MyPhotos

Antet:

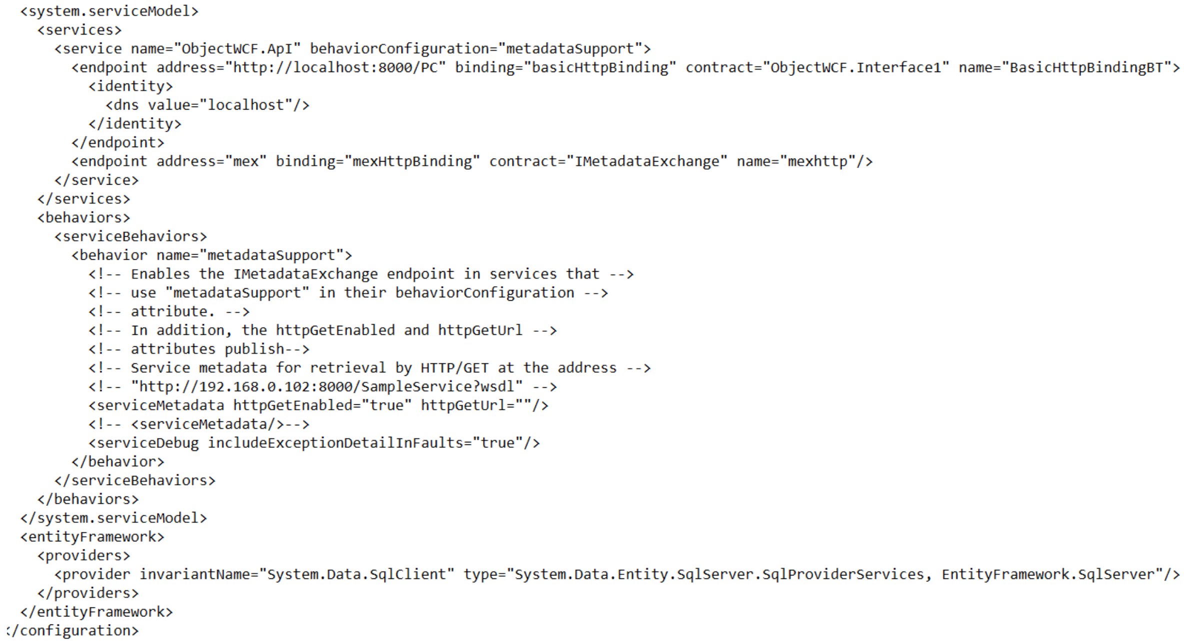
Subsemnatul Trufanda Bogdan Gheorghe, declar pe propria raspundere ca acest cod nu a fost copiat din Internet sau din alte surse. Pentru documentare am folosit urmatoarele surse:

https://docs.microsoft.com/en-us/dotnet/framework/wcf/how-to-host-a-wcf-service-in-a-managed-application

https://profs.info.uaic.ro/~iasimin/Laborator%20C%20S%20H/Laborator%20WCF%202020.pdf

Host pentru serviciu:

Adaugam o referinta la ObjectWCF, ServiceModel si ServiceModel.Description. Apoi adaugam in App.config urmatorul cod.



