
FacePlace

Stefan Sentić, 15359

Stefan Stanković, 15382

FacePlace

Društvena mreža za predlog destinacija za putovanja

Sadržaj

Sadržaj	0
Struktura projekta	0

Struktura projekta

Aplikacija *FacePlace* koristi Redis za keširanje često korišćenih podataka i generisanje jedinstvenih ID-eva i Neo4J kao bazu u kojoj se perzistiraju svi podaci i vrše složenije pretrage.

Aplikacija je izrađena u .NET- u a *Solution* sadrži 3 projekta- *FacePlace*, *FacePlace.DataLayerService* za rad sa Neo4J-om i *FacePlace.CachingService* za rad sa Redisom. U oba ova projekta se u folderu *Configuration* nalaze odgovarajuće klase i interfejsi za povezivanje sa bazama.

Model podataka se nalazi u folderu *Model* u *FacePlace.DataService*-u. U folderu *Repository* nalazi se generički interfejs:

```
public interface IRepository<Type>
{
    Type Get(string identifier);
    Type Create(Type typeInstance);
    void Delete(string identifier);
    Type Update(Type typeInstance);
    List<Type> GetAll();
}
```

Ovaj interfejs za konkretnu klasu iz modela implementiraju *Repository* klase, *PictureRepository*, *PlaceRepository* i *UserRepository*. U ovim klasama se nalaze upiti za rad sa Neo4J-om i one se instanciraju isključivo u klasi *DataLayerService*.

U projektu *FacePlace.CachingService* nalazi se folder *RedisLibrary* a u njemu klase za rad sa Redisom. Obe klase se instanciraju u isključivo u *CachingService* klasi. Klase iz foldera *Utility* služe za instanciranje, serijalizaciju i deserijalizaciju objekata i za generisanje ključeva za Redis.

Funkcionalnosti sistema enkapsulirane su u klasi *BusinessLogic* u folderu *DataProcessing* u projektu *FacePlace*.

Ova klasa koristi instancu klase *DataService* koja objedinjuje klase *CachingService* i *DataLayerService* za rad sa podacima iz Redisa i Neo4j-a.