

Subsemnata Andra – Elena N. Paduraru declara pe propria raspundere ca acest cod nu a fost copiat din Internet sau din alte surse. Pentru documentare am folosit urmatoarele surse:

Link-uri:

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

<https://profs.info.uaic.ro/~iasimin/Laborator%20C%20S%20H/Laborator1-2016.pdf>

<https://stackoverflow.com/questions/8986975/c-sharp-return-different-types>

<https://www.c-sharpcorner.com/UploadFile/mahesh/create-a-text-file-in-C-Sharp/>

<https://stackoverflow.com/questions/34851800/export-data-from-sql-server-to-text-file-in-c-sharpsaving-to-a-specific-folder>

<https://www.aspsnippets.com/Articles/Export-data-from-SQL-Server-to-Text-file-in-C-andVBNet.aspx>

<https://stackoverflow.com/questions/6062192/there-is-already-an-open-datareader-associated-with-thiscommand-which-must-be-c>

<https://stackoverflow.com/questions/18475195/error-there-is-already-an-open-datareader-associatedwith-this-command-which-mu/18475525>

<https://docs.microsoft.com/enus/dotnet/api/system.data.sqlclient.sqlparametercollection.addwithvalue?view=netframework-4.8>

<https://stackoverflow.com/questions/1132422/open-a-folder-using-process-start>

<https://docs.microsoft.com/en-us/dotnet/framework/winforms/controls/how-to-bind-data-to-thewindows-forms-datagridview-control>

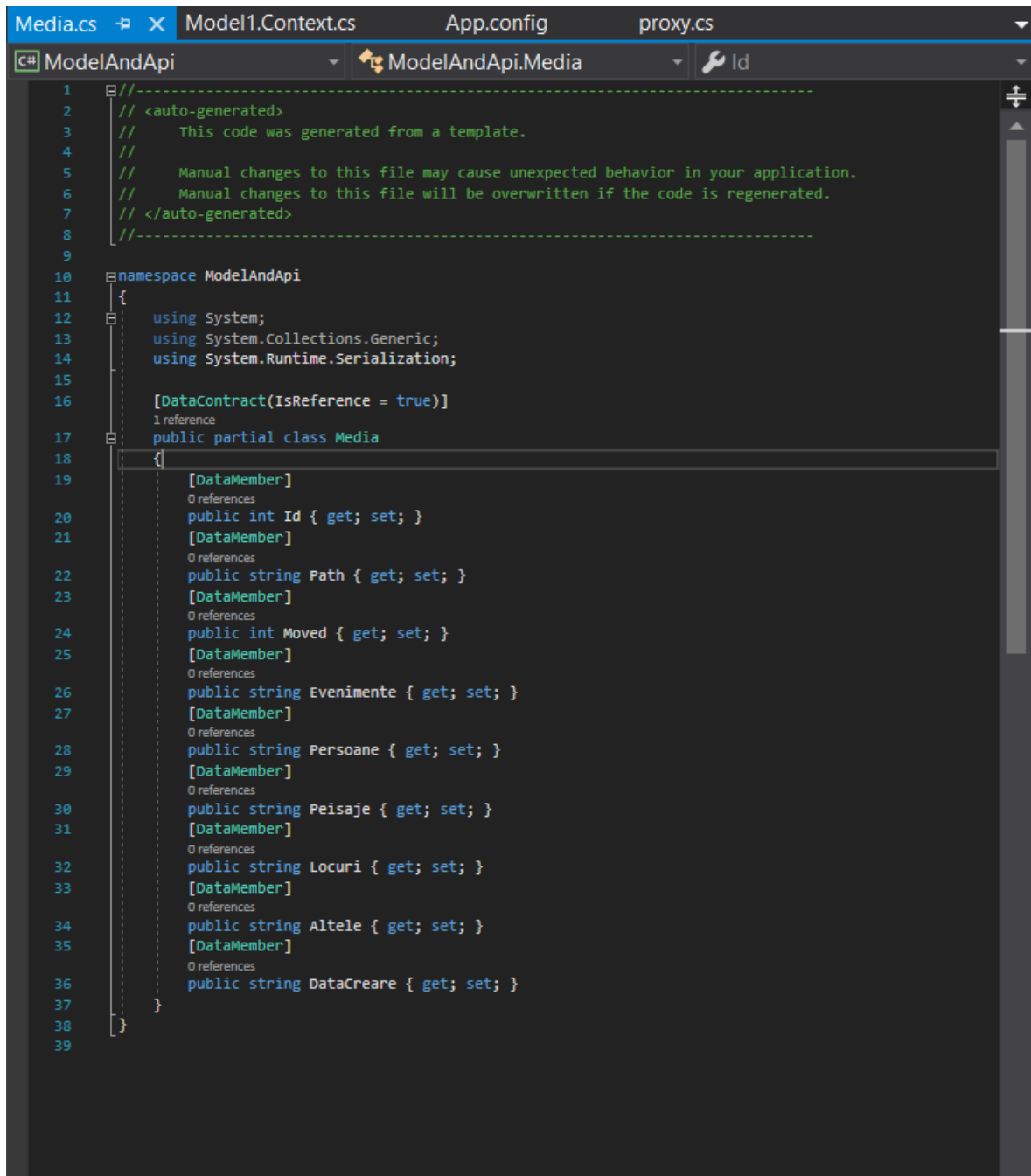
<https://stackoverflow.com/questions/27874566/getting-values-from-sql-reader-c-sharp>

