

MyPhotos

Grigoraş Alexandru-Ionel

Universitatea Alexandru Ioan Cuza
Facultatea de Informatică
Martie 2020

Cuprins

1	Definirea contractelor	3
2	Implementarea contractelor	4
3	Declarație	4

1 Definirea contractelor

La această etapă m-am ocupat de definirea contractelor. Mai exact am creat interfețele **IMedia** și **IPhotoManagerService** la care am adăugat atributul [ServiceContract] pentru a specifica faptul că interfața respectivă reprezintă un contract. În interiorul interfețelor se află definite signaturile unor metode ce vor reprezenta operațiile din contract. Cele din urmă vor avea atributul [OperationContract] pentru a specifica faptul că reprezintă operații din contract.

Operațiile din contract sunt asemenea cu API-ul static. Astfel avem:

- C: void SaveMedia(MediaDTO mediaDTO)
- R: MediaDTO GetMediaById(Guid id)
- R: List<MediaDTO> GetAll()
- R: List<MediaDTO> GetAllWhere(string toSearch, bool byDate, bool byEvent, bool byPeople, bool byLocation, bool byTags, bool byDescription)
- U: void UpdateMedia(Guid id, MediaDTO mediaDTO)
- D: void DeleteMedia(MediaDTO mediaDTO)
- D: void DeleteMediaById(Guid id)
- bool AlreadyInDatabaseAndNotDeleted(string path)

Pentru a nu expune clasa Media către exterior, am folosit un DTO declarat în proiectul cu baza de date. Astfel dacă dorim să schimbăm ce expunem către exterior ne este foarte ușor să ascundem anumite proprietăți ai clasei Media. MediaDTO va fi clasa la care utilizatorul va avea voie. El va putea folosi această clasă și va avea acces doar la ce expune clasa respectivă.

Ar mai fi de specificat faptul că IPhotoManager reprezintă clasa ce moșteneste restul interfețelor. Aceasta va fi implementată la punctul următor. Deoarece am avut doar o tabelă în baza de date, aceasta moșteneste doar contractul pentru Media. Dacă aveam multiple contracte, atunci aceasta ar fi moștenit și restul.

2 Implementarea contractelor

Implementarea a fost făcută în clasa **PhotoManagerService**. Aceasta are un constructor unde este configurat AutoMapperul ce mă ajută să transform un obiect de tip Media în obiect de tip MediaDTO și invers, sau același lucru dar lucrând cu liste.

Pentru implementarea fiecărei operații din contract m-am folosit de API-ul static. Având în vedere faptul că expun un DTO, înainte să apelez funcțiile din API folosesc mapperul pentru a transforma din MediaDTO în Media și invers.

3 Declarație

Subsemnatul Grigoraș Alexandru-Ionel declar pe propria raspundere ca acest cod nu a fost copiat din Internet sau din alte surse. Pentru documentare am folosit urmatoarele surse:

→Carti: -

→Link-uri:

- [ServiceContractWCF](#)
- [ImplementServiceContractWCF](#)