

Subsemnatul Strugari Stefan, declar pe propria raspundere ca acest cod nu a fost copiat din Internet sau din alte surse.

Raport

ABSTRACT

Acest raport specifica detaliile cu privire la tehnologiile folosite, etapele implementarii si design pattern-urile folosite in cadrul Proiectul 1 a [temei de laborator](#).

TEHNOLOGII

- .NET Framework v4.7.2
- Entity Framework v6.2.0
- ADO.NET

ETAPE

- optarea pentru modelul de programare "EF Model Designer First". M-am folosit de ADO.NET entity data model pentru a construi baza de date si un model initial.
- modificarea modelului generat prin incapsularea proprietatilor, adugarea constructorilor, validarea proprietatilor folosind Data Annotations sau Fluent API.
- construirea unor repository-uri menite sa furnizeze functionalitatile CRUD pentru fiecare entitate in parte.
- implementarea unui Repository Wrapper care va constitui principalul obiect folosit din cadru API-ului.

ARHITECTURA

API-ul poate fi vazut ca un Data Access Layer. Incearca sa implementeze Repository Pattern. Per total respecta principii arhitecturale moderne, in special loose coupling si open-closed principle, in general alte princii SOLID.

MODIFICARI – 27.03.2019

Precizez ca au fost efectuate numeroase modificari si nu am tinut evidenta acestora in mod punctual. Din acest motiv le voi evidentia doar pe cele majore, la modul general:

- schimbarea unor relatii intre entitati;
- fixarea unor constrangeri care sa permita stergerea in cascada a instantelor oricarei entitati;
- renuntarea la incapsularea proprietatilor din modele, fapt cauzat absentei dto-urilor mapate la modele, prin care in mod normal s-ar accesa datele din cadrul API-ului;

Subsemnatul Strugari Stefan, declar pe propria raspundere ca acest cod nu a fost copiat din Internet sau din alte surse.

- adaugarea metodelor prin care se pot detasa instante ale entitatilor asociate prin relatii many-to-many.

LINK-URI

- [EF - Model Designer First](#)
- [EF Utilizare](#)
- [Repository Pattern](#)