<https://forums.asp.net/t/1794939.aspx?How+to+read+output+of+sql+command+C+>

Modificari fata de proiectul p1:

Model:

S-a modificat clasa generata din model (Model1.Context.cs) si s-au adaugat in constructor Configuration.LazyLoadingEnabled = false; si Configuration.ProxyCreationEnabled = false;. La fel s-a procedat si pentru clasa Media.cs din model unde s-au adaugat DataContract si DataMember.



The screenshot shows the Visual Studio IDE with the 'ModelAndApi' project selected. The 'Model1.Context.cs' file is open, displaying the 'Media' class. The class is a partial class within the 'ModelAndApi' namespace. It includes several using statements for System, System.Collections.Generic, and System.Runtime.Serialization. The class is decorated with [DataContract(IsReference = true)] and contains several public properties, each decorated with [DataMember]. The properties are: Id (int), Path (string), Moved (int), Evenimente (string), Persoane (string), Peisaje (string), Locuri (string), Altele (string), and DataCreare (string). Each property has a get and set accessor. The class is also decorated with [DataContract(IsReference = true)] and [DataMember] attributes. The code is as follows:

```
1  //-----  
2  // <auto-generated>  
3  //   This code was generated from a template.  
4  //  
5  //   Manual changes to this file may cause unexpected behavior in your application.  
6  //   Manual changes to this file will be overwritten if the code is regenerated.  
7  // </auto-generated>  
8  //-----  
9  
10 namespace ModelAndApi  
11 {  
12     using System;  
13     using System.Collections.Generic;  
14     using System.Runtime.Serialization;  
15  
16     [DataContract(IsReference = true)]  
17     public partial class Media  
18     {  
19         [DataMember]  
20         public int Id { get; set; }  
21         [DataMember]  
22         public string Path { get; set; }  
23         [DataMember]  
24         public int Moved { get; set; }  
25         [DataMember]  
26         public string Evenimente { get; set; }  
27         [DataMember]  
28         public string Persoane { get; set; }  
29         [DataMember]  
30         public string Peisaje { get; set; }  
31         [DataMember]  
32         public string Locuri { get; set; }  
33         [DataMember]  
34         public string Altele { get; set; }  
35         [DataMember]  
36         public string DataCreare { get; set; }  
37     }  
38 }  
39
```

```
Media.cs  Model1.Context.cs  App.config  proxy.cs
C# ModelAndApi  ModelAndApi.Model1Contain  OnModelCreating(DbModelB

1  //-----
2  // <auto-generated>
3  //   This code was generated from a template.
4  //
5  //   Manual changes to this file may cause unexpected behavior in your application.
6  //   Manual changes to this file will be overwritten if the code is regenerated.
7  // </auto-generated>
8  //-----
9
10 namespace ModelAndApi
11 {
12     using System;
13     using System.Data.Entity;
14     using System.Data.Entity.Infrastructure;
15
16     1 reference
17     public partial class Model1Container : DbContext
18     {
19         0 references
20         public Model1Container()
21             : base("name=Model1Container")
22         {
23             Configuration.LazyLoadingEnabled = false;
24             Configuration.ProxyCreationEnabled = false;
25
26         0 references
27         protected override void OnModelCreating(DbModelBuilder modelBuilder)
28         {
29             throw new UnintentionalCodeFirstException();
30
31         0 references
32         public virtual DbSet<Media> Media { get; set; }
33     }
34 }
```

API: Au fost adaugate clase noi. Codul a fost de asemenea modificat. Exemple:

```
public void EditData(string pathForEdit, string editEvent, string editPerson, string editPeisaj, string editLoc, string editAltele)
{
    using (var context = new ModellContainer())
    {
        IQueryable<Media> mediaList = from elem in context.Media where elem.Path.ToString() == pathForEdit select elem;
        foreach (var data in mediaList.ToList())
        {
            string test = data.ToString();

            if (!String.IsNullOrEmpty(test) && data.Path.ToString() == pathForEdit)
            {
                if (!String.IsNullOrEmpty(editEvent))
                {
                    string eventChange = data.Evenimente.ToString();
                    eventChange = eventChange + ", " + editEvent;
                    data.Evenimente = eventChange;
                }

                if (!String.IsNullOrEmpty(editPerson))
                {
                    string persChange = data.Persoane.ToString();
                    persChange = persChange + ", " + editPerson;
                    data.Persoane = persChange;
                }

                if (!String.IsNullOrEmpty(editPeisaj))
                {
                    string peisajChange = data.Peisaje.ToString();
                    peisajChange = peisajChange + ", " + editPeisaj;
                    data.Peisaje = peisajChange;
                }

                if (!String.IsNullOrEmpty(editLoc))
                {
                    string locChange = data.Locuri.ToString();
                    locChange = locChange + ", " + editLoc;
                    data.Locuri = locChange;
                }

                if (!String.IsNullOrEmpty(editAltele))
                {
                    string alteleChange = data.Altele.ToString();
                    alteleChange = alteleChange + ", " + editAltele;
                    data.Persoane = alteleChange;
                }
                context.SaveChanges();
            }
        }
    }
}
```

```

public string FindData(string result, string editEvent, string editPerson, string editPeisaj, string editLoc, string editAltele)
{
    using (var context = new Model1Container())
    {
        IQueryable<Media> medialist = from elem in context.Media select elem;
        foreach (var data in medialist.ToList())
        {
            string test = data.ToString();
            if (!String.IsNullOrEmpty(test))
            {
                if ((!String.IsNullOrEmpty(editEvent) && data.Evenimente.ToString().Contains(editEvent)) ||
                    (!String.IsNullOrEmpty(editPerson) && data.Persoane.ToString().Contains(editPerson)) ||
                    (!String.IsNullOrEmpty(editPeisaj) && data.Peisaje.ToString().Contains(editPeisaj)) ||
                    (!String.IsNullOrEmpty(editLoc) && data.Locuri.ToString().Contains(editLoc)) ||
                    (!String.IsNullOrEmpty(editAltele) && data.Altele.ToString().Contains(editAltele)))
                {
                    result = result + "ID: " + data.Id.ToString() + "\n" +
                        " Path: " + data.Path.ToString() + "\n" +
                        " Moved: " + data.Moved.ToString() + "\n" +
                        " Evenimente: " + data.Evenimente.ToString() + "\n" +
                        " Persoane: " + data.Persoane.ToString() + "\n" +
                        " Peisaje: " + data.Peisaje.ToString() + "\n" +
                        " Locuri: " + data.Locuri.ToString() + "\n" +
                        " Altele: " + data.Altele.ToString() + "\n" +
                        " Data Create: " + data.DataCreate.ToString() + "\t";
                }
            }
        }
    }
    return result;
}

```

```

public void SaveMedia(string path, string events, string persons, string peisaj, string locatie, string altele, DateTime creationDate)
{
    int check = 1;
    using (var context = new Model1Container())
    {
        foreach (var data in context.Media)
        {
            if (data.Path.ToString() == path)
            {
                check = 0;
            }
        }

        if (check == 1)
        {
            Media newMedia = new Media()
            {
                Path = path,
                Moved = 0,
                Evenimente = events,
                Persoane = persons,
                Peisaje = peisaj,
                Locuri = locatie,
                Altele = altele,
                DataCreate = creationDate.ToString()
            };
            context.Media.Add(newMedia);
            context.SaveChanges();
        }
    }
}

```

ObjectWCF:

S-a creat proiectul cu referinta la ModelAndApi. Au fost adaugate doua clase: Interfata IMedia (echivalentul la InterfaceWCF) si clasa care implementeaza metodele din interfata: ModelAndApi.cs

```
using System;
using ModelAndApi;

namespace ObjectWCF
{
    public class ModelAndApi : IMedia
    {
        void IMedia.SaveMedia(string path, string events, string persons,
            string peisaj, string locatie, string altele, DateTime creationDate)
        {
            API api = new API();
            api.SaveMedia(path, events, persons, peisaj, locatie, altele, creationDate);
        }

        void IMedia.DeleteMedia(string path)
        {
            API api = new API();
            api.DeleteMedia(path);
        }

        object IMedia.ShowGridData()
        {
            API api = new API();
            return api.ShowGridData();
        }

        object IMedia.ShowData()
        {
            API api = new API();
            return api.ShowData();
        }

        string IMedia.SaveFile()
        {
            API api = new API();
            return api.SaveFile();
        }

        void IMedia.EditData(string pathForEdit, string editEvent, string editPerson,
            string editPeisaj, string editLoc, string editAltele)
        {
            API api = new API();
            api.EditData(pathForEdit, editEvent, editPerson, editPeisaj, editLoc, editAltele);
        }

        string IMedia.FindData(string result, string editEvent, string editPerson,
            string editPeisaj, string editLoc, string editAltele)
        {
            API api = new API();
            return api.FindData(result, editEvent, editPerson, editPeisaj, editLoc, editAltele);
        }
    }
}
```

```

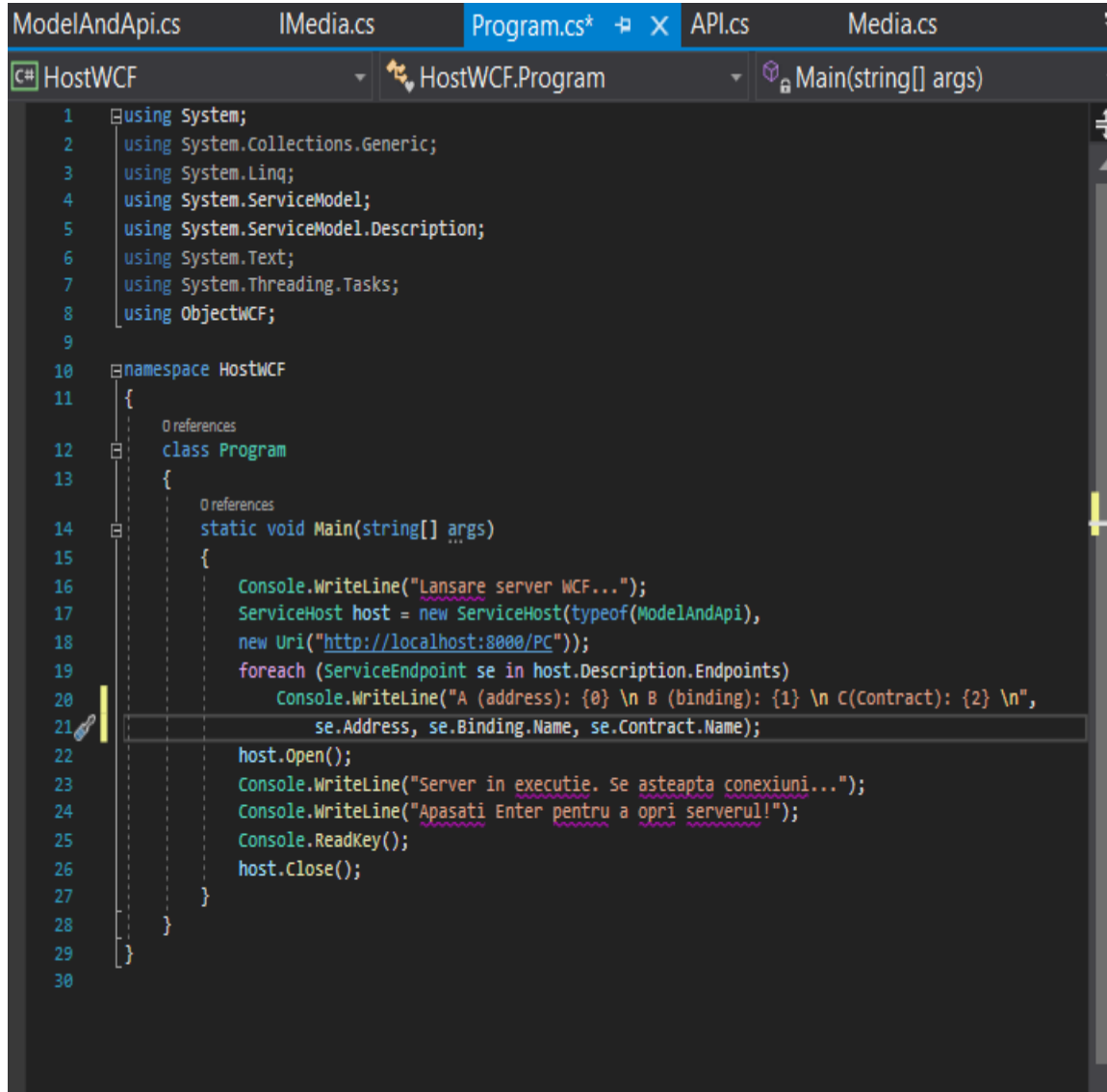
using System;
using System.ServiceModel;

namespace ObjectWCF
{
    [ServiceContract]
    interface IMedia
    {
        [OperationContract]
        void SaveMedia(string path, string events, string persons,
            string peisaj, string locatie, string altele, DateTime creationDate);
        [OperationContract]
        void DeleteMedia(string path);
        [OperationContract]
        object ShowGridData();
        [OperationContract]
        object ShowData();
        [OperationContract]
        string SaveFile();
        [OperationContract]
        void EditData(string pathForEdit, string editEvent,
            string editPerson, string editPeisaj, string editLoc, string editAltele);
        [OperationContract]
        string FindData(string result, string editEvent,
            string editPerson, string editPeisaj, string editLoc, string editAltele);
    }
}

