

TSP .NET

Documentație pentru WCF

Maxim Paul A6

Aprilie 2020

1 Introducere

Proiectul 'MyPhotos' cere managementul facil al fotografiilor și filmelor personale, cu ajutorul diverselor tehnologii din .NET Framework. Proiectul este separat în cinci foldere: Model, API, ObjectWCF, Host și GUI. Documentul prezintă vizează în special partea de ObjectWCF și în oarecare măsură și Host (WCF).

2 Interfața IMyPhotosService

Primul aspect care a trebuit hotărât a fost interfața pe care o voi expune ca serviciu. Practic, funcțiile din această interfață, atunci când vor fi implementate, vor trebui să apeleze API-ul, convertind când este cazul entitățile din DTO în non-DTO (și vice-versa). Am decis să am trei interfețe (**IPerson**, **ILocation**, **IMultimedia**), ce vor fi unite într-una singură (**IMyPhotosService**). Bineînțeles, mă folosesc de **System.ServiceModel** pentru a specifica atributele [**ServiceContract**] și [**OperationContract**]. DTO-urile se află în Model. Multe funcții sunt destul de simple, niște exemple fiind: 'AddLocation', 'UpdatePerson', 'GetEveryMultimedia', etc.

3 Clasa MyPhotosService

Clasa **MyPhotosService** implementează interfața **IMyPhotosService**, descrisă anterior. Se folosește de un **AutoMapper**, mai exact, de un **MapperConfiguration** (ce configurează ce entități convertim cu automapper-ul) și o interfață **IMapper** (cu care vom converti efectiv). În constructorul clasei se inițializează acestea două, și au ca scop convertirea din DTO în non-DTO și invers. În rest, funcțiile sunt implementate toate cam pe același șablon: se apelează API-ul cu entități non-DTO (se convertește înainte dacă este cazul) și returnăm entități DTO (desigur, dacă este cazul, putem return și numere întregi

sau valori boolene). Ne folosim de funcția `.Map()` a instanței 'mapper' definită în clasă ca să convertim. Clasa, bineînțeles, are o referință către MyPhotos API.

4 Host WCF

Hostul este foarte simplu, acesta folosindu-se de `ServiceHost` și `ServiceEndpoint` pentru a lansa server-ul la adresa 'http://localhost:8000/PC' (într-o aplicație adevărată, ar fi fost înlocuită adresa cu una reală). Se afișează endpointurile și se așteaptă conexiuni. Singura referință a Host-ului este către `ObjectWCF`, deoarece avem nevoie, atunci când instanțiem `ServiceHost`-ul, să-i dăm ca parametru tipul **MyPhotosService**, clasa descrisă anterior. Pentru a se folosi de serviciu, clientul trebuie să primească fișierele 'proxy.cs' și 'app.config', ambele fiind generate din Developer Command Prompt for VS 2019, prin comanda 'svcutil'. Clientul (**MyPhotosGUI**) NU are vreo referință către celelalte proiecte din cadrul soluției.

5 Folosirea serviciului de către client

Clientul se folosește de clasa `MyPhotosServiceClient`, aceasta expunând funcțiile cu care putem lucra. Exemplu:

```
private void SaveMultimedia()
{
    MultimediaDTO multimedia = new MultimediaDTO()
    {
        Id = currentMultimedia.Id,
        Path = currentMultimedia.Path,
        Description = textBoxGeneralDescription.Text,
        Event = textBoxGeneralEvent.Text,
        Weather = (WeatherDTO) comboBoxGeneralWeather.SelectedItem,
        DateCreated = dateTimePickerDateCreated.Value,
        DateAdded = currentMultimedia.DateAdded,
        DateModified = DateTime.Now,
        Locations = navigatingLocationsGroupBox.GetLocations().ToArray(),
        People = navigatingPeopleGroupBox.GetPeople().ToArray(),
        AdditionalLabels = textBoxAdditionalLabels.Text
    };

    MyPhotosServiceClient client = new MyPhotosServiceClient();
    client.UpdateWholeMultimedia(multimedia);

    SetCurrentMultimedia(currentMultimedia.Id);
}
```

Figure 1: Exemplu de folosire al serviciului

6 Încheiere

Subsemnatul **Maxim Paul** declar pe propria raspundere ca acest cod nu a fost copiat din Internet sau din alte surse. Pentru documentare am folosit urmatoarele surse:

Cărți: -

Link-uri:

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