Steps to Create & Host a WCF Service (Using WCF Library)

Step 1: Create a Service

namespace WCFServiceLibraray

{

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface name "IMaths" in both code and config file together.

[ServiceContract]

public interface IMaths

{

[OperationContract]

int add(int x, int y);

[OperationContract]

int subt(int x, int y);

}

}

Step 2: Maths.cs

namespace WCFServiceLibraray

{

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name "Maths" in both code and config file together.

public class Maths : IMaths

{

public int add(int x, int y)

{

return x + y;

}

public int subt(int x, int y)

{

return x - y;

}

}

}

Step 3: app.config file

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

<configuration>

<appSettings>

<add key="aspnet:UseTaskFriendlySynchronizationContext" value="true" />

</appSettings>

<system.web>

<compilation debug="true" />

</system.web>

<!-- When deploying the service library project, the content of the config file must be added to the host's

app.config file. System.Configuration does not support config files for libraries. -->

<system.serviceModel>

<services>

<!--<service name="WCFServiceLibraray.Maths">

<endpoint address="" binding="basicHttpBinding" contract="WCFServiceLibraray.IMaths">

<identity>

<dns value="localhost" />

</identity>

</endpoint>

<endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange" />

<host>

<baseAddresses>

<add baseAddress="http://localhost:8733/MathsService/" />

</baseAddresses>

</host>

</service>-->

<service name="WCFServiceLibraray.Maths">

<endpoint address="" binding="basicHttpBinding" contract="WCFServiceLibraray.IMaths">

<identity>

<dns value="localhost" />

</identity>

</endpoint>

<endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange" />

<host>

<baseAddresses>

<add baseAddress="http://localhost:8733/MathsService/" />

</baseAddresses>

</host>

</service>

</services>

<behaviors>

<serviceBehaviors>

<behavior>

<!-- To avoid disclosing metadata information,

set the values below to false before deployment -->

<serviceMetadata httpGetEnabled="True" httpsGetEnabled="True"/>

<!-- To receive exception details in faults for debugging purposes,

set the value below to true. Set to false before deployment

to avoid disclosing exception information -->

<serviceDebug includeExceptionDetailInFaults="False" />

</behavior>

</serviceBehaviors>

</behaviors>

</system.serviceModel>

</configuration>

Step 4: Build It

**NOW WE HAVE TO HOST IT , WE ARE USING CONSOLE APPLICATION**

Step 1:

Create a Console Project

Add Reference of the Assembly of the service

Step 2:

Add config file, Entries should be same

Step 3:

using System;

using System.Collections.Generic;

using System.Linq;

using System.ServiceModel;

using System.Text;

using System.Threading.Tasks;

namespace MathsHostService

{

class Program

{

static void Main(string[] args)

{

ServiceHost host = new ServiceHost(typeof(WCFServiceLibraray.Maths));

host.Open();

Console.WriteLine("Service started...");

Console.Read();

Console.WriteLine("Service Stopped");

Console.Read();