```

Host:

S-a creat proiectul HostWCF cu referinta la ObjectWCF. S-a adaugat codul pentru completare in program.cs si s-a editat app.config:

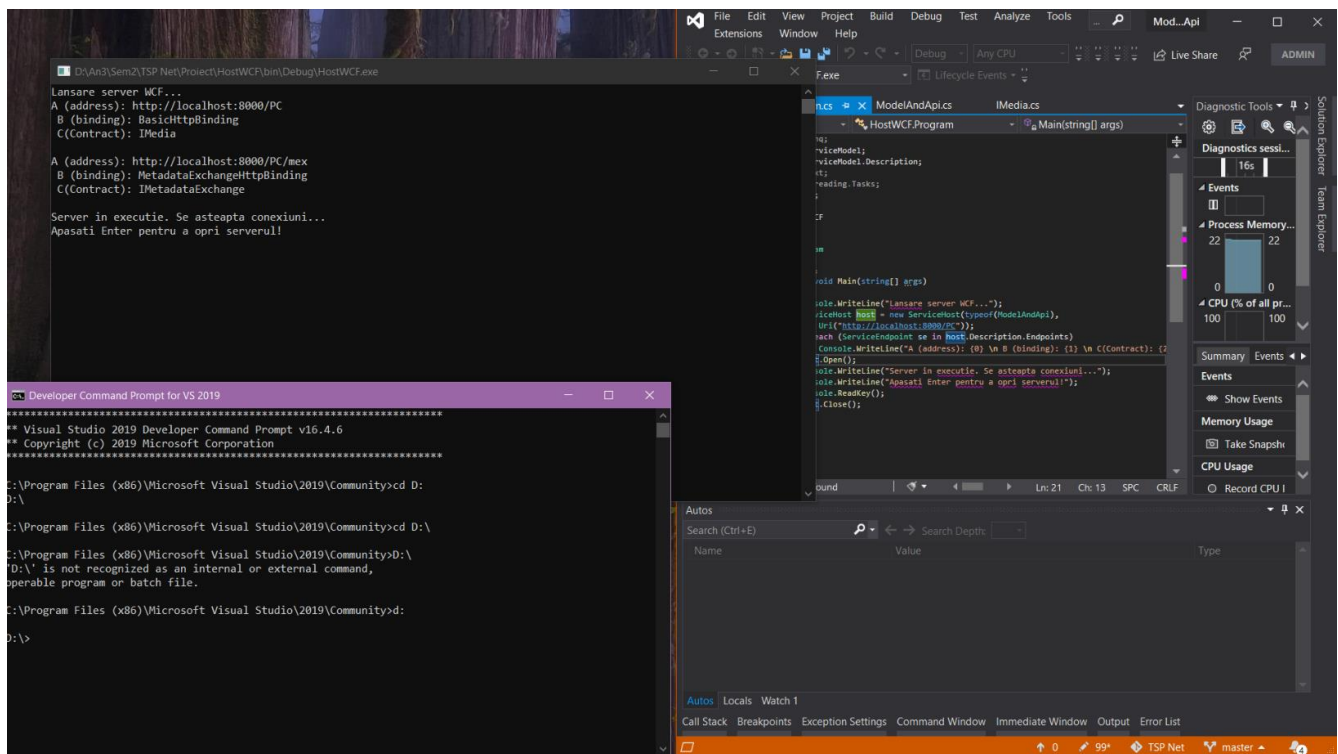


```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.ServiceModel;
5  using System.ServiceModel.Description;
6  using System.Text;
7  using System.Threading.Tasks;
8  using ObjectWCF;
9
10 namespace HostWCF
11 {
12     0 references
13     class Program
14     {
15         0 references
16         static void Main(string[] args)
17         {
18             Console.WriteLine("Lansare server WCF...");
19             ServiceHost host = new ServiceHost(typeof(ModelAndApi),
20             new Uri("http://localhost:8000/PC"));
21             foreach (ServiceEndpoint se in host.Description.Endpoints)
22                 Console.WriteLine("A (address): {0} \n B (binding): {1} \n C(Contract): {2} \n",
23                 se.Address, se.Binding.Name, se.Contract.Name);
24             host.Open();
25             Console.WriteLine("Server in executie. Se asteapta conexiuni...");
26             Console.WriteLine("Apasati Enter pentru a opri serverul!");
27             Console.ReadKey();
28             host.Close();
29         }
30     }
```



