  <Employee>
    <Eno> 101 </Eno>
    <Ename> Anil </Ename>
     <Salary> 20000 </Salary>
  </Employee>
  <Employee>
     <Eno> 102 </Eno>
     <Ename> sunil </Ename>
     <Salary> 23000 </Salary>
  </Employee>
</Employees>
```

DOCTYPE:-

It is used to mention the root element name.

Element:-

It is used to mention/maintain the element name.

*:-

It is used that more than one parent element is occured.

PCDATA:- passed character data

CDATA:- character DATA.

* Both PCDATA & CDATA are used to mention the type of data that we are storing in the XML document.

*. PCDATA will be validated by XML parser.
CDATA will not validated by XML parser.

*. PCDATA will not allows the character like "<" and "&" while CDATA will allow i.e, CDATA will allow any type of DATA.

2. External DTD:-

Creating a DTD file in a separate file and save with .dtd extension and importing the DTD in XML-file. is called as External DTD.

Ex:-

Employee.dtd → In notepad.

```
<!DOCTYPE Employees[
 <!ELEMENT Employees(Employee)*>
  <!ELEMENT Employee (Eno,Ename,Salary)>
  <!ELEMENT Eno (#PCDATA)>
  <!ELEMENT Ename (#PCDATA)>
   <!ELEMENT Salary (#PCDATA)>
]>
```

Save with Employee.dtd in E drive

```
<?xml version="1.0" encoding="utf-8"?>
 <!DOCTYPE Employees SYSTEM "file:///E:/Employee.dtd">
 <Employees>
```

10/2/15

XSD - Xml Schema Document

Differences b/w DTD & XSD

DTD	XSD
1. Document type Definition	1. XML Schma document.
2. DTD doesnot support namespace	2. XSD support name space
3. DTD was not developed based on XML	3. XSD was developed ~~by XSx~~ based on XML.
4. DTD doesnot support datatypes	4. XSD supports simple & complex datatypes.
5. DTD must be saved with .dtd	5. XSD must be saved with .xsd
6. DTD doesnot support user-defined data type	6. XSD supports user-defined data types.
7. DTD supports multiplicity operator i.e, it will mention the occurance of the element but if we want to declare ~~the~~ any element based on some specific element then dtd doesnot support	7. XSD ~~supports~~ will have two attributes minOcc$_x^{urance}$ and max occurance which will intimate the no.of times the element has appeared.

Q). what is XSD?

XSD is a file which will describe the structure of the XML document.

XSD is similar like dtd, but it is more powerfull than dtd

Q) while webService use XSD rather than DTD?

while working with web service the, at the time of communication the XML parser should verify not only the structure of the XML document. but also the datatypes, so we require a strongly typed language i.e, xsd.

* while creating xsd document it consists of two elements

1. simple element
2. complex element

simple element	complex element
1. It, will directly hold data	1. It can't hold data
2. It doesn't consists of subelements	2. It consists of sub elements
3. Simple element, does not have attribute	3. have attributes

* the root element for the xsd file is

```
<xs:schema> </xs:schema>
```

Syntax to declare simple element is

```
<xs:Element name="element name" type="xs:datatypename"/>
```

Ex:-

to Creating a simple element for the below XML document

```
<Employee>
 <Eno>101</Eno>
 <Ename>Anil</Ename>
 </Employee>

<xs:Element name="Eno" type="xs:int"/>
<xs:Element name="Ename" type="xs:string"/>
```

Sample xsd file for Employee.xml

Employee.xsd

```
<xs:schema>
 <xs:element name="Employee"/>
 <xs:complextype name="Employeetype">
 <xs:sequence>
  <xs:Element name="Eno" type="xs:int"/>
  <xs:Element name="Ename" type="xs:string"/>
  <xs:Element name="sal" type="xs:double"/>
  <xs:Element name="Dob" type="xs:date"/>
 </xs:sequence>
 </xs:complextype>
 </xs:schema>
```

* Sequence is used to mention the sequence of simple elements.

Ex:

```
<Student>
 <Sno>101</Sno>
 <Sname>anil</Sname>
 <age>22</age>
 <doj>2-3-2014</doj>
 <email>nani@gmail.com</email>
</Student>
```

11/2/15

Serialization & De-Serialization:-

object:-

→ object is instance of a class

→ Instance means allocating sufficient memory space for the instance variables that was declared with in the class

The data i.e, stored object is not persistent. i.e, the data that we will store in the object is temporary until the program is running on Ram.

Q). what is object persistency?

object persistency is a process of storing the object data perminently in a file or database or a separate memory

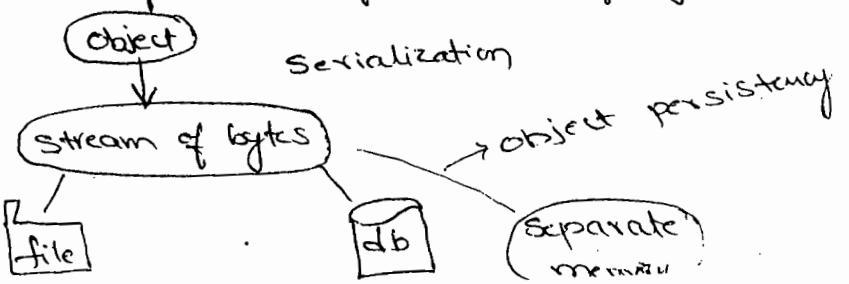
we can't directly store object data into file or database. If we want to store object data we have to convert object into stream of bytes.

Q). what is Serialization?

It is a process of converting object into stream of bytes

Q). what is de-Serialization?

It is a process of converting stream of bytes into object



Serialization is of 3 types.

1. Binary Serialization
2. XML Serialization
3. SOAP Serialization.

1. Binary Serialization:-

It is a process of converting object into stream of bytes and store the object data in the binary format is called as binary Serialization.

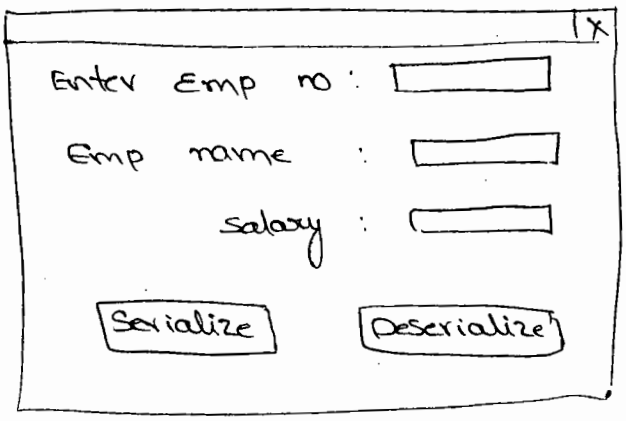
→ In order to work with binary Serialization we have to declare the name space "using System.Runtime.Serialization.Formatters.Binary".

→ under this name space we are having a predefined class with name "Binary formatter".

→ under Binary formatter class we have two methods

1. Serialize(): This method is used to convert object data into stream of bytes and store in binary format.
2. DeSerialize(): This method is used to convert stream of bytes into object.

Ex:-



```
using System.Runtime.Serialization.Formatters.Binary;
using System.IO;
Public partial class Form1:Form
{
  BinaryFormatter b=new BinaryFormatter();
   Filestream fs;
   Employee e1;
   private void button1_click()
   {
     int eno=int.parse(textBox1.Text);
      string ename = textBox2.Text;
       double sal = int.parse(textBox3.Text);
       e1 =new Employee(eno,ename,sal);
       fs =new Filestream("D:\\Emp.txt", Filemode.Create,
                                          FileAccess.write);
        b.Serialize(fs,e1);
        fs.Flush();
        fs.close();
        messageBox.show("Object is serialized");
  }
  Button2_click()
  {
    fs=new Filestream("D:\\Emp.txt", Filemode.Open, FileAccess
                                                    Read);
     e1 = (Employee)b.serialize(fs);
     textBox1.Text = e1.eno.ToString();
      textBox2.Text = e1.ename;
      textBox3.Text = e1.salary.ToString();
       messageBox.show("Deserialized");
     }
```

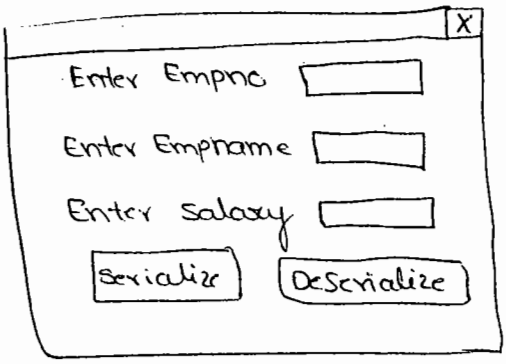
goto → project → Add class

```
[Serializable]
class Employee
{
    public int cno;
    public string ename;
    public double salary;
    public Employee(int x, string y, double z)
    {
      cno = x;
      ename = y;
      salary = z;
    }
}
```

12/2/15

2. XML Serialization :-

It is a process of converting object into Stream of bytes & store the object data in a XML format.



code for Employee.s

```
public class Employee
{
  public int Empno { set; get; }
  public string Empnam { set; get; }
  public double salary { set; get; }
}
```

code for Form1_Load(

```
using system.xml.Serialization;
using system.IO;
public partial class Form1:Form
{
    Employee el = new Employee();
    XmlSerializer xs;
    private void button1_click()
    {
```

```
        e1.Empno = int.parse(textBox1.Text);
        e1.Empname = textBox2.Text;
        e1.Salary = double.parse(textBox3.Text);
        XS = new XmlSerializer(typeof(Employee));
        StreamWriter SW = new StreamWriter("D://Emp.xml");
        XS.Serialize(SW, e1);
    }
    private void button2_click()
    {
        XS = new XmlSerializer(typeof(Employee));
        StreamReader SR = new StreamReader("D://Emp.xml");
        Employee e1 = (Employee) XS.DeSerializer(SR);
        textBox1.Text = e1.Empno.ToString();
        textBox2.Text = e1.Empname;
        textBox3.Text = e1.Salary.ToString();
    }
```

3. SOAP Serialization :-

It is a process of converting object into stream of bytes and store the object data in a file in the form of SOAP format.

