

Chapter 9

Introduction to Windows

Communication Foundation

Mrs. Swati Satpute
Fergusson College

Objective

- What does it mean to be SOA?
- Contracts and Service Implementation
- Bindings and Behaviors
- Hosting the Service
- Consuming WCF Services

Remoting



Web services



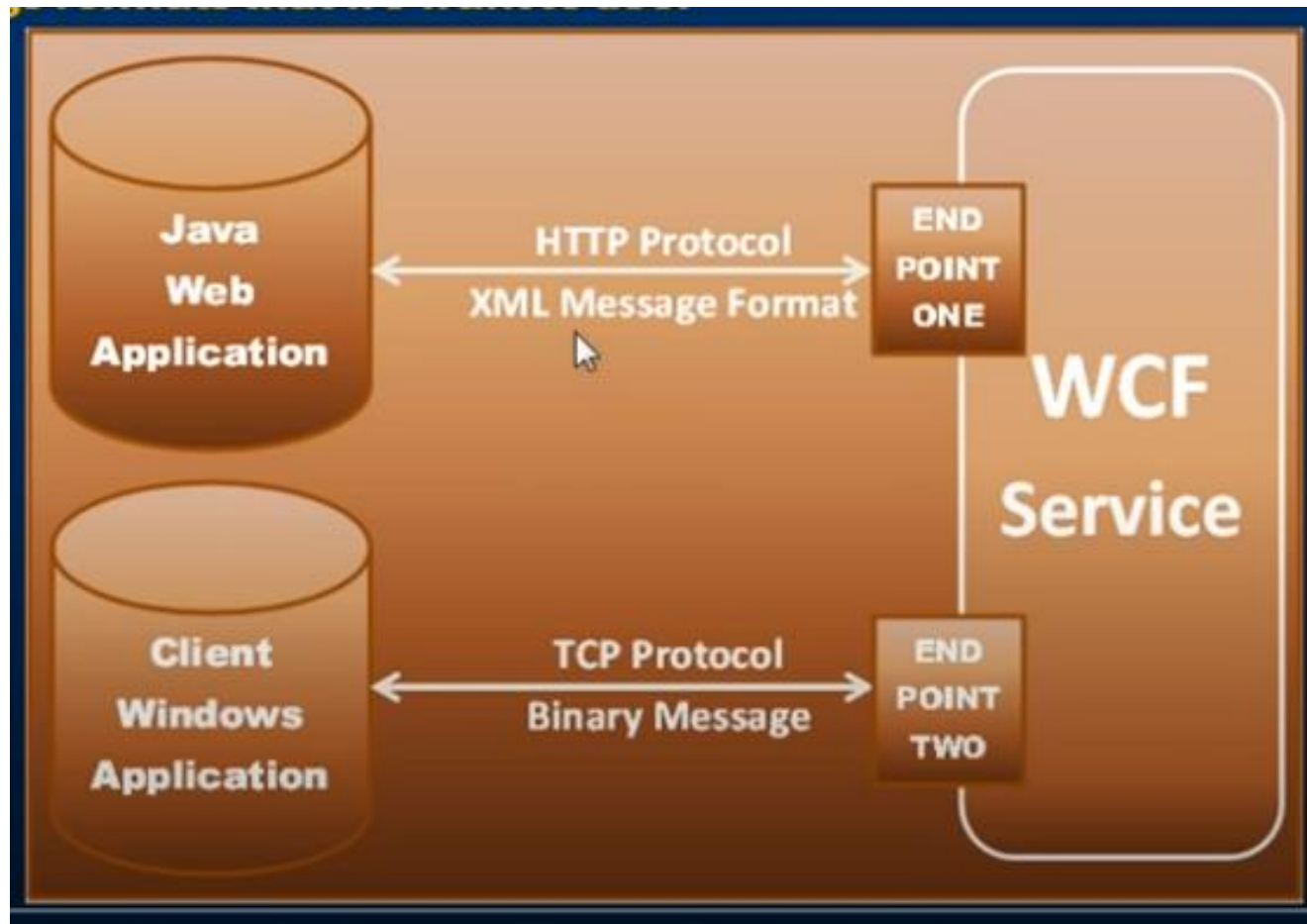
WCF

- It is a framework for **building, configuring, and deploying network-distributed and interoperable services.**
- It is used for building **Service Oriented applications.**

WCF

- **Distributed system** means system spread across multiple nodes to offer a solution
 - E.g. Three tier architecture system
 - Supports scalability as each layer has its own hardware.
- **Interoperability**
 - An application that can communicate with any other application that is built on any platform.
 - E.g. Website developed in ASP. Net can use yahoo finance services.