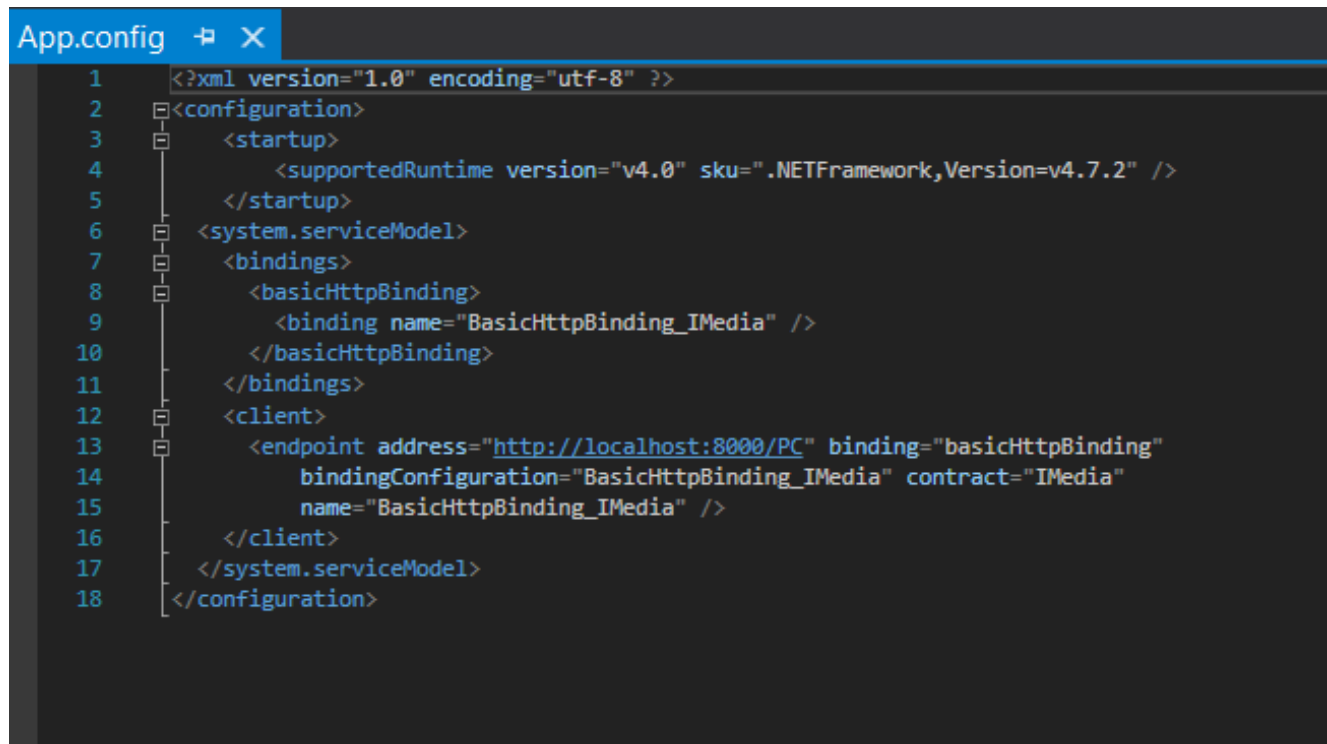
```
App.config*  ModelAndApi.cs  IMedia.cs  Program.cs*  API.cs
1  <?xml version="1.0" encoding="utf-8" ?>
2  <configuration>
3      <startup>
4          <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
5      </startup>
6
7      <connectionStrings>
8          <add name="ModelAndApiContainer"
9              connectionString="metadata=res://*/Model1.csdl|res://*/Model1.ssdl|res://*/Model1.msl;
10                 provider=System.Data.SqlClient;provider connection string=" data source=DESKTOP-A6HN40I;
11                 initial catalog=ProiectMedia;integrated security=True;MultipleActiveResultSets=True;
12                 App=EntityFramework";"
13                 providerName="System.Data.EntityClient"/>
14      </connectionStrings>
15
16      <system.serviceModel>
17          <services>
18              <service name="ObjectWCF.ModelAndApi" behaviorConfiguration="metadataSupport">
19
20                  <endpoint address="http://localhost:8000/PC" binding="basicHttpBinding"
21                      contract="ObjectWCF.IMedia" name="BasicHttpBinding_IMedia">
22                      <identity>
23                          <dns value="localhost"/>
24                      </identity>
25                  </endpoint>
26                  <endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange"
27                      name="mexhttp"/>
28              </service>
29          </services>
30          <behaviors>
31              <serviceBehaviors>
32                  <behavior name="metadataSupport">
33                      <!-- Enables the IMetadataExchange endpoint in services that -->
34                      <!-- use "metadataSupport" in their behaviorConfiguration -->
35                      <!-- attribute. -->
36                      <!-- In addition, the httpGetEnabled and httpGetUrl -->
37                      <!-- attributes publish-->
38                      <!-- Service metadata for retrieval by HTTP/GET at the address -->
39                      <!-- "http://192.168.0.102:8000/SampleService?wsdl" -->
40                      <serviceMetadata httpGetEnabled="true" httpGetUrl="" />
41                      <!-- <serviceMetadata/>-->
42                      <serviceDebug includeExceptionDetailInFaults="true"/>
43                  </behavior>
44              </serviceBehaviors>
45          </behaviors>
46      </system.serviceModel>
47  </configuration>
```

Rezultatul in urma rularii host-ului:



Client:

S-a folosit GUI de la proiectul 1. A fost creat app.config si proxy.cs care arata in felul urmatoar:



```
1  <?xml version="1.0" encoding="utf-8" ?>
2  <configuration>
3  <startup>
4      <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
5  </startup>
6  <system.serviceModel>
7      <bindings>
8          <basicHttpBinding>
9              <binding name="BasicHttpBinding_IMedia" />
10         </basicHttpBinding>
11     </bindings>
12     <client>
13         <endpoint address="http://localhost:8000/PC" binding="basicHttpBinding"
14             bindingConfiguration="BasicHttpBinding_IMedia" contract="IMedia"
15             name="BasicHttpBinding_IMedia" />
16     </client>
17 </system.serviceModel>
18 </configuration>
```

```
proxy.cs
Miscellaneous Files
IMedia

66 [System.Diagnostics.DebuggerStepThroughAttribute()]
67 [System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "4.0.0.0")]
68 public partial class MediaClient : System.ServiceModel.ClientBase<IMedia>, IMedia
69 {
70
71     public MediaClient()
72     {
73     }
74
75     public MediaClient(string endpointConfigurationName) :
76         base(endpointConfigurationName)
77     {
78     }
79
80     public MediaClient(string endpointConfigurationName, string remoteAddress) :
81         base(endpointConfigurationName, remoteAddress)
82     {
83     }
84
85     public MediaClient(string endpointConfigurationName, System.ServiceModel.EndpointAddress remoteAddress) :
86         base(endpointConfigurationName, remoteAddress)
87     {
88     }
89
90     public MediaClient(System.ServiceModel.Channels.Binding binding, System.ServiceModel.EndpointAddress remoteAddress) :
91         base(binding, remoteAddress)
92     {
93     }
94
95     public void SaveMedia(string path, string events, string persons, string peisaj, string locatie, string altele, System.DateTime creationDate)
96     {
97         base.Channel.SaveMedia(path, events, persons, peisaj, locatie, altele, creationDate);
98     }
99
100     public System.Threading.Tasks.Task SaveMediaAsync(string path, string events, string persons, string peisaj, string locatie, string altele, System.DateTime creationDate)
101     {
102         return base.Channel.SaveMediaAsync(path, events, persons, peisaj, locatie, altele, creationDate);
103     }
104
105     public void DeleteMedia(string path)
106     {
107         base.Channel.DeleteMedia(path);
108     }
109
110     public System.Threading.Tasks.Task DeleteMediaAsync(string path)
111     {
112         return base.Channel.DeleteMediaAsync(path);
113     }
114
115     public object ShowGridData()
116     {
117         return base.Channel.ShowGridData();
118     }
119
120     public System.Threading.Tasks.Task<object> ShowGridDataAsync()
121     {
122         return base.Channel.ShowGridDataAsync();
123     }
124 }
```

71 % No issues found

În urma modificării GUI prin adăugarea instanței clasei MediaClient codul arată în felul următor, exemplu:

```
namespace GUI
{
    public partial class Form1 : Form
    {
        MediaClient newMediaClient = new MediaClient();
        string imgLocation = "";
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void buttonAdd_Click(object sender, EventArgs e)
        {
            try
            {
                OpenFileDialog fileDialog = new OpenFileDialog();
                fileDialog.Filter = "JPG Files (*.jpg)|*.jpg|GIF Files (*.gif)|*.gif|PNG Files (*.png)|*.png|MP4 Files (*.mp4)|*.mp4|All Files (*.*)|*.*";
                fileDialog.Title = "Select Media";
                if (fileDialog.ShowDialog() == DialogResult.OK)
                {
                    imgLocation = Path.GetFullPath(fileDialog.FileName.ToString());
                    pictureBox.ImageLocation = imgLocation;
                }
            }
            catch (Exception except)
            {
                MessageBox.Show(except.Message);
            }
        }

        private void buttonSave_Click(object sender, EventArgs e)
        {
            byte[] img = null;
            FileStream fs = new FileStream(imgLocation, FileMode.Open, FileAccess.Read);
            BinaryReader br = new BinaryReader(fs);
            img = br.ReadBytes((int)fs.Length);
            newMediaClient.SaveMedia(imgLocation, textBoxEvent.Text, textBoxPerson.Text, textBoxPeisaj.Text, textBoxLoc.Text, textBoxAltele.Text, dateTimePicker1.Value.Date);
        }

        private void buttonDelete_Click(object sender, EventArgs e)
        {
            string pathToDelete = textBoxDeleteEditPath.Text;
            newMediaClient.DeleteMedia(pathToDelete);
        }
    }
}
```

```
private void buttonEdit_Click(object sender, EventArgs e)
{
    string pathForEdit = textBoxDeleteEditPath.Text;
    string editEvent = textBoxDeleteEditEvent.Text;
    string editPerson = textBoxDeleteEditPerson.Text;
    string editPeisaj = textBoxDeleteEditPeisaj.Text;
    string editLoc = textBoxDeleteEditLoc.Text;
    string editAltele = textBoxDeleteEditAltele.Text;

    newMediaClient.EditData(pathForEdit, editEvent, editPerson, editPeisaj, editLoc, editAltele);
}
```

Rezultatul in urma pornirii serverului si clientului in VS2019 ca administrator:

