Documentatie Client

Miron Robert-Andrei

April 14, 2020

1 Declaratie

Subsemnatul Miron Robert-Andrei, declar 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://docs.microsoft.com/en-us/dotnet/framework/`,
`https://profs.info.uaic.ro/~iasimin/Laborator%20C%20S%20H/Laborator%20WCF%202020.pdf`,
`https://docs.microsoft.com/en-us/dotnet/framework/wcf/feature-details/`,
`https://profs.info.uaic.ro/~iasimin/Special/Curs_WCF.pdf`

2 Introducere

Proiectul al doilea este unul de tip Class Library (EF .NET Framework) ce contine definitia si implementarea unui serviciu WCF, serviciu ce va apela metodele din API creat in cadrul proiectului 1.

In aceasta documentatie vom vorbi despre client.

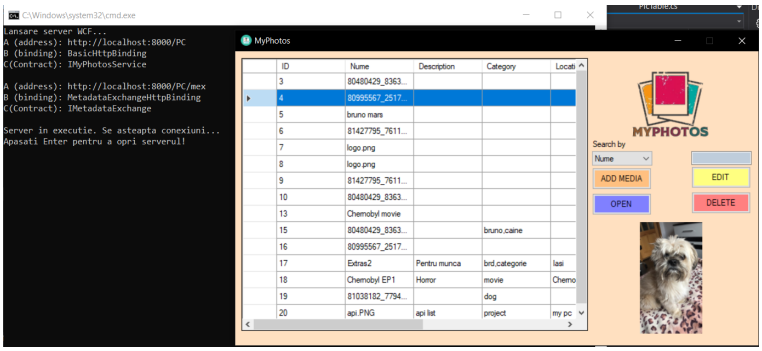

Figure 1: Host

3 Descriere etapa

Ajungand la aceasta etapa, proiectul nu mai utilizeaza direct API-ul din baza de date, ci se realizeaza comunicarea cu service-ul pus la dispozitie de catre host. Acesta, la randul lui, apeleaza baza de date.

```
MyPhotosServiceClient myPhotosServiceClient = new MyPhotosServiceClient();

var items = myPhotosServiceClient.GetAllPicsForLoad();
```

```
public MyPhotos.PicTable[] GetAllPicsForLoad()
{
    return base.Channel.GetAllPicsForLoad();
}
```

Figure 2: Exemplu apelare serviciu

S-a adaugat la proiect clasa proxy generata.

Unde a fost nevoie, s-a realizat conversie de la List¡string¿ la String[]. Proxy-ul a generat automat un array de string-uri, in loc de List¡string¿, din acest motiv fiind necesara conversia.

```
api.updateEntry(id, textBoxName.Text, textBox1.Text, textBox2.Text, categories.ToArray(), peoples.ToArr
```

Figure 3: Exemplu conversie

Proxy este clasa care ajuta la realizarea conexiunii intre interfata si host(interfata apeleaza functiile din proxy, iar proxy apeleaza functiile din host).

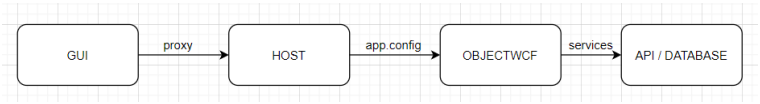

Figure 4: Diagrama

In fisierul de configuratie al proiectului curent s-a adaugat legatura cu interfata prin intermediul setarilor ¡system.serviceModel¿ pentru a putea accesa host-ul:

```
<system.serviceModel>
  <bindings>
    <basicHttpBinding>
      <binding name="BasicHttpBinding_IService" />
    </basicHttpBinding>
  </bindings>
  <client>
    <endpoint address="http://localhost:8000/PC" binding="basicHttpBinding"
        bindingConfiguration="BasicHttpBinding_IService" contract="IMyPhotosService"
        name="BasicHttpBinding_IService" />
  </client>
</system.serviceModel>
```

Figure 5: App.config

Am rulat, de asemenea, comanda svcutil http://localhost:8000/PC -out:proxy.cs –config:app.config, care a generat proxy.cs si app.config. Inainte de a rula comanda, am urmat pasii descrisi in documentul New Model, pentru a se putea realiza generarea corecta a acestora.

```
namespace MyPhotos
{
    using System.Runtime.Serialization;


    [System.Diagnostics.DebuggerStepThroughAttribute()]
    [System.CodeDom.Compiler.GeneratedCodeAttribute("System.Runtime.Serialization", "4.0.0.0")]
    [System.Runtime.Serialization.DataContractAttribute(Name="Categories", Namespace="http://schemas.datacor
    6 references
    public partial class Categories : object, System.Runtime.Serialization.IExtensibleDataObject
    {

        private System.Runtime.Serialization.ExtensionDataObject extensionDataField;

        private string CategoryNameField;

        private int ID_CategoryField;

        private MyPhotos.PicTable[] PicTableField;

        2 references
        public System.Runtime.Serialization.ExtensionDataObject ExtensionData
```

Figure 6: proxy.cs