```
using System.Runtime.Serialization;
using System.IO;
Public partial class Form1 : Form
{
    Employee e1 = new Employee();
    Private void button1_click()
    {
        SoapFormatter S = new SoapFormatter();
        FileStream fs = new FileStream("D://Employee.xml",
                FileMode.Create, FileAccess.Write);
```

```
e1.EmpNo = int.Parse(textBox1.Text);
e1.EmpName = textBox2.Text;
e1.Salary = int.Parse(textBox3.Text);
s.Serialize(fs,e1);
MessageBox.Show("Serialized");
}
```

13/2/15

SOAP (Simple object Access protocol) :-

* Soap is a protocol for accessing web services.
* Soap is message transfer protocol.
* Soap is a format for sending messages via network
* Soap communicates with internet.
* Soap is platform independent & language independent.
* Soap is developed based on xml.
* Soap is simple & extensable.
* Soap allows you to get around firewalls.
* soap is a w3c recommendation.
* In webservices the communication b/w service consumer and service is by using http protocol. and the message transfer is due to soap protocol. i.e, http is communication protocol. & Soap is message transfer protocol.
* Soap provides away to communicate b/w applications running on different operating systems with different technologies & programming languages.

The structure of Soap:-

```
<soap:Envelope>
  <soap:Header>
  <soap:Fault>
  ------
  <soap:Fault>
  <soap:Body>
```

```
<soap:Envelope>
  <soap:Header>
  ------
  </soap:Header>
  <soap:Body>
  <soap:Fault>
  ------
  </soap:Fault>
  </soap:Body>
  </soap:Envelope>
```

14/2/15

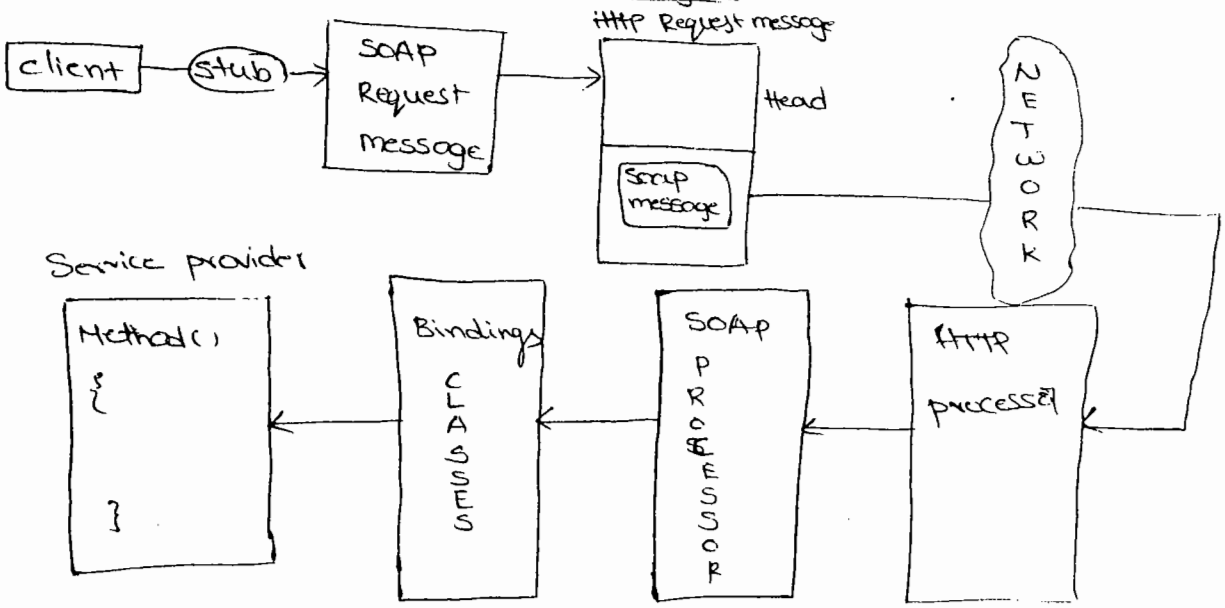
SOAP Envelop:- It is the root element for the soap message

SOAP Header:- It consist of the application specific information like Authentication details, payment details etc about the soap message.

SOAP Body:- this element consists of the actual soap message that we want to pass via internet.

SOAP Fault:- this element consists of the Exception Details

SOAP Request flow Diagram



Steps:-

1. Client Application will download the WSDL file. and with in the wsdl file by seeing the service url, client application will send the request to the server.

2. with in the client Application a stub object is created and this stub object will serialize the request object into soap request message.

Note:- Serialization is a process of converting object into stream of bytes (soap Serialization)

3. As we know that http is a communication protocol which is used to transfer the data.

Http protocol is of 2 types

1. Http Header 2. Http body

1. Http Header:- It consists of information about the request message

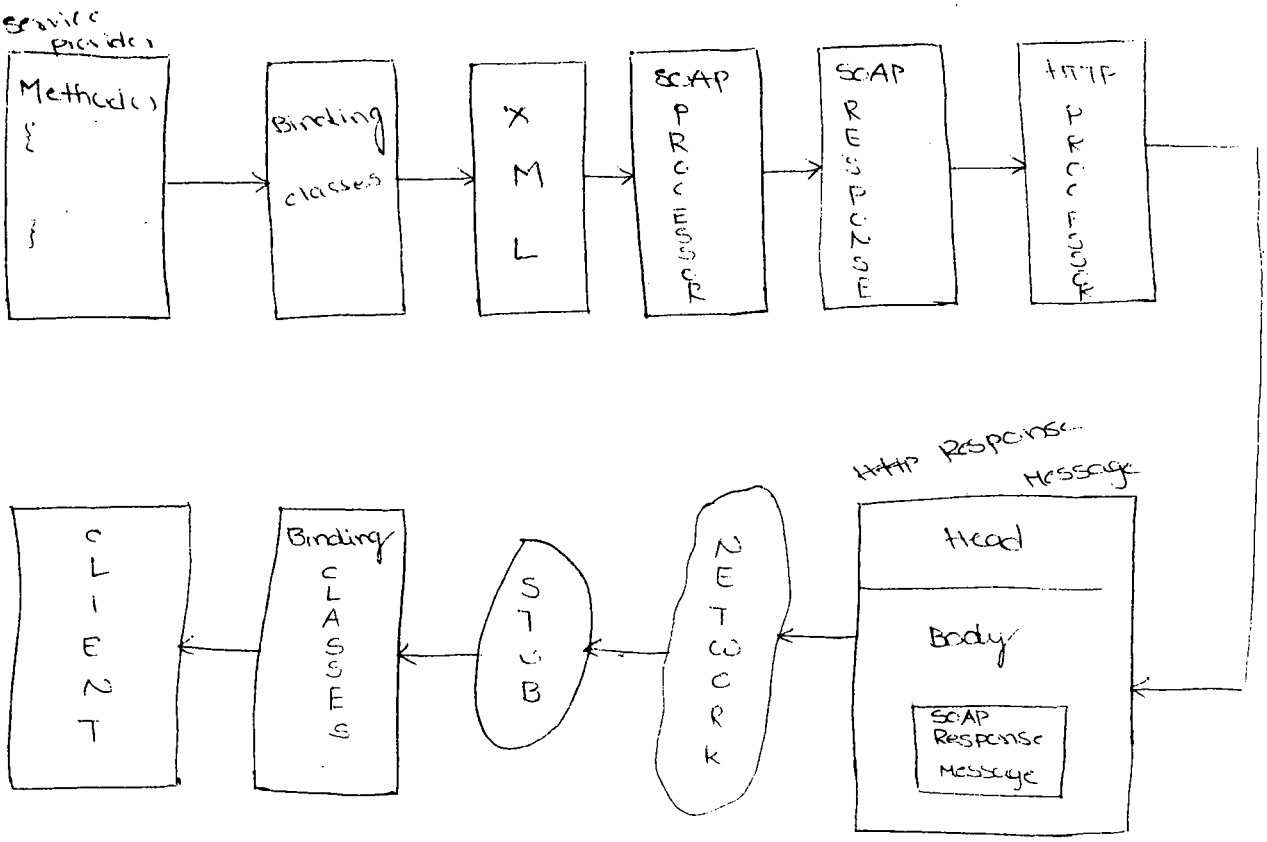
The soap request message was inserted into Http body and send via network.

4. with in the Server side Http process will recognise that a soap message was inserted into Http body and send the soap message to soap process

5. Soap processor will forward the soap message to Binding classes

6. Binding classes will deserialize (convert soap format to object) the soap message and give the request object to server.

Response flow Diagram of SOAP



Step-1 :-

Server will process the request object and it will generate the response object

2. The response object was given to binding classes
3. The binding classes will Serialize i.e, the response object will convert into soap (xml) and given to soap processor.
4. Soap processor via Http processor will insert the soap response object in Http body part.
5. Http protocol will send the soap response object via network to the client Application

6. With in the client Application the Stub object will understand the response object is in Soap format and it will forward the Soap response object to Binding classes

7. Binding classes will deserialize the Soap response object and convert into object and given to client Application.

16/2/15

Q) what is a webservice?

→ A web service is a method of communication b/w interoparable applications via internet

→ web services are of two types

1. Soap web Service
2. Restfull Service

*. Soap web services can be developed by using 2 Approach

1. Topdown Approach
2. Bottomup Approach

Topdown Approach :-

In this approach we have to first prepare the contract i.e, (WSDL file) and based on the wsdl file we have to prepare the service. i.e, contract first and service last.

Bottomup Approach :

In this approach first we have to prepare the service and based on the service we have to prepar WSDL file. i.e it follows Service first and contract Last.

*. In Soap web service wsdl file will play the major role

*. wsdl is an XML file which will advertise the Service that was created by service provider.

* By using WSDL file we can locate the web Service.

* WSDL file consists of 6 sections.

1. <Definition>
2. <Types>
3. <Messages>
4. <port>
5. <Bindings>
6. <Services>

Services – classnam, Serviceurl

porttype :- method name that are available in service class and we can

1. <Definition> :- As wsdl file is an XML document. the root element for wsdl file is definitions tag.

* If the wsdl file consists of definitions, types, messages, port then that wsdl file is called as Abstract wsdl file.

* If the wsdl file consists of All the 6 sections then that wsdl file is called as concrete wsdl file

2. <Types> :- This section of wsdl file consists of the simple types and complex types declaration for input & output parameters of webService methods.

```
class webservice
{
    public String getproducts(product p)
    {
        return "some message";
    }
}
```

```
class product
{
  int pid;
  string pname;
  int qty;
} double price;
```

```
<types>
<xs:schema target namespace:" ">
 <xs:complex type name:" product">
  <xs:sequence>
   <xs: element name:" pid" type="xs:int"/>
   <xs: element name:"pname" type = "xs:string"/>
   <xs: element name:"qty" type:"xs:int"/>
   <xs: element name:"price" type:"xs:double"/>
  </xs:sequence>
  </xs: complextype>
  </xs:schema>
</types>.
```

17/2/15

<portT>:-

→ In this section we can identify the method names.

→ The method name in wsdl file can be represented with operation element.

Syn:-

```
<wsdl:porttype name:"classname">
  <wsdl:operation name:"methodname"/>
  </wsdl:porttype>
```

Ex:-

```
<wsdl:porttype name:"product">
  <wsdl: operation name:"Get product"/>
  </wsdl:porttype>
```

<messages> -

In this section for every method 2 messages will be created.

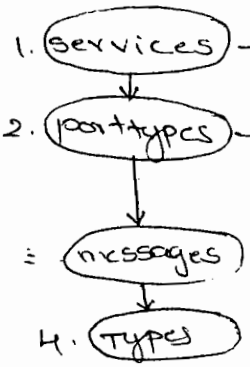
1. Input message
2. output message

For Ex:- If there are 3 methods in service class then 6 messages will be created. 3 i/p msgs & 3 o/p msgs will be created.

