

VISOKA ