

Documentatie ObjectWCF

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 Introdocere

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.

Documentul actual va prezenta partea de Object WCF.

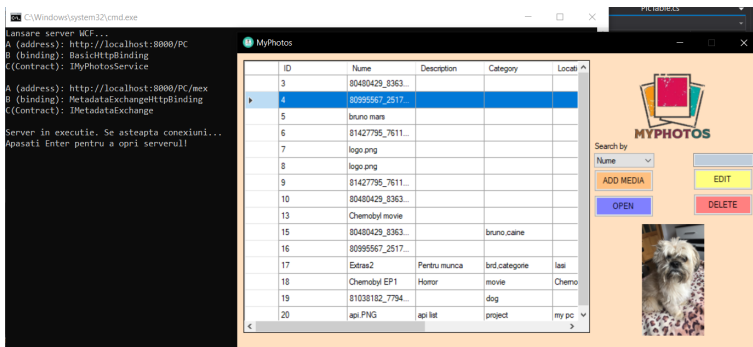


Figure 1: Host

3 Descriere etapa

In aceasta etapa s-a implementat pas cu pas serviciul WCF. Initial, am preluat toate functiile din API-ul construit pentru proiectul anterior. Definitiiile lor au fost, mai apoi, utilizate in interfetele aferente(figura 2 - ICategories, IPerson, IPicTable), interfete care, in final, s-au unit in una comuna(IMyPhotoInterface).

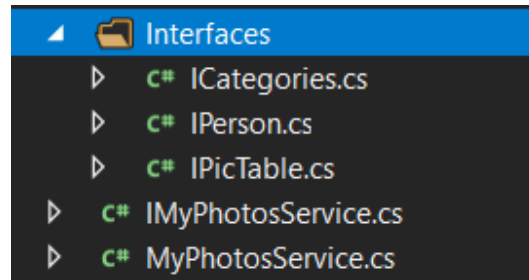


Figure 2: Interfete si Service

```
namespace MyPhotosProject2.Interfaces
{
    [ServiceContract]
    1 reference
    interface IPicTable
    {
        [OperationContract]
        1 reference
        void AddEntry(string name, string path, string description, string location, DateTime dt,
        [OperationContract]
        1 reference
        void updateEntry(int id, string title, string description, string location, List<string>
        [OperationContract]
        1 reference
        List<PicTable> GetAllPicsForLoad();
    }
}
```

Figure 3: ICategories, IPerson, IPicTable

```
namespace MyPhotosProject2
{
    [ServiceContract]
    1 reference
    interface IMyPhotosService : ICategories, IPerson, IPicTable
    {
    }
}
```

Figure 4: IMyPhotosService

Dupa cele mentionate anterior, am creat MyPhotosService, clasa in care am implementat functiile din interfetele create. In vederea realizarii acestui lucru, folosesc API-ul din baza de date deja cunoscuta.

```

namespace MyPhotosProject2
{
    2 references
    public class MyPhotosService : IMyPhotosService
    {
        private readonly API myAPI;

        0 references
        public MyPhotosService()
        {
            myAPI = new API();
        }

        1 reference
        public void AddEntry(string name, string path, string description, string location, DateTime dt, string categories, string peoplys)
        {
            myAPI.AddEntry(name, path, description, location, dt, categories, peoplys);
        }

        1 reference
        public bool existPerson(string nameP)
    }
}

```

Figure 5: MyPhotosService

Este utilizat spatiul de nume System.ServiceModel pentru a ne putea folosi de [ServiceContract] si [OperationContract]. Cele doua sunt necesare pentru a putea realiza comunicarea serviciului catre un client (host).

- Atributul [ServiceContract] este de obicei atasat interfetei descrie, desi poate fi atasat si unei clase, si descrie metodele vizibile unei terte parti (altfel spus, clientului).
- Atributul [OperationContract] este folosit asupra metodelor ce vor fi vizibile

In cele din urma, pentru ca serviciile sa poata fi utilizate, se face legatura intre ObjectWCF si host prin intermediul fisierului de configuratie:

```

<services>
<service name="MyPhotosProject2.MyPhotosService" behaviorConfiguration="metadataSupport">
  <endpoint address="http://localhost:8000/PC" binding="basicHttpBinding" contract="MyPhotosProject2.IMyPhotosService" >
    <identity>

```

Figure 6: App.config