```
<wsdl: message name="methodnamesoapIn">
</wsdl: message>
<wsdl: message name="methodnameSoapout">
</wsdl: message>.
```

<Bindings> :

This section consists of the communication protocol information i.e, By using which protocol we are communicating with the service.



→ classname, Service url

→ method names, that are available in service class and we can identify the i/p and o/p message names

→ i/p message section

Example for bottom Approach

```
[web service]
public class productService : system.web.Services.webService
{
    [web method]
    public string Getproducts (product p)
    {
        return "Hello";
    }
}
public class product
{
    int pid; string pname;
    int qty; double price;
}
```

WSDL file :-

```
<wsdl:definitions>
 <wsdl:service name="product service">
  <soap:address location="http://localhost:2271/productsService.asmx"/>
 </wsdl:service>
 - - - - - - - - - -
 <wsdl:porttype name="productServiceSoap">
  <wsdl:operation name="Get products">
   <wsdl:input message="tns:GetproductsSoapIn" />
   <wsdl:output message="tns:Get productsSoapout" />
  </wsdl:operation>
  </wsdl:porttype>
 - - - - - - - - - -
 <wsdl:message name="GetproductsSoapIn">
  <wsdl:part name="parameters" element="tns:Getproducts" />
  </wsdl:message>
  <wsdl:message name="GetproductSoapout">
   <wsdl:part name="parameters" element="tns:GetproductReg ResponseRes" />
   </wsdl:message>
```

```
<wsdl:types>
<s:schema>
<s:element name="Getproducts">
 <s:complexType>
  <s:sequence>
   <s:element name="pid" type="xs:int"/>
   <s:element name="pname" type="xs:string"/>
   <s:element name="qty" type="xs:int"/>
  </s:sequence>
 </s:complexType>
 </s:element>
 </s:schema>
 </wsdl:definitions>
```

How to read the wsdl file.

1. goto service section identify the classname & service url (location)
2. goto port types and identify the method names that are available with in the service class. For every method two messages will be created "soapin and Soapout". Identify the input & output message names under porttype section.
3. goto messages section and with in the soapin message we can identify the input parameters and within soapout message we can identify the output parameters datatypes under types section.

4. goto types section and identify the simple element and complex elements and their datatypes.

5. goto Binding section and identify the protocol information.

18/2/15

Example to create webService and consume in web Applica

steps - 1: goto → file → new → project → select ASP.net empty web Application → Add

goto → project → Add → new item → Select web Service → name = employeeService → Add

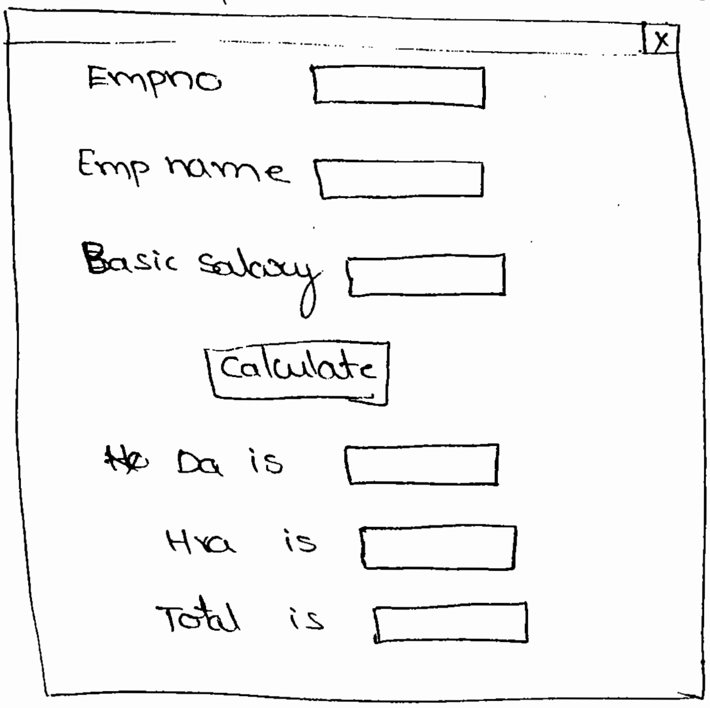
```
[web Service]
public class EmployeeService : system.web.Services.webServic
{
  [web method]
  public double calculateDa(double bsal)
  {
    return 0.2 * bsal;
  }
  [web method]
  public double calculateHra(double bsal)
  {
    return 0.4 * bsal;
  }
```

press F5 and Test the Service

open the Service description and read the WSDL file and copy the Service Url under "service" section.

```
<soap:address location="http://localhost:3820/EmployeeService.asmx"/>
```

goto → project → Add New item → select webform → name: "consumeempservice.aspx" → Add → Design Below page



goto → solution Explorer → rc on the project name → Add service reference → paste the service url in address bar → go → namespace = Servicereference1.

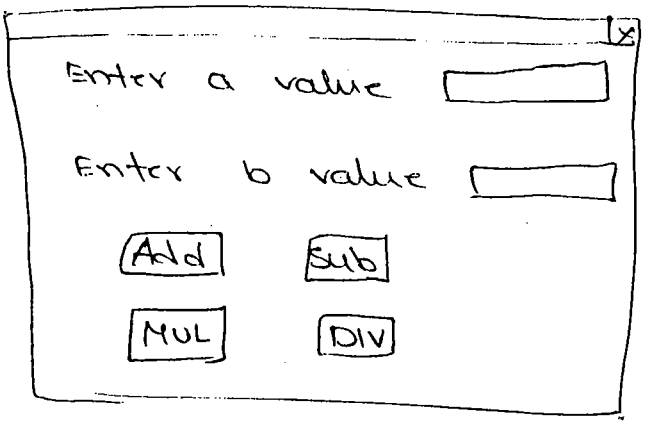
Here the namespace is used to consume the service i.e, in order to create an object for service class we have to use namespace.

goto → webform1.aspx and write the code.

```
using consumeempService.serviceReference1;
protected void Button1_click()
{
    EmployeeserviceSoapclient obj = new EmployeeServiceSoapclient();
    double bsal = double.parse(TextBox3.Text);
    double da = obj.CalculateDa(bsal);
    double Hra = obj.calculateHra(bsal);
    double Tsal = bsal + da + Hra;
```

```
TextBox4.Text = da.ToString();
TextBox5.Text = Hra.ToString();
TextBox6.Text = Tsal.ToString();
}.
```

press F5 and check the output.




```
class webservice
{
Public int Add(int a, int b)
{
  return x+y;
}
```

19/2/15

Create a Service to insert the record in employee table

Step

1. goto → sqlserver and create a table "eno ename salary"
2. create the webService

goto → file → New project → select Asp.net empty web application

goto → project → Add new item → Select web Service →

name = empservice.asmx → OK.

goto → web.config and declare the connection String

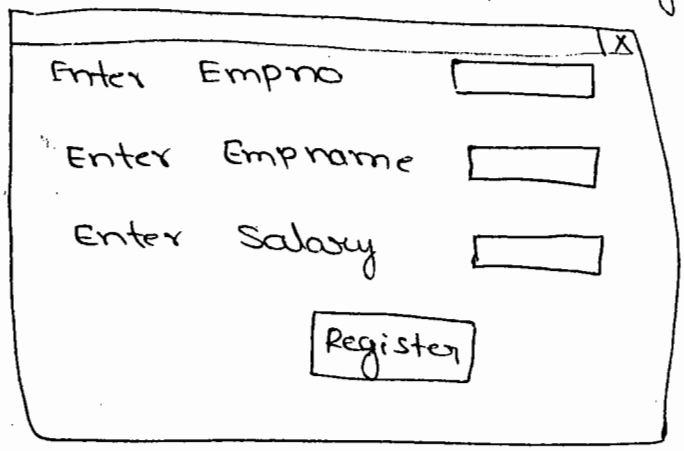
```
<connectionStrings>
<add name="constr" connectionstrings="User id=sa; password
    abc; database=nani; Data source=Nani-pc";
</connectionstrings>.
```

code for EmpService.asmx.cs :

```
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
[WebService]
public class Empservice:System.Web.Services.WebService
{
    [web method]
    public int createEmployee(Employee obj)
    {
        // create the connection
        SqlConnection con=new SqlConnection()
        // open the connection
        con.open();
        //pass the query
        string q="insert into emp values('"+obj.Eno+"',
                '"+obj.cname+"','"+obj.Salary+"')";
        SqlCommand cmd=new SqlCommand(q,con);
        int i=cmd.ExecuteNonQuery();
    } return i;
}
public class Employee
{
    public int Eno {get; set;}
    public string Ename {get; set;}
} public double Salary {get; set;}
```

press F5 and read the wsdl file and copy the Service url.

goto → webApplication.aspx(design)



goto → solutionExplorer → rc on project → Add Service reference → paste the url of the service and click on go.

```
using webApplication2.ServiceReference1;
protected void button1_click()
{
    Employee e1=new Employee();
    EmployeeServiceSoapClient obj=new EmpServiceSoapClient();
    e1.Eno = int.parse(TextBox1.Text);
    e1.Ename = TextBox2.Text;
    e1.Salary = double.parse(TextBox3.Text);
    int i=Obj.CreateEmployee(e1);
    if(i==1)
    {
      Label1.Text = "record is inserted";
    }
    else
    {
      Label1.Text = "insertion failed";
    }
}
```

21/2/15

working with live webServices i.e weather Service.

1. Type the below url with in the browser

   http://wsf.cdyne.com/weatherws/weather.asmx

2. goto Service description read the wsdl file and Test the methods. there are 3 methods

   1. GetcityForecastByzip:- this method is used to allows you to get your city forecast over the next 7 days, which is updated hourly. U.S only.

   2. GetcityweatherByzip:- Allows you to get your city's weather, which is updated hourly, U.S. only.

   3. GetweatherInformation:- Get information for each weatherID.

3. goto → Default.aspx(design)

   drag and drop image control and 9 Label Controls

   Enter zipcode : [ ] and one [button]

4. goto → solution explorer → rc on references → paste the url → namespace = Servicereference1 → OK.

```
using Servicereference1;

protected void button1_click()
{
    string zip= TextBox1.Text;
    Service Reference1.weatherSoapclient obj = new
                                    weatherSoapclient();
    Label1.Text= obj.GetcityweatherByzip(zip).City.ToString();
```

```
Label2.Text = obj.GetCityWeatherByZip(zip).Description.ToString();
Label3.Text = obj.GetCityWeatherByZip(zip).Pressure.ToString();
Label4.Text = obj.GetCityWeatherByZip(zip).RelativeHumidity.ToString
Label5.Text = obj.GetCityWeatherByZip(zip).Remarks.ToString();
Label6.Text = obj.GetCityWeatherByZip(zip).State.ToString();
Label7.Text = obj.GetCityWeatherByZip(zip).Success.ToString();
Label8.Text = obj.GetCityWeatherByZip(zip).Temperature.ToString(
Label9.Text = obj.GetCityWeatherByZip(zip).WeatherStationCity.
                                                    ToString();
int i = obj.GetCityWeatherByZip(zip).WeatherID;
 if(i==1)
 {
   image1.ImageUrl = "http://ws.cdyne.com/WeatherWS/Images/
                                          thunderstorms.gif";
 }
 if(i==10)
 {
   image1.ImageUrl = "      ";
 }
}
```

22/2/15

Example to work with currency converter web Service:-

1. Open www.webservicex.net click on currency convert
2. click on Service description
3. copy the Service url under Service section under Soap address.
4. goto default.aspx (design).

Enter from currency [ ]

Enter to currency [ ]

5. goto → solution Explorer → Add Service reference → OK

6. double click on button & write the code.

```
using ServiceReference1;
protected void Button1_Click()
{
    servicereference1.currencyconverterSoapclient obj =
            new currencyconverterSoapclient();
    String from = TextBox1.Text.Toupper();
    String to = TextBox2.Text.ToUpper();
    ServiceReference1.Currency cf = (ServiceReference1.
                                    currency) Enum.Parse
    (typeof(serviceReference1.currency), from);
    ServiceReference1.currency ct = (serviceReference1.
    (currency).Enum.parse(typeof(ServiceReference1.
                                    currency), to);
    Label1.Text = obj.conversionRate(cf, ct).ToString();
}.
```

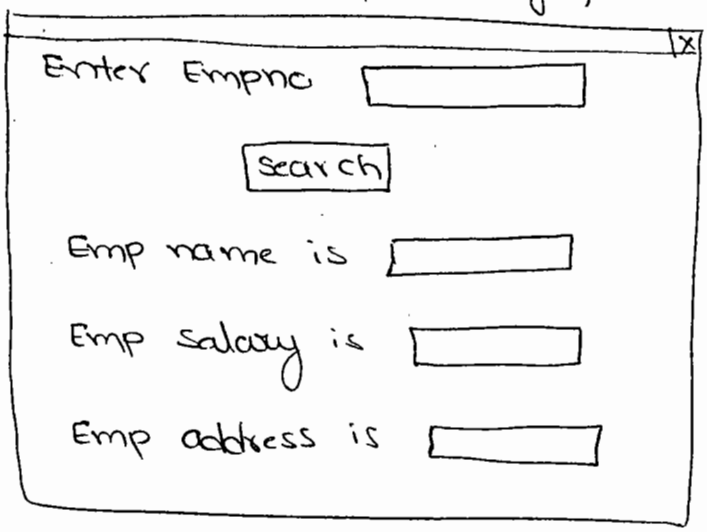
consuming the web service by using java script and Ajax:-

1. goto → sqlserver and create a table with name emp

eno	ename	Salary	address

2. goto → file → new project → select ASP.net Empty web Application → name = webApplication7 → ok.

goto → webform1.aspx (design)



3. goto → project → Add class → name = Employee.cs

```
public class Employee
{
    public int Eno { get; set; }
    public string Ename { get; set; }
    public double salary { get; set; }
    pubic string Address { get; set; }
}
```

goto → project → Add new item → select webservice → name = webservice1.asmx → ok

```
using system.data;
using system.data.sqlclient;
using system.configuration;
// to allow this web service to be called from script
  using Asp.net Ajax, uncomment the following line.
  [System.web.script.Services.ScriptService]
```

```
public class webService1 : System.Web.Services.WebService
{
    [WebMethod]
    public Employee GetEmployeeId(int id)
    {
        Employee el = new Employee();
        SqlConnection con = new SqlConnection();
        con.Open();
        SqlCommand cmd = new SqlCommand("proc_displaydata", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@eno", id);
        SqlDataReader dr = cmd.ExecuteReader();
        if(dr.HasRows)
        {
            if(dr.Read())
            {
                el.Eno = Convert.ToInt32(dr[0].ToString());
                el.Ename = dr[1].ToString();
                el.Salary = double.Parse(dr[2].ToString());
                el.Address = dr[3].ToString();
            }
        }
        con.Close();
        return el;
    }
}
```

press F5 and check the service

4. goto → webform1.aspx (Source)

```
<div>
    <asp:ScriptManager ID="ScriptManager" runat="Server">
    <Services>
        <asp:ServiceReference path="~/webService1.asmx"/>
    </Services>
    </asp.ScriptManager>
```

5. write the java script code with in the head tag

```
<script type = "text/javascript" language = "javaScript">
  language =*
   function GetEmployeeDataByid()
   {
      var id = document.getElementById("TextBox1").value
    webApplication7.webService1.GetEmployeeById(Id,
                GetEmployeeById Failed callBack);
   }
   function GetEmployeeByIdsuccess callBack(result)
   {
     document.GetElementById("TextBox2").value =
                                    result["ename"];
     document.GetElementById("TextBox3").value =
                                    result["salary"];
      document.GetElementById("TextBox4").value =
                                    result["Address"];
     }
     function GetEmployee ById Failed callback(errors)
     {
        alert(err.get_message());
      }
     </script>
```

2.3.2015

Differences b/w Remoting and webServices?

	.Net remoting	webService
protocol support	TCP, HTTP	HTTP
Data Format	Binary, SOAP	SOAP, Binary with some extra work.
Data types	Rich data types	Limited to datatype support
platform interoperability	client platform dependent .net application only	platform independent
Reliability	requires plumbing if not hosted on IIS.	Highly reliable hosted on IIS.
development	Complex	very easy
ease of use	complex	very easy
Audiences	Limited registered network and specified part only.	unlimited worldwide web.
Data Transfer	Faster	Slower
development & maintenance	complex Need to develop assemblies configuration files and object registration on the client and add reference to get the meta data of the assembly. Anychange in the code need to redeploy these files on the clients.	Easy Just add web referenc to the project using URL no deployment required.
Security	more secure - available thrugh Specified part only.	Less secure is proper precautions are not being taken.
scalability	Less scalable. only thrugh single call method but it may get costly if too	Highly Scalable

A new object will be created for each new client request

Host Application	windows forms, console Application, windows services and ASP.net application hosted by IIS and Asp.net		web Service only.
Application Dependency	Dependent. Needs a host Application on the Server to start the process.		Independent host.

26/2/15

Interface :-

* Interface is a contract or agreement blw itself and its implemented class.
* Interface will provide some set of specifications i.e. any class which is implementing the interface must obey the specifications.
* Interface is used to achieve multiple inheritence.
* By default interface members are public and abstract

Ex:-

```
using system;
Interface A
{
  void show();
}
interface B
{
  void show();
}
class C:A,B
{
  void A.show()
  {
  C.w.L("I am A show");
  }
  void B.show()
  {
  }C.w.L("I am B show");
}
```

```
    void B.Show()
    {
      C.W.L("I am B Show");
    }
}
class P
{
  Static void Main()
    {
      C C1 = new C();
      ((A)(C1)).Show();
      ((B)(C1)).Show();
    }
}
```

27/2/15

WCF (Windows Communication Faundation) :-

WCF is windows communication foundation which is used to develop distributed applications in .net

WCF = webServices + remoting

Step-1 :- Create the WCF Service

goto → start → run → devenv → OK

File → new project → select WCF under languages →

Template = webService Application → ok

Step-2 -

Go to solution Explorer → double click on iservice.cs and write the code.

```
namespace wcfservice4
{
    [Service contract]
    public interface IService1
    {
        [operation contract]
        int add(operation o)
    }
```

```
        [Data contract]
        public class operation
        {
            [Data member]
            public int A { get; set; }
            [Data member]
            public int B { get; set; }
            [Data member]
            public int c{ get; set; }
        }
    }
```

goto → service1.svc.cs and write the code

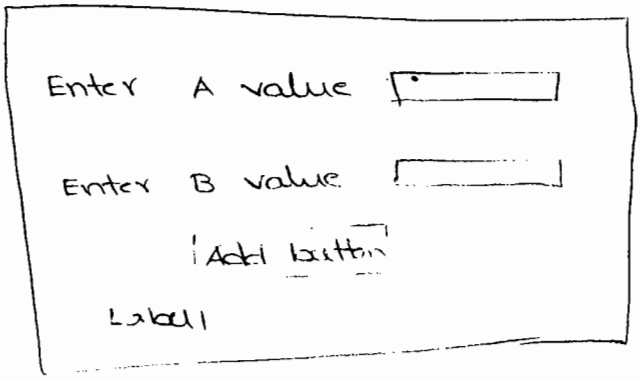
```
    public class service1 : Iservice1
    {
        public int add(operation o)
        {
            o.c = o.A + o.B;
            return o.c;
        }
    }
```

goto → solution explorer → rc on service.svc → view in browser, copy the url of the service

Step-3:- Goto → Start → run → devenv → ok

file → new → project → select Asp.net empty webapplication

goto → webform1.aspx.cs and write the code.



goto → solution Explorer → rc on references → Add service reference → paste the url.

namespace = ServiceReference1.

goto → webform1.aspx.cs and write the code

double click on button and write the code.

```
using webApplication1.ServiceReference1

protected void Button1_Click()
{
    operation o=new operation();
    o.A = int.parse(TextBox1.Text);
    o.B = int.parse(TextBox2.Text);
    ServiceReference1.ServiceIclient obj = new ServiceIclient();
    int result = obj.Add(o);
    Label1.Text = result.Tostring();
}
}.
```

28/2/15

In WCF the communication blw service provider and service consumer is through end point.

End point is the ABC's of the WCF service

A represents Address → Link (url)

B represents Binding → protocol

C represents contract → Interface.

Address :- It consists of the location of the WCF service like IP Address / servername i.e, whenever we develop a wcf service we have to host the service on the server and the service url will be available under address section of end point.

Ex:- http://localhost:1234/Service1.svc

net.tcp://localhost:4567/Service1.svc

msmq://localhost:4587/Service1.svc

Binding :- It is used to mention the protocol information i.e, How to communicate with the service

In order to communicate with wcf service we have to use protocol.

Different types of protocols are

1. Basic Http binding
2. nettcp binding
3. wshttp binding
4. msmq binding

contract :- contract is an agreement b/w wcf service and client application.

i.e, in w.c.f the interface will provide some set of specifications and the client application will consume the service.

with in the contract the interface name is mentioned.

Ex. for endpoint :-

