Services

- web service-is a legacy technology
- newer- wcf is used(uses SOAP and returns data as XML,only supports HTTP protocol)
- newest - web api(HTTP based service and returns JSON or XML data by default)

web api-http based services whhich can be consumed by browsers,mobile applications,desktop applications,IOTs.

Web Service, WCF and now Web API are used to create distributed applications(service).

The .Net framework has a number of technologies that allow you to create HTTP services such as Web Service, WCF and now Web API.

Web Service

- It is based on SOAP and return data in XML form.
- It support only HTTP protocol.
- It is not open source but can be consumed by any client that understands xml.
- It can be hosted only on IIS.

WCF

- It is also based on SOAP and return data in XML form.
- It is the evolution of the web service(ASMX) and support various protocols like TCP, HTTP, HTTPS, Named Pipes, MSMQ.
- The main issue with WCF is, its tedious and extensive configuration.
- It is not open source but can be consumed by any client that understands xml.
- It can be hosted with in the applicaion or on IIS or using window service.

WCF Rest

- To use WCF as WCF Rest service you have to enable webHttpBindings.
- It support HTTP GET and POST verbs by [WebGet] and [WebInvoke] attributes respectively.
- To enable other HTTP verbs you have to do some configuration in IIS to accept request of that particular verb on .svc files
- Passing data through parameters using a WebGet needs configuration. The UriTemplate must be specified.
- It support XML, JSON and ATOM data format.
-

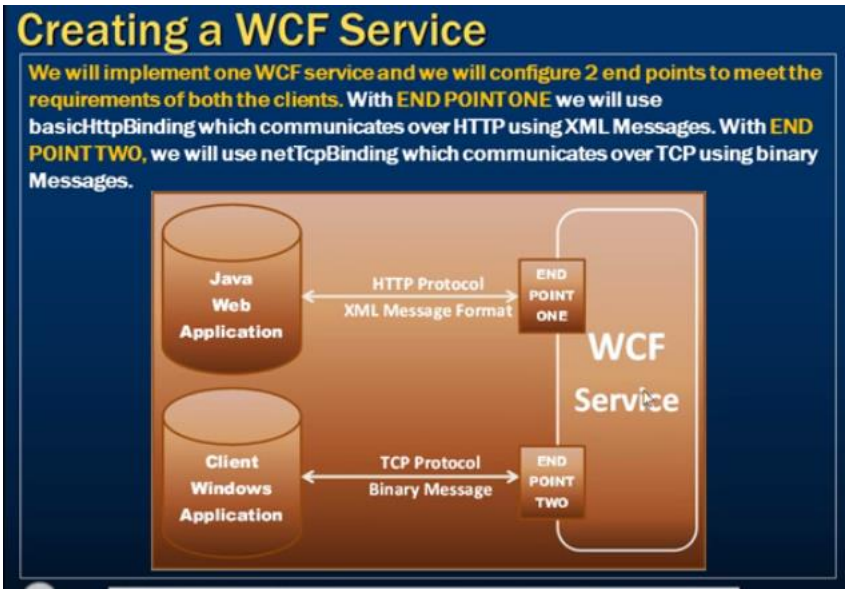
Web API

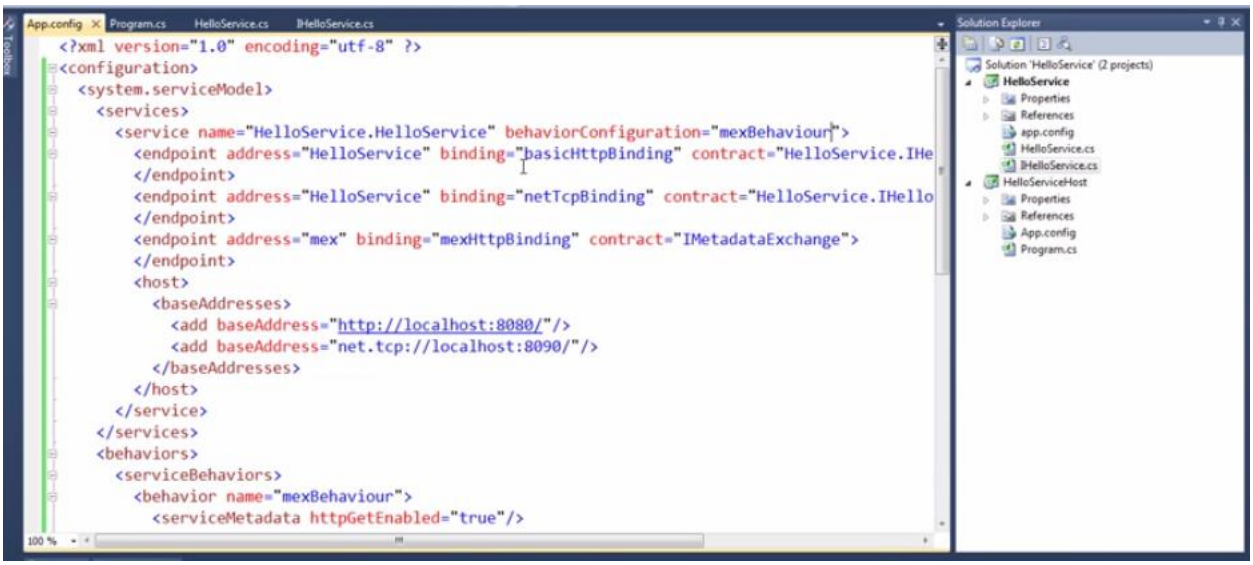
- This is the new framework for building HTTP services with easy and simple way.
- Web API is open source an ideal platform for building REST-ful services over the .NET Framework.