WCF



Service oriented architecture and principles

What is Service-Oriented Architecture ?

- **Service-Oriented Architecture (SOA)** is a set of principles and methodologies for designing and developing software in the form of **interoperable services**.
- These services are **well-defined business functionalities** that are built as software components (discrete pieces of code and/or data structures) that can be **reused for different purposes**.
- SOA design principles are used during the phases of systems development and integration.

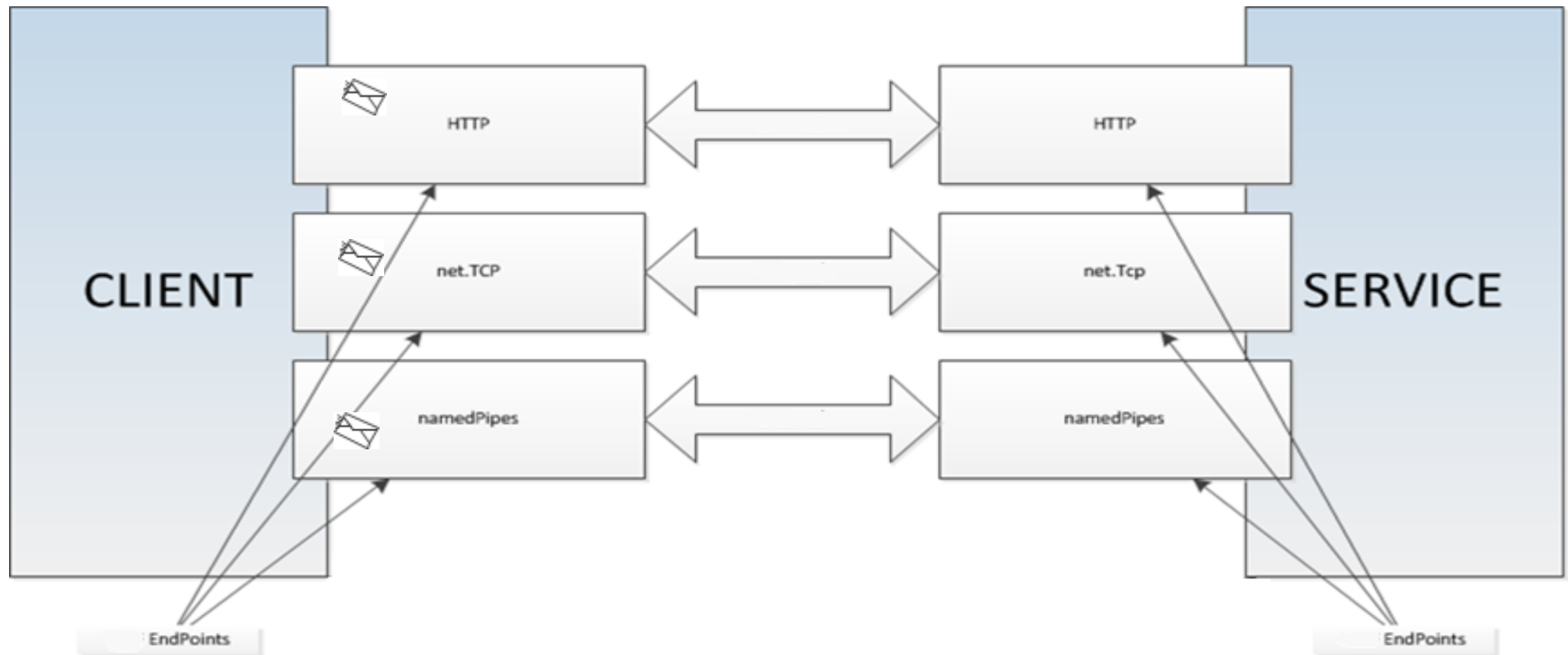
SOA Design principles

- Explicit Boundaries
- Share Contract and Schema, not Classes
- Autonomous
- Stateless
- Interoperable

What is WCF?