```
<end point Address = "http://localhost:1234/service1.cs"
  Binding = "Basic http binding"
  contract = "IService">
</end point>.
```

→ while creating the service the interface must declare with "service contract attribute", if we not declare the above attribute then the interface cannot expose outside the world.

→ the methods that we write with in the interface must declare with "operation contract" attribute

→ the class where we declare properties must be declare with "Data contract attribute"

→ generally data contract class will act like the i/p parameters for operation contracts.

→ the properties that we declare within data contract class must declare with "Data member" Attribute

*. Remaining code in fb

11/3/15

Hosting mechanism in WCF :-

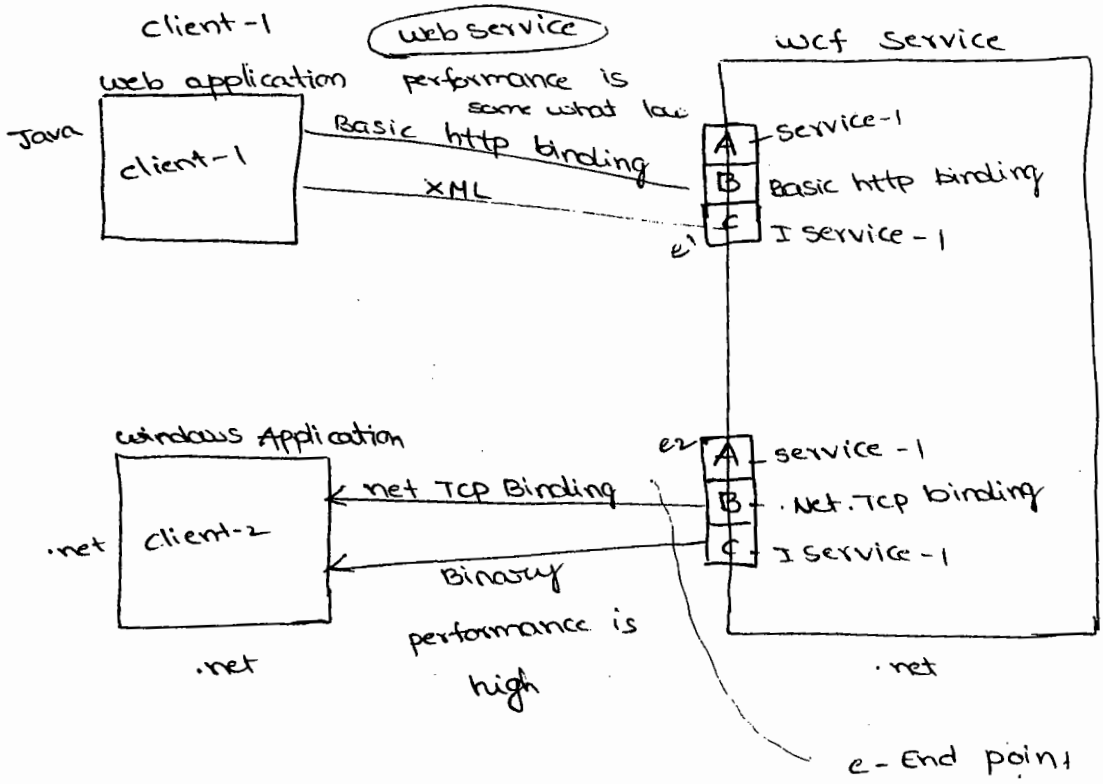
It is a process of placing the wcf service in application location in wcf. we can host the service in 3 ways.

1. self hosting
2. IIS hosting
3. WAS hosting

To host the WCF service we have 3 steps

1. create the service
2. host the service
3. consume the service in client application

Example for self-hosting :-



Example for selfhosting:

step-1:- file → new → project → select other project types
Visual Studio Solution → select blank solution
→ Name = DemoService → ok

R.c on solution add new project → select WCF →
Template = WCF Service Application → ok.

Create the service:-

code for IService.cs

```
[Service Contract]
public Interface IService1
{
    [operation contract]
    string Getmessage(string name);
}
public class Service1 : IService1
{
    public string Get message(string name)
    {
        return "welcome" + name;
    }
}
```

Step-2:- Host the service

R.c on solution name → Add new project → name =
service host → select console Application → Add → r.c on
Service host project → Add new item → select application
configuration file → name = App.config → Add

```
<configuration>
 <system.serviceModel>
  <services>
  <service name = "HelloService.Service1"
  behavior configuration = "mex behaviour">
   <end point address = "service1" binding = "basic http binding"
        contract = "Hello Service.IService1">
    </end point>
    <end point address = "service1" binding = "net TCP binding"
                contract = "HelloService.IService1">
     </end point>
    <end point address = "mex" binding = "mexHttp binding"
                 contract = "Imetadata Exchange">
     </end point>
    <host>
    <base address>
     <add base address = "http://localhost:8080/"/>
     <add base address = "net.tcp://localhost:8090/"/>
     </base addresses>
    </host>
  </service>
 </services>
  <behaviours>
    <service behaviours>
      <behaviour name = "mex Behaviour">
       <service metadata http Get Enabled = "true" /> </behavior>
     </service behaviors>
```

```
</behaviors>
</system.servicemodel>
</configuration>
```

within Service host project → rc on References →

Add reference → .net → system.servicemodel

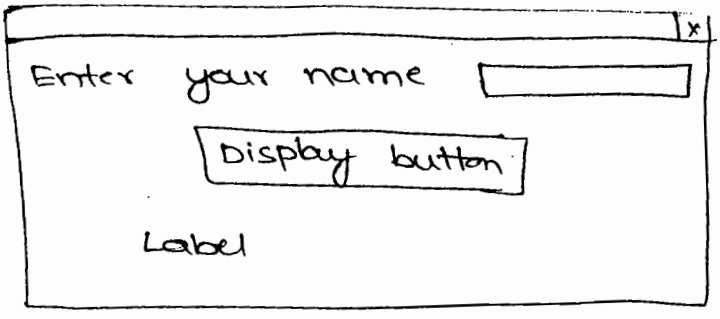
//y r.c on references → browse → select Hello.service → Add

goto → program.cs and write the code

```
using system.servicemodels;
class program
{
    static void Main(string[] args)
    {
        ServiceHost host=new servicehost(typeof(HelloServic
                                              Service1));
        host.open();
        c.w.L("Service is started" + system.DateTime
    }}      c.ReadLine();          Now.Tostring());
}
```

select servicehost → set as startup project and press F5 then the Service will Start.

consume the service in web application file → new project → Add new item → select web form → ok.



goto → solution explorer → Add Service reference →

paste the url → http://localhost:8080/ → go → ok.

Double click on button and write the code

```
using web application3.ServiceReference1;
protected void button1_click()
{
    ServiceReference1.Service1client obj = new
                                        Service1client();
    Label1.Text = obj.GetMessage(TextBox1.Text);
}
```

goto → web config file and remove net.tcp endpoint because client application cannot have more than one endpoint to communicate with Service

press F5 and check the o/p.

My consume the Service in windows forms application and remove endpoint1.

ch.rani999@gmail.com

9550497527

3/3/15

Example to host the service in windows applications and consuming multiple endpoints in a single client application

Step-1: create the service

goto → file → new project → select other project types → visual studio solution → select blank solution → ok.

Step-2: goto → solution explorer → rc on solution → Add new project → select wcf → name : finalmarksService → ok

create the service

code for IService.cs

```
[Service Contract]
public interface IInternalmarks
{
    [operation contract]
    int calculateTotalInternalMarks(int m1, int m2, int m3);
}
[Service contract]
public interface IExternalMarks
{
    [operation contract]
    int calculateTotalInternalMarks(int m1, int m2, int m3);
}
```

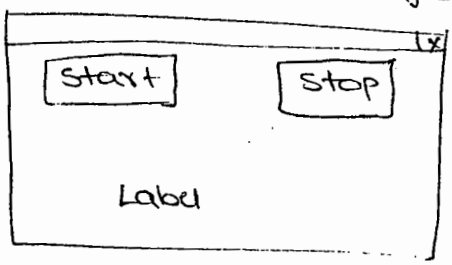
code for service1.cs:

```
public class service1 : Iinternalmarks, IExternalmarks
{
    public int calculateTotalInternalmarks(int m1, int m2,
                                           int m3)
    {
        return m1+m2+m3;
    }
    public int calculateTotalExternalmarks(int m1, int m2,
                                           int m3)
    {
        return m1+m2+m3;
    }
}
```

Step 3 :- Host the Service in windows Application

goto → solution explorer → rc on solution → Add new project → select windows forms application → name = servicehost1 →

goto → form1.cs[design]



Step-4 :- Create the end points in App.config file → rc on Service1 project → Add new item → select Application configuration file → name = App.config

```
<configuration>
  <system.servicemodel>
   <services>
    <service name="Finalmarksservice.Service1"
        behaviourconfiguration = "mexBehaviour">
     <endpoint address = "service1" binding = "BasicHttpBinding"
          contract = "Finalmarksservice.IInternalmarks">
     </endpoint>
     <endpoint address = "Service1" binding = "netTCPBinding"
          contract = "Finalmarksservice.IExternalmarks">
     </endpoint>
     <endpoint address = "mex" binding = "mexHttpBinding"
          contract = "ImetadataExchange">
     </endpoint>
     <host>
      <base addresses>
       <add baseAddress = "http://localhost:8081/" />
       <add baseAddress = "net.tcp://localhost:8091/" />
```