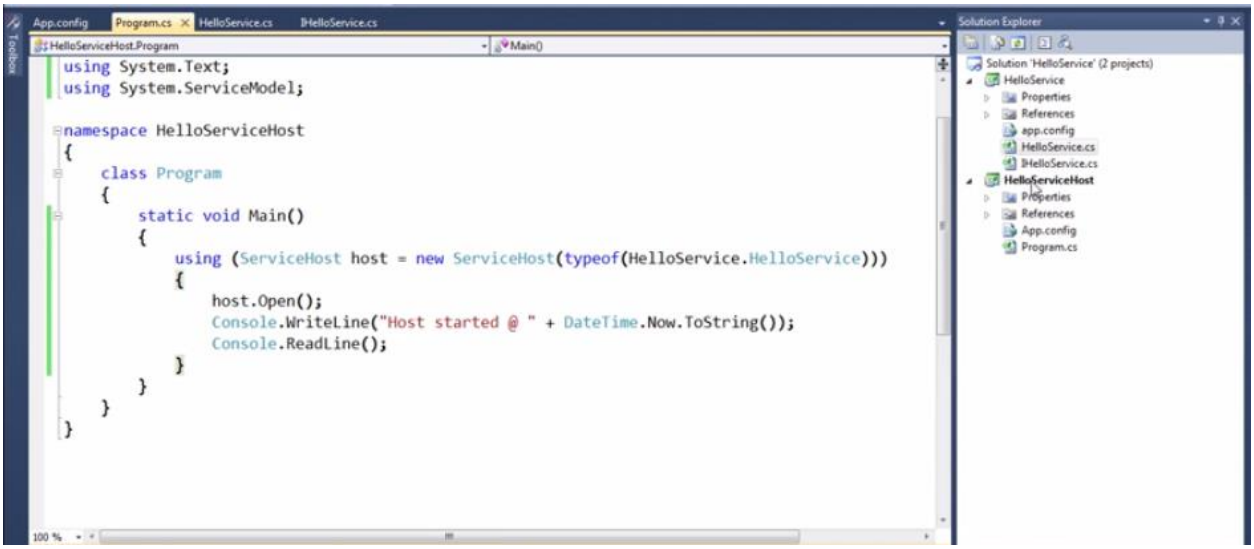
- Unlike WCF Rest service, it use the full feature of HTTP (like URIs, request/response headers, caching, versioning, various content formats)
- It also supports the MVC features such as routing, controllers, action results, filter, model binders, IOC container or dependency injection, unit

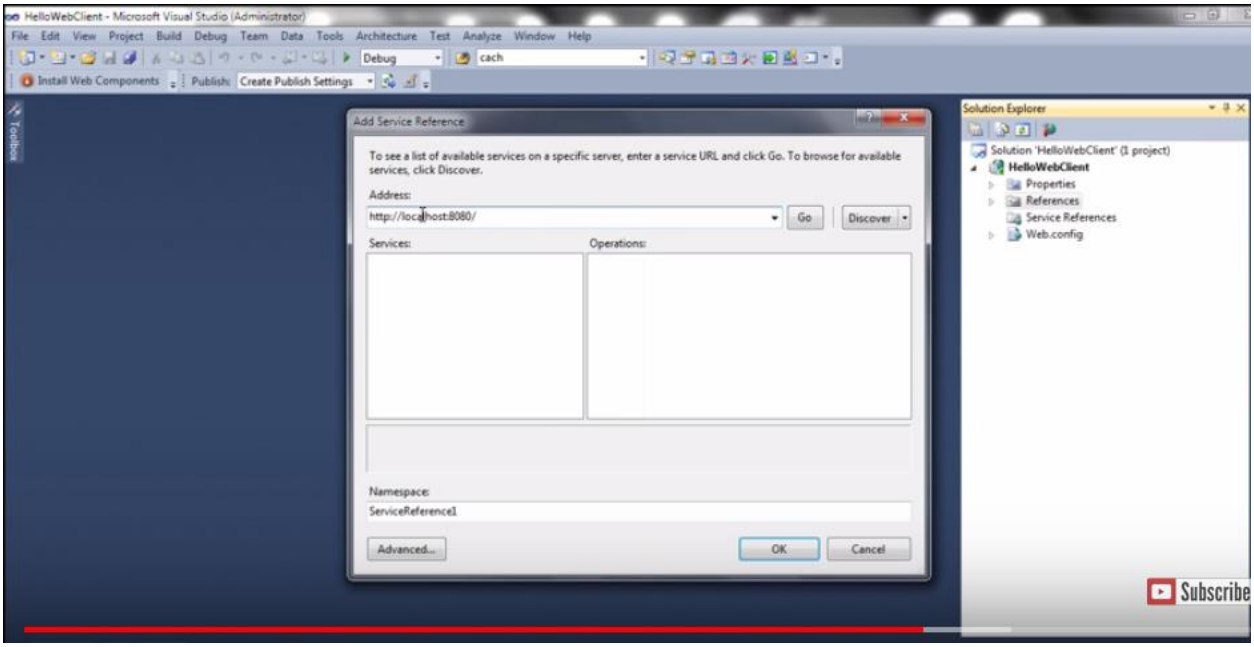
- testing that makes it more simple and robust.
- It can be hosted with in the application or on IIS.
- It is light weight architecture and good for devices which have limited bandwidth like smart phones.
- Responses are formatted by Web API's MediaTypeFormatter into JSON, XML or whatever format you want to add as a MediaTypeFormatter.

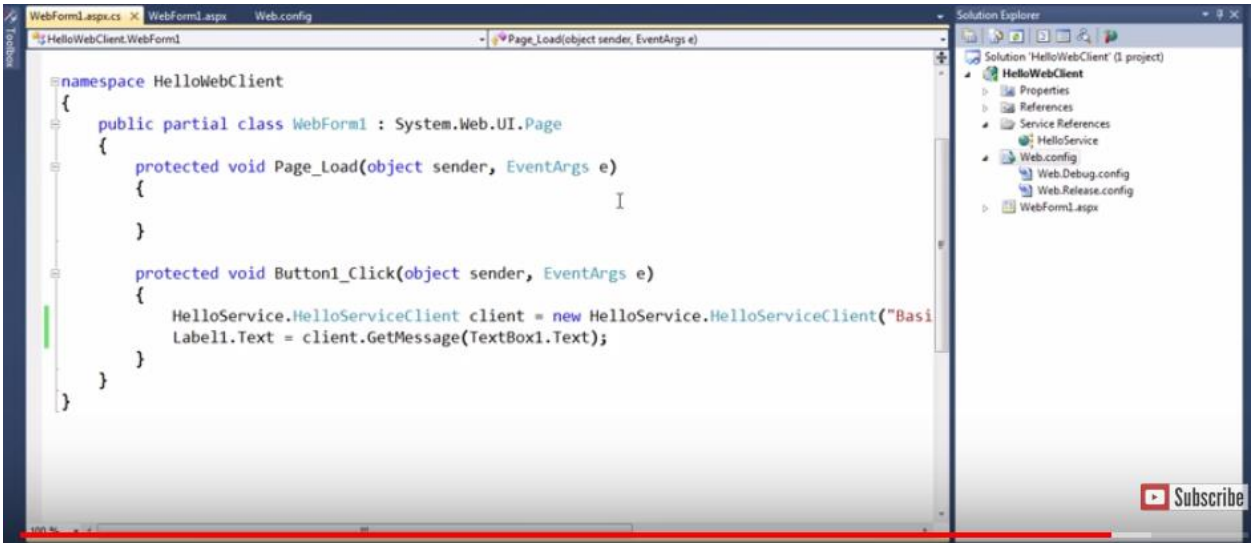
WCF






App.config
Program.cs
HelloService.cs
IHelloService.cs
HelloServiceHost.Program
Main()
using System.Text;
using System.ServiceModel;
namespace HelloServiceHost
{
class Program
{
static void Main()
{
using (ServiceHost host = new ServiceHost(typeof(HelloService.HelloService)))
{
host.Open();
Console.WriteLine("Host started @ " + DateTime.Now.ToString());
Console.ReadLine();
}
}
}
}
Solution Explorer
Solution 'HelloService' (2 projects)
HelloService
Properties
References
app.config
HelloService.cs
IHelloService.cs
HelloServiceHost
Properties
References
App.config
Program.cs


HelloWebClient - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
cach
Install Web Components
Publish:
Create Publish Settings
Add Service Reference
To see a list of available services on a specific server, enter a service URL and click Go. To browse for available services, click Discover.
Address:
http://localhost:8080/
Go
Discover
Services:
Operations:
Namespace:
ServiceReference1
Advanced...
OK
Cancel
Solution Explorer
Solution 'HelloWebClient' (1 project)
HelloWebClient
Properties
References
Service References
Web.config
Subscribe


WebForm1.aspx.cs
WebForm1.aspx
Web.config
HelloWebClient.WebForm1
Page_Load(object sender, EventArgs e)
namespace HelloWebClient
{
public partial class WebForm1 : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
}
protected void Button1_Click(object sender, EventArgs e)
{
HelloService.HelloServiceClient client = new HelloService.HelloServiceClient("Basi
Label1.Text = client.GetMessage(TextBox1.Text);
}
}
}
Solution Explorer
Solution 'HelloWebClient' (1 project)
HelloWebClient
Properties
References
Service References
HelloService
Web.config
Web.Debug.config
Web.Release.config
WebForm1.aspx
Subscribe