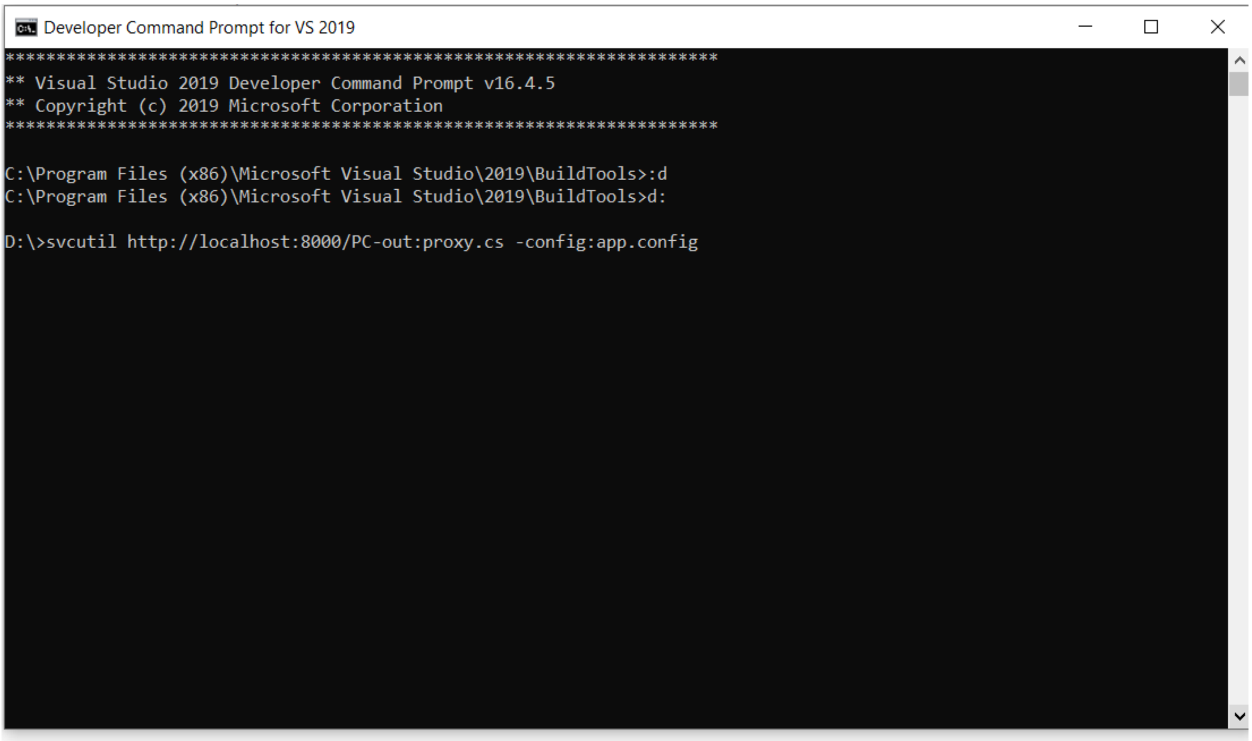
```
  <system.serviceModel>
    <services>
      <service name="ObjectWCF.ApI" behaviorConfiguration="metadataSupport">
        <endpoint address="http://localhost:8000/PC" binding="basicHttpBinding" contract="ObjectWCF.Interface1" name="BasicHttpBindingBT">
          <identity>
            <dns value="localhost"/>
          </identity>
        </endpoint>
        <endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange" name="mexhttp"/>
      </service>
    </services>
    <behaviors>
      <serviceBehaviors>
        <behavior name="metadataSupport">
          <!-- Enables the IMetadataExchange endpoint in services that -->
          <!-- use "metadataSupport" in their behaviorConfiguration -->
          <!-- attribute. -->
          <!-- In addition, the httpGetEnabled and httpGetUrl -->
          <!-- attributes publish-->
          <!-- Service metadata for retrieval by HTTP/GET at the address -->
          <!-- "http://192.168.0.102:8000/SampleService?wsdl" -->
          <serviceMetadata httpGetEnabled="true" httpGetUrl=""/>
          <!-- <serviceMetadata/>-->
          <serviceDebug includeExceptionDetailInFaults="true"/>
        </behavior>
      </serviceBehaviors>
    </behaviors>
  </system.serviceModel>
  <entityFramework>
    <providers>
      <provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer"/>
    </providers>
  </entityFramework>
</configuration>
```

In program.cs vom introduce urmatorul cod pentru a afisa detaliile endpointurilor si pentru a porni serverul:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.ServiceModel.Description;
using ObjectWCF;
using System.ServiceModel;


namespace HostWCF
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Lansare server WCF...");
            ServiceHost host = new ServiceHost(typeof(ApI), new Uri("http://localhost:8000/PC"));
            foreach (ServiceEndpoint se in host.Description.Endpoints)
                Console.WriteLine("A (address): {0} \nB (binding): {1} \nC (Contract): {2}\n", se.Address, se.Binding.Name, se.Contract.Name);
            host.Open();
            Console.WriteLine("Server in executie. Se asteapta conexiuni...");
            Console.WriteLine("Apasati Enter pentru a opri serverul!");
            Console.ReadKey();
            host.Close();
        }
    }
}
```

Mai apoi vom extrage cu ajutorul programului svcutil.exe metadata necesara pentru a construi clientul. Vom deschide Developer Command prompt si vom lansa comanda exact ca in poza de mai jos.



```
**********************************************************************
** Visual Studio 2019 Developer Command Prompt v16.4.5
** Copyright (c) 2019 Microsoft Corporation
**********************************************************************

C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools>:d
C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools>d:

D:\>svcutil http://localhost:8000/PC-out:proxy.cs -config:app.config
```

Se vor genera 2 fisiere (proxy.cs si app.config) pe care le vom folosi in crearea clientului.