```
<behaviors>
 <serviceBehaviors>
  <behavior name = "mexBehaviour">
   <serviceMeteData  httpgetErabled = "True"/>
  </behavior>
 </serviceBehaviours>
</behaviors>
</system.servicemodel>
</configuration>.
```

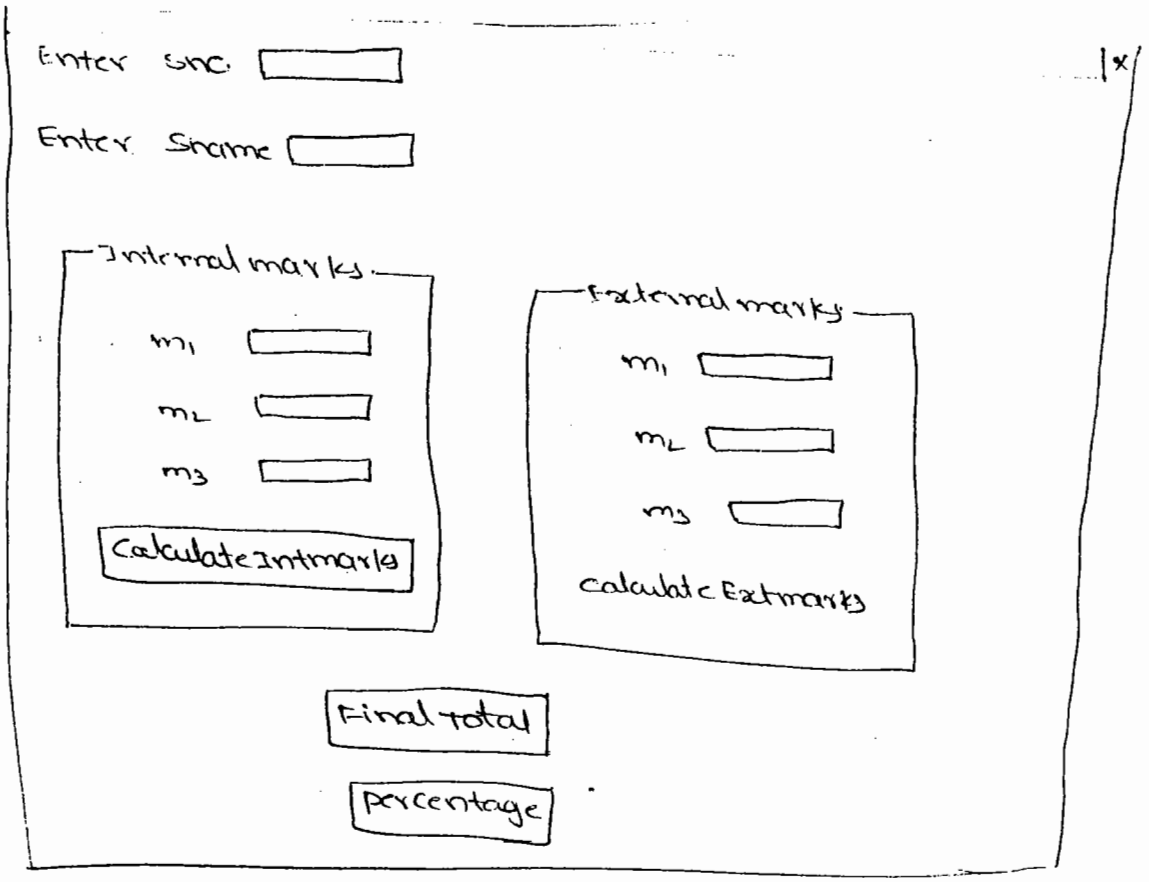
goto → Solution explorer → rc on Service host1
project → set as startup project → press F5

Step-5:- consume the service with multiple endpoints in windows forms applications.

double click on start button and start the service

```
using System.Servicemodel;
private void Button1_click()
{
   ServiceHost h=new ServiceHost(typeof(FinalmarksService.
                                        Service1));
    h.open();
    Label1.Text = "Service is started @" + System.DateTime.
                                        Now.Tostring();
}
 private void Button2_click()
 {
    ServiceHost h=new Servicehost(typeof(Finalmarksservice.
                                        Service1));
    h.close();
    Label1.Text = "service is stopped';
 }
```

→ Step-5:-



Step-6:- goto → solution Explorer → rc Add Service reference → http://localhost:8081/ and press go → ok.

```
using windowsapp1.ServiceReference1;

public partial class Form1 : Form
{
    int totintmarks;
    int totextmarks;
    int totfinalmarks;
    double percentage;
    private void button1_click()
    {
      ServiceReference1.IinternalmarksClient obj = new
          IinternalmarksClient("BasicHttpBinding_IinternalmarksClient");
      int m1 = int.parse(TextBox3.Text);
       int m2 = int.parse(TextBox4.Text);
        int m3 = int.parse(TextBox5.Text);
```

```
totintmarks = obj.calculatetotalInternalmarks(m1,m2,m3);
}
private void button2_click()
{
  ServiceReference1.ExternalmarksClient obj = new
   ExternalmarksClient("NetTcpBinding_Externalmarks");
    Int m1 = int.parse(TextBox6.Text);
    int m2=int.parse(TextBox7.Text);
    int m3 = int.parse(TextBox8.Text);
   totExtmarks = obj.calculatetotalExternalmarks(m1,m2,m3),
  }
}
```

4/3/15

windows Service :-

*. A windows service is similar to any other program or application that was running on windows machine.

* The differences b/w windows service & other application.

1. windows service runs in back ground
2. windows service was directly run by O.S.
3. They will automatically configure when, the O.S starts.
4. It doesnot require user interface.
5. All the windows Services are available under Services.msc

goto → Start → run → Services.msc → OK.

proxy

proxy is Local copy of the Service

Hosting the wcf service in windows service

1. Create the service

rc on solution add new project → select → wcf → Template = wcf Service application → name = myservice → ok.

create the Service

code for IService.cs

```
[Service Contract]
public interface IService1
{
    [Operation Contract]
    String DisplayMessage(String name);
}
```

code for Service.cs

```
public class Service1 : IService1
{
    public string DisplayMessage(string name)
    {
        return "welcome" + name;
    }
}
```

2. Host the wcf Service in windows Service

goto → solution explorer → rc on add new project → select windows → Template = windows Service → name = demoservicehost → ok.

goto → service1.cs(design) → rc → viewcode.

rc on demoService host project → references → Add reference → .Net → Add System.ServiceModel namespace

rc on reference add reference → projects → myservice → ok.

goto → Service1.cs and write the code

```
using System.ServiceModel;
public partial class Service1 : ServiceBase
{
    ServiceHost host
    host = new
    ServiceHost host;
    protected override void OnStart()
    {
        host = new ServiceHost (typeof (myservice.Service1));
        host.Open();
    }
    protected override void OnStop()
    {
        host.Close();
    }
```

3. goto → App.config file and write the code
   see the previous example

4. goto → service1.cs (design) → rc → Add installer → then 2 installers will be available
   → ServiceInstaller1
   → ServiceProcessInstaller1
   rc on ServiceInstaller1 → properties
   ServiceName = mydemoservice (this name will be displayed in service.msc)
   StartType = automatic
   rc on ServiceProcessInstaller1 → properties
   Account = LocalSystem → Build → build solution.

5 install the Service in service.msc

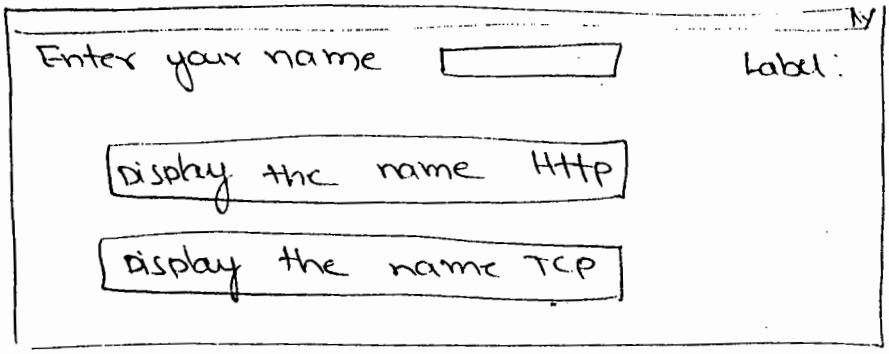
goto → visual studio command prompt →

Installutil -i pathof the service \demo.servicehost.exe

6. go and check in services.msc

demo.servicehost is available, rc start the Service.

7 consume the service in windows application.



8. consume the service we see the previous example

6/3/15

IIS Hosting :-

→ It is a process of hosting the wcf service under IIS

→ IIS is the web server for Asp.net ~~app~~ wcf applications

→ Before hosting the wcf service under IIS we need to install IIS.

Step-2 :- goto → start → run → type inetmgr → OK

if IIS was installed then a window with name Internet Information Services will be opend

Step-3 :- In order to work with IIS hosting we have to follow 3 steps

1. create the service
2. host the service under IIS
3. consume the service

Create the service

Step-4 : Start → run → devenv → ok

file → new → project → select otherproject types → visualstudio solutions → blank solution → name = solution1 → browse → D:\myService

Step-5 :- rc on the solution → Add new project

Select wcf service → ok.

```
namespace wcfservice1
{
    [service contract]
    public interface IService1
    {
        [operation contract]
        String displaymessage(string name);
    }
}
```

code for service1.cs

```
namespace wcfservice1
{
  public class service1 : IService1
  {
    public string displaymessage(string name)
    {
      return "welcome" + name;
    }
  }
}
```

build → build solution

Step 6: Host the Service

goto -> solution explorer -> rc on solution -> Add new website
-> Select wcf service -> Browse -> path = D:\myservice\iishost -> ok

Step 7:- when we add the service a folder with name App_code was created. delete IService1.cs and Service1.cs under Appcode folder.

note:- In IIS hosting it is not required to write the code to host the service.

goto -> servi rc on IIS host -> Add reference -> projects -> Select wcfservice1 -> ok.

goto -> IIS host -> Service.svc -> modify the code

<%@ ServiceHost Language = "C#" Debug = "true" Service = "wcfService.Service1" %>

Step 7:- prepare the configuration

goto -> web.config file under IISHost and write the code

```
<?xml version="1.0"?>
 <configuration>
  <system.servicemodel>
   <services>
    <servicename = "wcfService1.Service1" behaviorConfiguration=
                                          "mex behaviour">
    <end point address = "Service1" binding = "BasicHttpBinding"
                contract = "wcfService1.IService1">
     </endpoint>
    <end point address = "mex" binding = "mexHttpBinding"
                contract = "IMetaData Exchange">
     </end point>
```

```
<host>
<baseAddresses>
<add baseAddress = "http://localhost:8091/" />
</baseAddresses>
</host>
</service>
</services>
<behaviors>
<serviceBehaviors>
<behavior name = "mexBehaviour">
<serviceMetadata httpGetEnabled = "true" />
</behavior>
</serviceBehaviors>
</behaviors>
</system.serviceModel>
</configuration>
```

build → build solution

step-8:- create a virtual directory under IIS and host the service

goto → IIS → connections → sites → Default website → rc on default website → Add virtual directory → name = IISHost → physical path = D:\myservice\IISHost → OK.

rc on IISHost → convert to application →

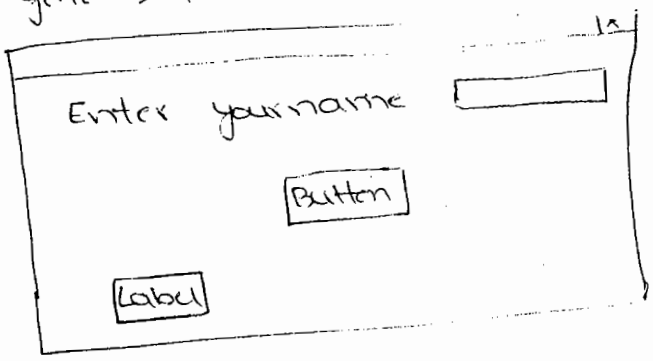
select applicationpool = Asp.net v4.0

rc on IISHost → switch to contentview →

rc on service.svc → Browse → it has to display the test service window and copy wsdl url.

Step 9:- consume the service in console / windows / web application

Step-10:- goto → Form1.xaml.cs (design)



goto → solution explorer → rc addservice reference → paste the url wsdl/url → ok

double click on button and write the code