- Windows Communication Foundation (WCF) is a framework for **building service-oriented applications**.
- It is a runtime and a set of APIs for creating systems that **send messages between services and clients**.
- WCF makes it possible to have a **unified API irrespective of diverse transport mechanisms**.
- WCF is the foundation for other distributed technologies by Microsoft, such as Azure, AppFabric, and BizTalk

How it works: A WCF Overview



Application Components

- A WCF application consists of three components
 - WCF service,
 - WCF service host
 - WCF service client
- WCF platform is also known as the **Service Model**.

Fundamental Concepts of WCF

Message:

- It is a communication unit that comprises of several parts
- Messages are exchanged between Client and Server

Hosting

- WCF service hosting can be done through self-hosting, IIS hosting, and Windows Activation Service hosting.

Metadata

- It facilitates easy interaction between a client application and a WCF service.

Fundamental Concepts of WCF

Endpoint

- It defines the **address where a message is to be sent** or received.
- It also **specifies the communication mechanism** to describe how the messages will be sent along with defining the set of messages.
- Endpoint includes Address, Binding and Contracts

Structure of Endpoint

ABC's of WCF

- **Address**: Address specifies the **exact location** to receive the messages. It is expressed as scheme://domain[:port]/[path].
E.g. **net.tcp://localhost:9000/ServiceA**
- **Binding** : binding specifies **how to communicate with the endpoint**. It **states name of the transport protocol**(e.g. TCP or HTTP), **format of message encoding** (text or binary) and the **protocols** related to security (SSL or SOAP message security) as well as reliability.
- **Contracts** :It is a collection of operations that specifies what **functionality the endpoint exposes to the client**. It generally consists of an interface name.
 - What operations can be called by a client.
 - The form of the message.
 - The type of input parameters or data required to call the operation.
 - What type of processing or response message the client can expect

ABC's: Address, Binding, Contract

```
<!-- service endpoints -->  
<endpoint address="http://localhost:8000/CampGoldy/service"  
          binding="basicHttpBinding"  
          contract="CampGoldySvc.IReservation" />  
<endpoint address="net.tcp://localhost:8001/CampGoldy/service"  
          binding="netTcpBinding"  
          contract="CampGoldySvc.IReservation" />
```

Advantages of WCF

- It is interoperable. Can communicate with other services.
- It offer enhanced reliability as well as security compared to Web Services.
- Built in built-in logging mechanism .
- WCF has integrated AJAX and support for JSON (JavaScript object notation).
- It has a default security mechanism which is extremely robust.

WCF - Versus Web Service

- **Services** – WCF supports a **robust security**, **trustworthy messaging**, **transaction** and **interoperability**, while a web service only supports security services.
- **Serializer** – WCF Supports **DataContract** serializer by employing `System.Runtime.Serialization`, whereas a web service supports XML serializer by making use of `System.Xml.Serialization`.
- **Exception Handling** - In WCF, unhandled exceptions are handled by making use of **FaultContract**

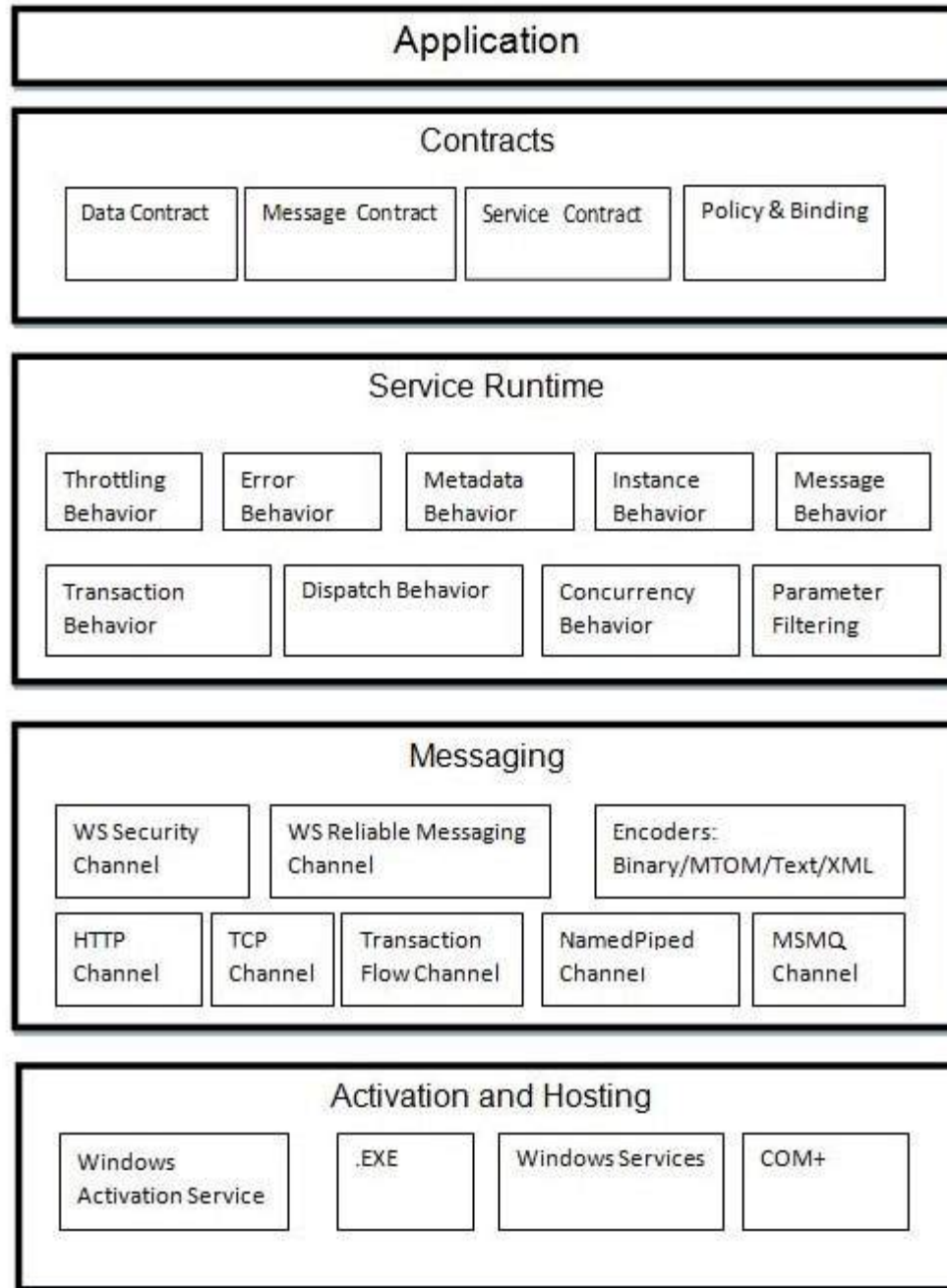
WCF - Versus Web Service

- **Bindings** – WCF supports several types of bindings like BasicHttpBinding, WSDualHttpBinding, WSHttpBinding, etc., while a web service supports only SOAP or XML.
- **Multithreading** – WCF supports multithreading by using the [ServiceBehavior](#) Class, whereas this is not supported in a web service.
- **Duplex Service Operations** – WCF supports duplex service operations apart from supporting one-way and request-response service operations, whereas a web service does not support duplex service operations.

WCF - Versus Web Service

- **Protocols** – WCF supports a range of protocols, i.e., **HTTP, Named Pipes, TCP, and MSMQ**, whereas a web service only supports HTTP protocol.
- **Hosting Mechanisms** – Various activation mechanisms are there for WCF hosting, i.e., **IIS** (Internet Information Service), **WAS** (Windows Activation Service), **Self-hosting and Windows Service**, but a web service is hosted only by IIS.

WCF - Architecture



WCF Architecture

Contracts

- Contracts determine what data and operations are exposed
- **Service contracts** define **operations**
- **Data contracts** define the **data**
- Other contracts
 - Message Contracts
 - Fault Contracts

WCF – Architecture - Contracts

Contract : Specifies the operation of a service and the kind of information it will make accessible.

- **Service contract:**

- It provides information to the client about the **offerings of the endpoint** and the **protocols** to be used in the communication process.

- **Data contract:**

- It specifies the **data exchanged by a service**.

- **Message contract**

- A data contract is controlled by a message contract. It primarily does the customization of the type formatting of the SOAP message parameters. WCF employs **SOAP format** for the purpose of communication.