}

}

}

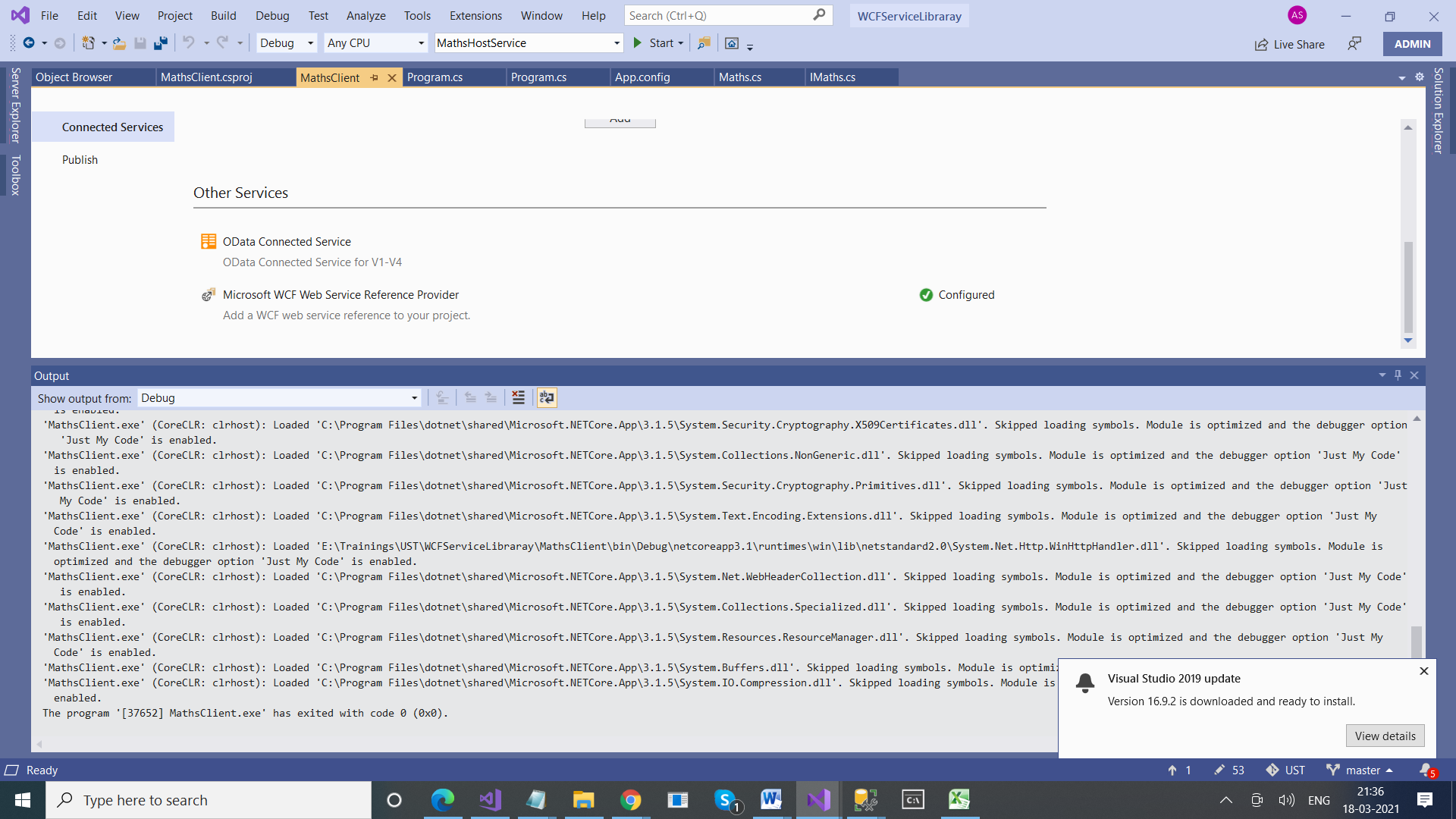
Step 4: Run this project

Service is hosted & running, Now we can consume it

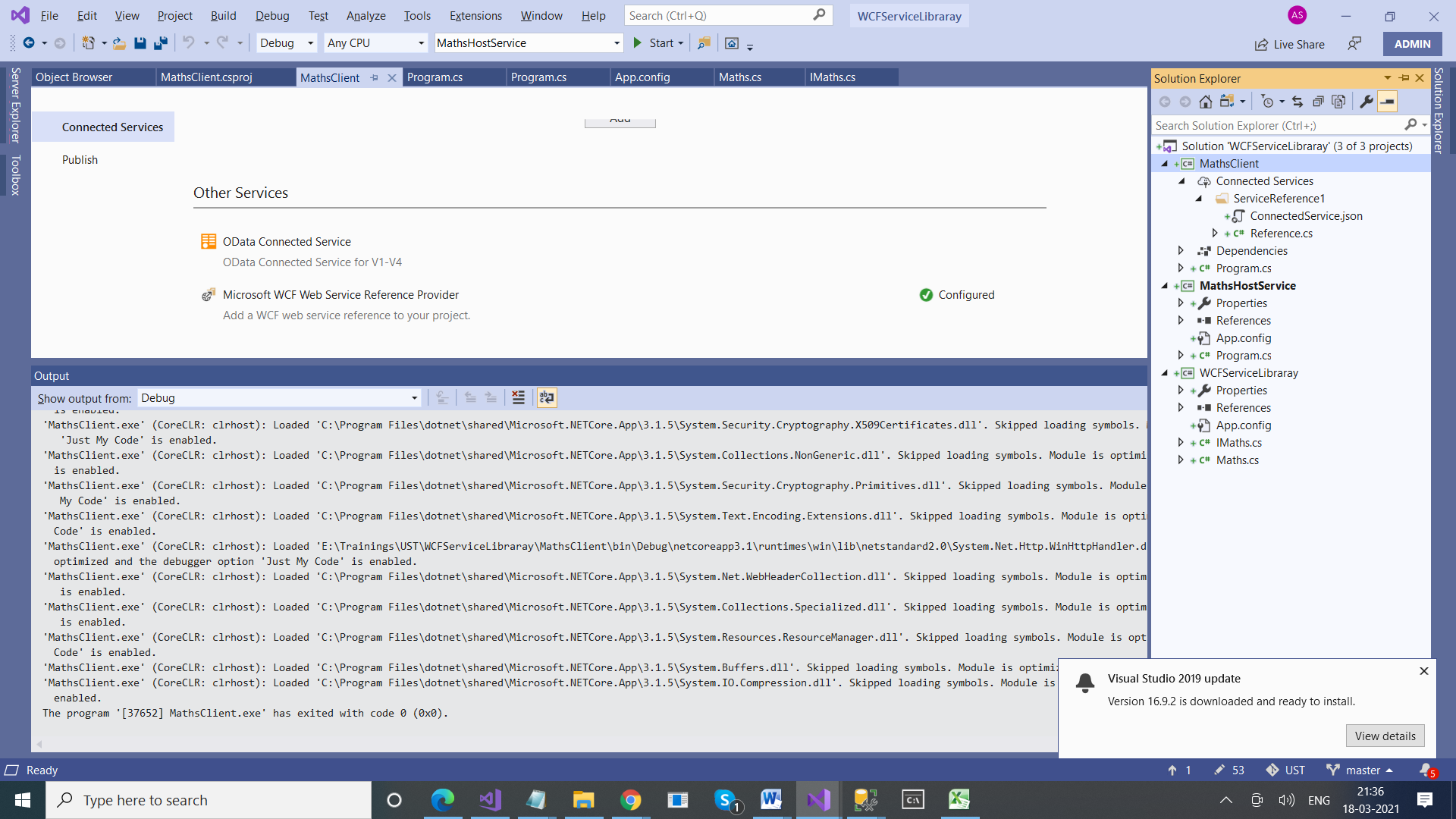
**NOW WE WILL CONSUME IT**

Step 1: Create Console Application (.Net Core)

Step 2: Add Connected Service



Step 3:



Step 4:

using System;

namespace MathsClient

{

class Program

{

static void Main(string[] args)

{

ServiceReference1.MathsClient mathsClient = new ServiceReference1.MathsClient();

int res = mathsClient.add(20, 30);

Console.WriteLine(res);

}

}

}