```
using windowsFormsApplication1.ServiceReference1;

    private void button1_click()
    {
      ServiceReference1.Service1Client obj = new Service1Client();
      Label2.Text = obj.DisplayMessage(textBox1.Text);
    }
```

press F5 and check the output.

WAS Hosting :-

By default IIS supports only http protocol. If we want to use non http protocol like tcp we need to install windows communication fundation non-http activation under control panel

was hosting will support only in IIS7.0 and above versions. "windows vista/7/8".

To install non Http activation goto → start → run → control panel → programs and features → turn windows features on/off → IIS

+ microsoft .Net Framework 3.5.1

+ WCF Http Activation

+ WCF NonHttp Activation

goto → IISHost project → and add net.tcp binding endpoint → build solution → again run Service.svc under IIS.

goto → IIS → rc on IISHost → manage applications → Advanced settings → Enabled protocols = http, net.tcp → ok.

rc on Service.svc → Browse → It has to display the test service.

if still error occurs

run → visualstudiocommandprompt under administration mode → and type the below command "aspnet_regiis_iru" → enter → again run service.svc → ok.

7/3/15

Exceptions

* An Exception is a runtime error.
* Exceptions will occurs at the time of execution of the program
* CLR will identify Exceptions
* we cannot rectify the exceptions by using exception handling mechanism.
* we can handle the exceptions in .net in 2 ways
    1. Logical implementation
    2. Try-catch implementation.

generally wherever an Exception will occur in WCF Service then the Service will not throw the Exception details to the client application rather it will display a message saying that an internal error was occurred

Ex:- to handle the runtime errors in wcf we have to make use of fault Exceptions.

Q:) How to handle the exceptions in wcf?

By using fault Exceptions.

Ex:- code for IService.cs

```
[Service contract]
public interface IService1
{
    [Operation contract]
    int add(int x, int y)
}
```

code for Service.svc.cs

```
public class Service1 : IService1
{
    public int add(int x, int y)
    {
        try
        {
            return x/y;
        }
        catch (Exception e1)
        {
            throw e1;
        }
    }
}
```

Consume the Service in console application

goto → new → project → select console application → ok → add the service reference

```
using consoleapplication1.ServiceReference1;
class program
{
    static void Main()
    {
        try
        {
```

```
ServiceReference1.ServiceClient obj=new ServiceClient();
  console.writeLine("Enter a number");
  int x=int.parse(console.ReadLine());
   console.writeLine("Enter b number");
   int y=int.parse(console.ReadLine());
   int result=obj.Add(x,y);
  }
  catch(Exception e1)
  {
     console.writeLine(e1.message);
   }
   finally
   {
     Console.ReadLine();
   }
  }
}
```

observation :- when we execute the above client app if we give input as 10 and 0 then an Exception will occur at wcfservice instead of displaying the actual error it will display a error message saying that an internal error was occured at service so it is difficult for the clientapplication to identify what type of error massage occured to overcome the above program MS has given a new concept to

We can implement fault exceptions in 3 ways:

1. By using Service behaviour attribute
2. Service Debug tag in web.config file.
3. Fault contract
   goto → wcfService and modify the code.

```
[ServiceBehaviour(includeExceptionDetailInFaults = true)]
public class Service1 : IService1
{
    Public int Add(int x, int y)
    {
        try
        {
            return a/b;
        }
        catch(Exception e1)
        {
            throw e1;
        }
    }
}
```

Observation:- update the Service reference in client application and press F5 x=10, y=0. The instead of displaying internal error it will display divid by zero i.e the client appn can able to identify what type of exception has been occured.

2. goto → web.config file & write the code.

```
<ServiceDebug includeExceptionDetailInFaults="true" />
```

Fault contract :-

It is used to handle the exceptions at operation contract

Ex:- Code for iservice.cs

```
[Service contract]
Public interface IService1
{
    [Operation contract]
    [fault contract (typeof (Errordata))]
    int Division(int x, int y);
}
[Data contract]
public class ErrorData
{
    [Data member]
    public string operation { get; set; }
    [Data member]
    public string ErrorText { get; set; }
    [Data member]
    public string Reason { get; set; }
}
```

Code for service.cs

```
public class : IService1
{
    Public int Divison (int x, int y)
    {
        try
        { return x/y; }
```

```
        catch (Exception)
        {
            ErrorData e2 = new ErrorData();
            e2.operation = "Division operation";
            e2.ErrorText = "Dinominator must not be 0";
            e2.Reason = "Because the Denominator value is 0";
            FaultException<ErrorData> obj = new
                                FaultException<ErrorData>(e2);
            throw obj;
        }
    }
}
```

consume the service in console App'n :-

```
class program
{
    Static void Main()
    {
        try
        {
            ServiceReference1.ServiceClient obj = new ServiceClient();
            C.W.L("Enter a no");
            int a = int.parse(C.R.L());
            C.W.L("Enter b no");
            int b = int.parse(C.R.L());
            int result = obj.Division(a,b);
            C.W.L(result);
        }
        catch (FaultException<errordata> e1)
        {
            ErrorData er = e1.Detail;
            C.W.L("operation" + er.operation);
```

```
        c.w.L (" Reason" + ex.Reason);
        c.w.L (" ErrorMessage" + ex.ErrorText);
    }
    finally
    {
        c.R.L();
    }
}
}
```

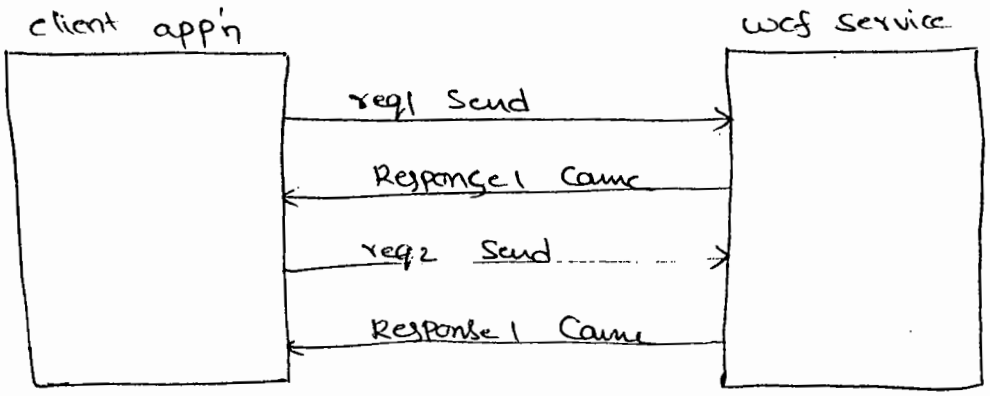
9/3/15

Message Exchange pattern in wcf:-

It is a mechanism to exchange the message blw client appn and wcf service.

Different exchange patterns are

1. Request - Reply communication
2. one - way communication (or) one way contract
3. Duplex communication (or) two way contract



In request - Reply communication client application will send req to server and wait for response

client can't send req until unless Response will came from service. i.e, in request - reply communication client must be in blocking mode or waiting mode until the response came from the Service.

Ex:- code for IService.cs

```
[Service Contract]
public interface IService1
{
    [Operation Contract]
    void DoSomething(int milliseconds);
    [Operation Contract]
    string Display(string name);
}
```

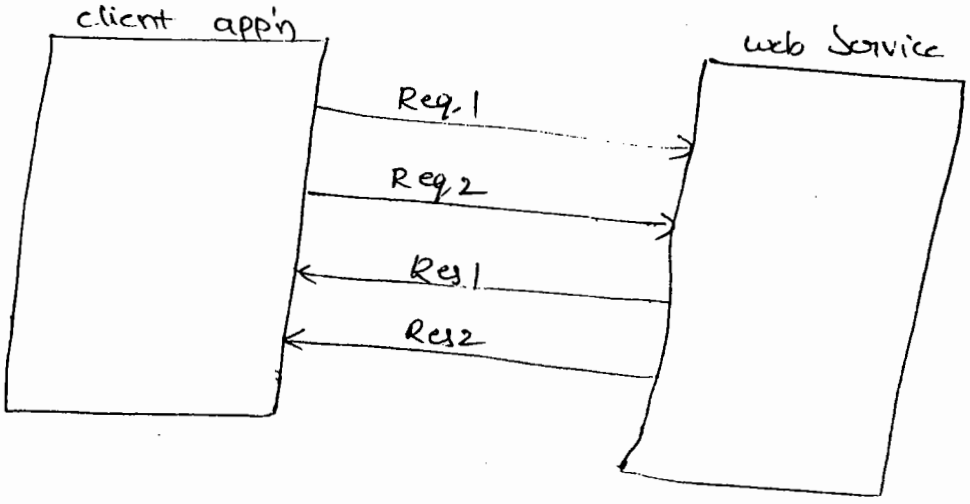
code for Service1.svc.cs

```
public class Service1 : IService1
{
    public void DoSomething(int milliseconds)
    {
        System.Threading.Thread.Sleep(milliseconds);
    }
    public string Display(string name)
    {
        System.Threading.Thread.Sleep(1000);
        return "welcome" + name;
    }
}
```

Consuming the Service in console App^{n}:-

```
using consoleapplication8.ServiceReference1;
class program
{
   Static void Main()
   {
     ServiceReference1.Service1Client obj = new Service1Client();
     C.W.L("Request1Sent" + System.Datetime.Now.ToString());
     obj.DoSomething(10000);
     C.W.L("Response1 came" + System.Datetime.Now.ToString());
     C.W.L("Request2 send" + System.Datetime.Now.ToString());
     String res2 = obj.Display("sathya");
     C.WL(res2+" is response2" + System.Datetime.Now.
                                 ToString());
     console.ReadLine();
   }
}
```

2. One-way Contract:-



In one way contract client appln will fire the request and forget about response i.e, we can apply one way contract only for the method have a return type as void.

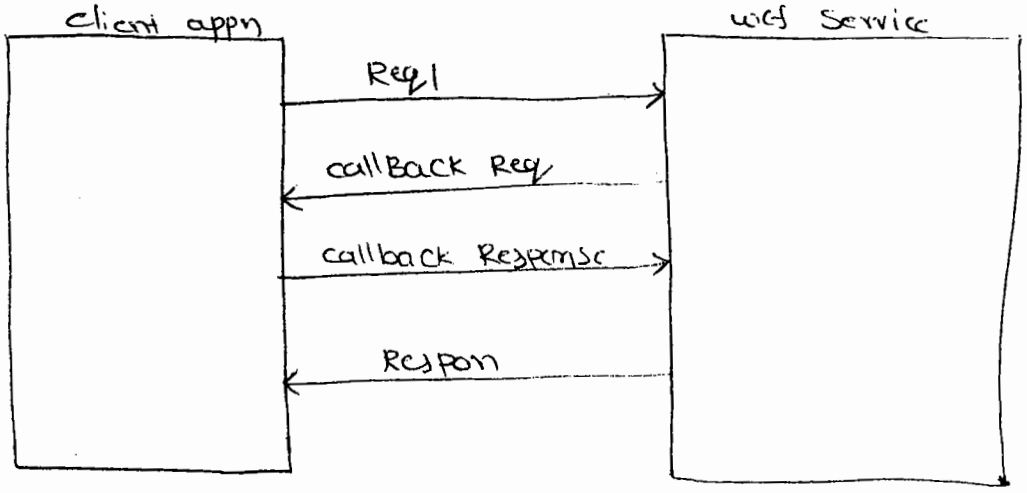
* we can't apply one way contract for the methods have a return type as datatype

Ex:-