- **Policy and Binding** – There are certain pre-conditions for communication with a service, and such conditions are defined by policy and binding contract. A client needs to follow this contract.

A Contract is just a Schema...

```
<tem:MakeReservation>
  <tem:reservation>
    <cam:Camper>
      <cam:CamperNotes>Happy Camper</cam:CamperNotes>
      <cam:FirstName>John</cam:FirstName>
      <cam:LastName>Doe</cam:LastName>
      <cam:Address1>501 2nd Ave</cam:Address1>
      <cam:Address2>Suite 123</cam:Address2>
      <cam:City>Minneapolis</cam:City>
      <cam:State>MN</cam:State>
      <cam:ZipCode>55401</cam:ZipCode>
      <cam:Email>John@Doe.com</cam:Email>
      <cam:Phone>6125551212</cam:Phone>
    </cam:Camper>
    <cam:Campsite>Goldy</cam:Campsite>
    <cam:ArrivalDate>2012-10-06</cam:ArrivalDate>
    <cam:DepartureDate>2012-10-07</cam:DepartureDate>
    <cam:ConfirmationCode></cam:ConfirmationCode>
    <cam:RegistrationNotes>Happy Camping!</cam:RegistrationNotes>
  </tem:reservation>
</tem:MakeReservation>
```

Service contracts & data contracts

```
[ServiceContract]
public interface IReservation
{
    [OperationContract]
    string MakeReservation(ReservationInfo reservation);

    [OperationContract]
    string ChangeReservation(ReservationInfo reservation);

    [OperationContract]
    string CancelReservation(string confirmationCode);

    [OperationContract]
    ReservationInfo GetReservation(string confirmationCode);

    [OperationContract]
    CampsiteInfoCollection GetCampsites();
}

[DataContract]
public class ReservationInfo
{
    [DataMember (Order = 0)]
    public CamperInfo Camper { get; set; }

    [DataMember (Order = 1)]
    public string Campsite { get; set; }

    [DataMember (Order = 2)]
    public DateTime ArrivalDate { get; set; }

    [DataMember (Order = 3)]
    public DateTime DepartureDate { get; set; }
}
```

WCF – Architecture - **Service Runtime**

Service Runtime:

- It specifies the various service behaviors that occur during runtime. Behaviors can undergo configuration
- **Throttling Behavior** – Manages the number of messages processed.
- **Transaction Behavior** – Enables a change in transaction state in case of any failure.
- **Concurrency Behavior** – Controls the functions that run parallel during a client-server communication.
- **Parameter Filtering** – Features the process of validation of parameters to a method before it gets invoked.

WCF – Architecture - Messaging

Messaging:

- This layer, **composed of several channels** and mainly **deals with the message content** to be communicated between two endpoints.
- Two major types of channels that comprise the channel stack are:
 - **Transport Channels:** responsible for sending and receiving messages using transport protocols like **HTTP, TCP, Peer-to-Peer, Named Pipes, and MSMQ**.
 - **Protocol Channels:** They implement wire-level protocols by modifying messages.

Hosting

WCF – Architecture - **Activation and Hosting**

Hosting is **the place where services are actually hosted** or can be executed for easy access by the client.

WCF Service can be hosted as any one of the following

- **IIS:** If hosted in IIS, service code gets activated automatically.
- **Windows Activation Service:** In WAS hosting Both HTTP and non-HTTP based communication is possible here by using TCP or Namedpipe protocol.

WCF – Architecture - **Activation and Hosting**

- **Self-hosting:** In this WCF service gets self-hosted as a console application.
- **Windows Service:** In this approach WCF service remain activated and accessible to the client due to no runtime activation.

Hosting the service

- A WCF Service is a library, it has **no life of its own**.
- The host brings the WCF service to life by providing the process in which it operates.
- Most of the time, WCF services will run in a ready-made host environment such as Internet Information Server (IIS) or Azure

Consuming a service

- Because WCF exposes service functionality through open standards, such as SOAP, you can use almost any type of client to consume the service.
- Allowing a .NET client to consume the service is as easy as creating a service reference or generating a proxy through a command-line utility (svcutil.exe)
- Non-.NET clients would typically use SOAP to consume a WCF Service

Create Service Demo

Demo

- Option1: WCF Service Library - template

Hosting in IIS

Step 1: Create a WCF service by selecting

- New → Web Site -> WCF Service

Step2 : Launch Inetmgr

- Run / Browse .svc page from IIS

Step3: Create client proxy by running svcutil from command prompt.

svcutil <http://localhost/IISHostedService/Service.svc>

- **Service.cs** – Proxy class for the WCF service
- **output.config** – Configuration information about the service

Step 4: Consumer :

- Create Console Based App
- Add Service.cs proxy file to project
- Copy “system.serviceModel” node from output.config to App.config
- Use client proxy and invoke WCF functionality

Hosting in IIS


- Step1: Create WCF Service Project
 - [Note - in the same folder where actual service is]
 - Remove 2 service related files from app_data folder
 - Make relevant changes in web.config
 - Rename **service.svc** file to meaningful name
 - Edit svc file to specify fully qualified Service name and remove code behind attribute

Svc file

- Svc file contain the ServiceHost directive specifying which service this file points to.
- The service code can reside in
 - The svc file
 - A separate assembly
 - A file in App_code folder
- The configuration for the wcf service goes in web.config file
- The ServiceHost directive is responsible for creating instance of ServiceHost whenever required.
- Note: There is no need to write code to instantiate and start ServiceHost






Hosting in IIS

Internet Information Services (IIS) Manager

← →  ▶ LAPTOP-FU5JM60V ▶ Sites ▶ Default Web Site ▶

File View Help

Connections

 |
▼  LAPTOP-FU5JM60V (LAPTOP-FU5JM60V)
 ▼  Application Pools
 ▼  Sites
 ▼  Default Web Site

**Physical folder path of
IIS host application**

Add Application

Site name: Default Web Site

Path: /

Alias:

HelloService

Application pool:

DefaultAppPool

Select...

Example: sales

Physical path:

/ork\DOT NET\Sample Projects\WCF\HelloService\IHost

...

Pass-through authentication

Connect as...

Test Settings...


☐ Enable Preload

OK

Cancel

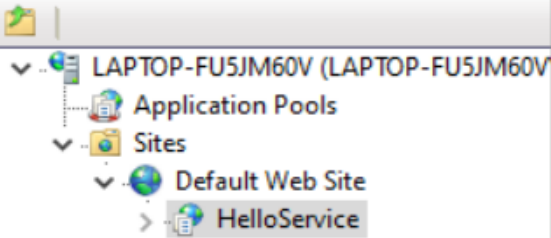
Hosting in IIS

- Compile app
- In iis, click on service application i.e. HelloService
- Go to content view
- Select svc file and select browse


← →  ▶ LAPTOP-FU5JM60V ▶ Sites ▶ Default Web Site ▶ HelloService ▶









File View Help

Connections



/HelloService Content

Filter:  Go  Show All | Group by: No Grouping ▼

Name	Type
 .vs	File Folder
 App_Code	File Folder
 App_Data	File Folder
 Bin	File Folder
 HelloService.svc	SVC File
 HelloServiceIISHost.sln	Microsoft Visual Studio Solution
 Web.config	CONFIG File
 Web.Debug.config	CONFIG File

<  >

 Features View  Content View

Ready

HelloService Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool.

```
svcutil.exe http://laptop-fu5jm60v/HelloService/HelloService.svc?wsdl
```

You can also access the service description as a single file:

```
http://laptop-fu5jm60v/HelloService/HelloService.svc?singleWsdl
```

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application.

C#

```
class Test
{
    static void Main()
    {
        HelloServiceClient client = new HelloServiceClient();

        // Use the 'client' variable to call operations on the service.

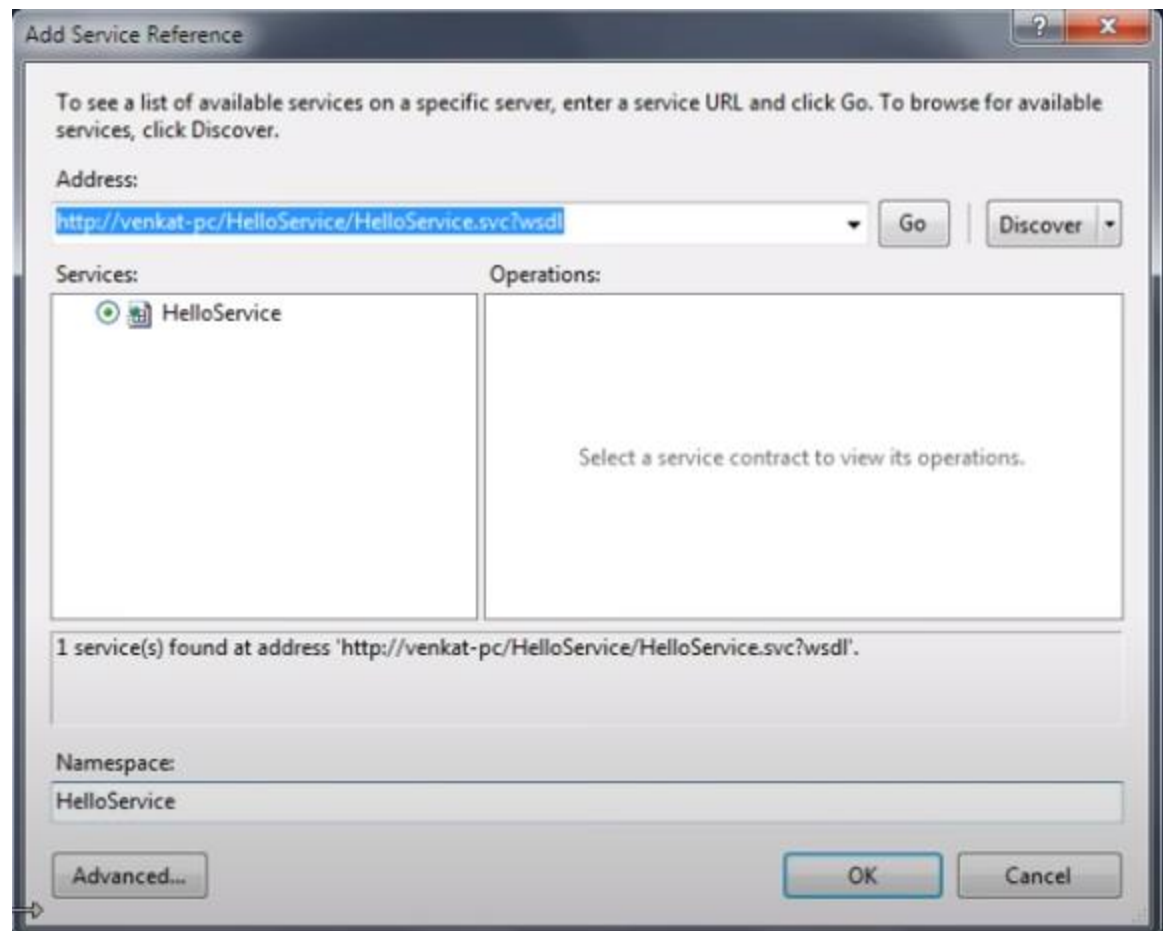
        // Always close the client.
        client.Close();
    }
}
```

Visual Basic

```
Class Test
```

Test Client for IIS hosted service

- One can create Client application to consume this service
- While adding IIS hosted service specify wsdl path while adding reference
- E.g. `http://laptop-fu5jm60v/HelloService/HelloService.svc?wsdl`



Other things you can do with wcf

- Secure Services
- RESTful Services
- Routing
- Streaming Data
- Discovery
- Web Sockets
- ...and much, much, more

Lets try

Demo

- Data Contract
- Data member
- Check product stock

The screenshot shows a Windows Form titled 'Form1'. It contains two text boxes: 'Product:' with the value '14' and 'In Stock' with the value '35'. To the right of the 'Product:' text box is a button labeled 'Get in Stock'. Below these elements is a section titled 'Product Details' which contains a text box displaying the following information:

```
Name: Tofu
UnitPrice: 23.2500
UnitsInStock: 35
UnitsOnOrder: 0
```

References

- <https://docs.microsoft.com/en-us/dotnet/framework/wcf/feature-details/endpoints-addresses-bindings-and-contracts>
- <https://www.c-sharpcorner.com/UploadFile/rkartikcsharp/abc-of-wcf/>