```
[operation contract(IsOneWay = true)]
void doSomething(int milliseconds);
[operation contract]
String Display(String name);
```

10/3/15

Duplex contract (or) 2-way communication:-



* In duplex communication is the combination of request reply and one-way

* In duplex communication client appn will send request to service then service will send a callback request

to client and the callback method is implemented in client appn.

11/3/15

Security in WCF :-

In wcf we can Security in 3 ways

1. By using message level Security
2. Transport level Security
3. Mixed level Security.

Message level Security :-

(username and password)

In this model the user credentials are encripted in the messages using the web services security Specifications (ws-security).

*. Message level Security ensures contidentiality, intigrity and authentication at the Soap message

```
<wsHttpBinding>
  <binding name="wshttp">
    <Security mode="Message">
    <Message client credential type="username">
    </Security>
  </bindings>
</wsHttpBinding>
```

Different bindings that are supported

Bindingname	Transportmode	message mode
Basic HTTP binding	yes	yes
wsHTTP binding	yes	yes
wsDual http binding	No	yes
NetTcp binding	yes	yes
NetNamedpipebinding	yes	No
Net MSMQ binding	yes	yes
MSMq integration binding	yes	yes
ws Fedaration HTTP binding	No	yes

12/3/15

Step-1:- create the Service

```
using System.Net.security;
  [service contract]
public interface IService1
{
  [operation Contract (protectionLevel= protectionLevel.None)]
  string displayMessage withoutprotectionlevel ();
  [operation contract(protectionLevel= protectionLevel.sign)]
    string DisplayMessage signin();
    [operation contract(protectionLevel = protectionLevel.Encrypandsign)]
    string DisplayMessage withEncrypted sign();
}
```

code for Service1.svc.cs :-

implement the methods in Service class

Step 2 :- Host the Service in Console App'n

write the code for App.config See previous example

write Service host code in program.cs

Step-3 :- goto → tools → click on WCF ServiceConfiguration Editor File → open → config file → Select → App.config from the current app'n click on diagnostics → LogAuto

LogautoFlash → on

MessageLogging → on

Tracing → on

Select diagnostics → Message logging

LogEntiremessage = true → close the window → save the configuration in App.config file.

Step-4 :- Consume the Service in client App'n.

Drag & drop 3 buttons

click on button1

goto → App'n folder → double click on App_messages

+ Activities

+ click on last activity

+ Messages → check the Soap body the data will be stored in plain text because

protectionlevel = None.

* By default wsHttpbinding provides message level Security

* Message level Security automatically encripts & Signs the message to provide confidentiality and intigrity.

* Net TCP binding supports transport level Security even with transport level Security the message will be encrypted and signed.

* ws httpbinding also supports transport level Security

13/3/15

Transport level Security :-

It is used to provide Security for the message by using protocol.

* In order to enable transport level Security we have to convert http to https

* HTTPS will always communicate with Secure Socket layer

Syntax for applying transport level Security

```
<bindings>
 <wsHttpBinding>
  <binding name = "wshttp">
   <secure mode = "Transport">
    <transport clientCredentialType = "Basic">
    </transport>
   </Security>
  </bindings>
```

Q) How create SSC?

SSC is self signed certificate which is used to register with https

Q) How to add https under bindings?

start → run → inetmgr → Sclect default web site browser →

features view → bindings → Add

type = https

Ipaddress = systemIP

port = 443

14/3/15

Restfull Services :-

Q) what is HTTP?

→ http is communication. protocol which is used to access the resource via internet from server.

→ HTTP is a stateless protocol which does not maintain the state of the previous response.

Q). what is Soap?

→ Soap is simple object access protocol.

→ A Soap is a message transfer protocol which is used to transfer the message b/w client application and service

→ HTTP is communication protocol.

→ soap message can be transffeid via http protocol.

Q). what is REST?

→ REST is Representation state Transfer protocd.

→ REST is an architectural style which was build upon certain principles using the current web fundamentals.

→ Rest is used to access the resource that are available on the server via internet by using http protocd.

→ while working with Soap Services there is a heavy weight component called WSDL file, binding classes, to make the wcf service as light weight component Rayfield has invented a new architectural style with name rest to access the resource via internet by using http protocol.

→ In Restfull Service

1. Everything is treated as a resource

A resource may be an image, video, audio, file, db, etc

Ex:-

1. www.sathyatech.com/images/logo.gif → Image resource

2. www.satyatech.com/videos/demo.wmv
~~images/logo.gif~~ → video resource

→ Resource is any data that is accessing via internet

→ we can access the resource by using uri

URI - uniform Resource Identifier

→ generally in web appn we will access the resource by using url but in restfull Services by using uri

→ In restfull Services simple and uniform interfaces

→ Restfull Service can be consume by using 4 pre-defined methods that belongs to a 40 year old protocol i.e http protocol.

1. Get → This method is used to get the resource from the server.
2. put → It is used to create the resource
3. post → It is used to update resource
4. Delete → It is used to delete the resource.

These four methods are used to perform CRUD Operations

→ create (put)
→ Retreive (Get)
→ Update (post)
→ Delete (delete)

Method name	Http method	restful uri
Add customer	put	customer/Anil
create orders	put	orders/101
select customer	Get	customer/101
Delete customer	Delete	customer/101
update profik	post	customer/image/pic.jpg

→ In restfull Service the communication b/w client appn and services are called as representation.

→ In restfull Service each & every request and response is independent because they are dependent on http protocol as http is stateless protocol it will not maintain the state of previous response so the request and response are independent.

15/3/15

Example for restfull Service:-

code for IService.cs

```
[service contract]
public interface IService1
{
    [operation contract]
    [web get]
    String DisplayMessage(string name);
    [operation contract]
    [webget(UriTemplate="/DisplayMessage/{name}')]
    String DisplayMessageByUri(String name);
}
```

code for Service1.svc.cs

```
public class Service1 : IService1
{
    public String DisplayMessage(string name)
    {
        return "welcome" + name;
    }
```

```
public String DisplayMessageByUri(String name)
{
    return "welcome" + name;
}
}
```

goto → Service1.svc and write the code

```
<%@ ServiceHost Language="C#" Debug='true' Service=
"wcfService2.Service1' codebehind="Service1.svc.cs"
Factory="System.ServiceModel.Activation.WebServiceHost
Factory"%>
```

goto → web.config file and comment the below code

```
<ServiceHostingEnvironment multipleSiteBindingsEnabled=
"true"/>
```

press F5 and check the output

in the above example we can retrieve the service in 2 ways

1. By using methodname
2. By using uri

accessing the service by using method

http://localhost:1076/Service1.svc/DisplayMessage?name=nani

Accessing the service by using uri

http://localhost:1076/Service1.svc/DisplayMessage/nani

9) what is the difference blw Soap & rest ?

SOAP	REST
1. simple object Access protocol	1. Representational stat transfer protocol.
2. SOAP is XML based protocol	2. REST is not a protocol. It is an architectural style.
3. SOAP has a standard specifications	3. REST doesnot have any specification
4. Soap is developed based on SOA	4. REST is developed based on web style.
5. SOAP has specification for Statefull implementation	5. REST is stateless model.
6. SOAP is complex to implement	6. REST is simple to implement
7. SOAP supports different protocols like http, tcp/ip, MSMQ	7. REST supports only http and https
8. In Soap message format is XML inside Soap body	8. In REST message format is xml.
9. Soap supports WSDL	9. REST support WADL (web appn discription Language)
10. To Consume Soap service we require client appn	10. We can test the REST service.

FAQS

Q). what is wcf ?

wcf is windows communication fundation. which is used to develop interoparable applications

Q) what is difference blw wcf and webService?

wcf	webService
1. Service contract and operation contract attributes are used for defining wcf service	1. webservice and webmethod attributes are used for defining web service
2. supports various protocols like HTTP, HTTPS, TCP, named pipes and MSMQ	2. It supports like HTTP, HTTPS protocol.
3. Hosted in IIS, WAS (windows Activation service), self hosting, windows service	3. Host only IIS.
4. Supports security, relaible messaging, transaction and AJAX and REST supports	4. supports security but is less secure as compared to wcf
5. supports Data contract Serializer by using system. Runtime. Serialization	5. supports XML Serializer by using system. XML. Serialization.

6. Supports one-way, Request-Response and duplex service operations	6. Supports one-way and request-response service operations.
7. WCF are faster than web service.	7. Web services are slower than WCF.
8. Hash table can be serialized	8. Hash table cannot be serialize it can serializ only service.
9. unhandled exceptions does not return to the client as SOAP faults. WCF supports better exception handling by using fault contract.	9. unhandled exceptions return to the client as SOAP faults
10. Supports XML, MTOM, binary message encoding.	10. Supports XML and MTOM (Message transfer optimi-zation mechanism) message encoding.
11. Supports multi-threading by using service behaviour class	11. Doesnot support multi threading.

Q). What is Serialization?

Serialization is a process of converting object into stream of bytes.

Q) How many types of Serialization?

1. Binary Serialization
2. SOAP Serialization
3. XML Serialization.

Q). what is DeSerialization?

DeSerialization is a process of converting stream of bytes into object.

Q). what is end point?

→ End points are used to expose the Service to the client app'n

→ End points are ABC's of wcf service

A - Address: location of service

B - Binding: How to communicate with Service

C - contract: ~~with~~ it consists of the interface name

Q). what are the possible ways to host the wcf Service?

1. Self hosting
2. IIS hosting
3. WAS hosting.

Q). what are the message exchange patterns in wcf?

1. Request - Reply
2. one - way contract
3. Duplex contract.

Q). How to provide Security in wcf?

1. Message level
2. Transport level
3. Mixed level

Q). what are contract in wcf?

contract is an aggrement between 2 or more parties i.e, service and client appn

1. Service contract
2. Data contract
3. operation Contract.

Q). what is binding? How many bindings are there in wc

→ A binding defines how an endpoint communicates to the world

→ A binding defines the transport and encoding we us

→ Different types of binding are

1. Basic HTTP binding
2. WS HTTP binding
3. TCP binding
4. Fedaration ws binding
5. MSMQ binding.

Q) what is the address formats of the WCF transport schemas?

HTTP address format

TCP address format

MSMQ address format

Q). what is proxy?

→ The proxy is the local copy of the WCF Service.

→ we can create the proxy by using visual studio editor by simply add "Service reference".

→ we can create the proxy, by using "SvcUtil.exe".

Q). what is client and server in WCF?

in WCF client is web app'n, Server is WCF Service

Q). what is the diff b/w SOAP & REST

Q). What is the configuration tool to enable tracing and message level security or diagnostics

"WCF configuration editor".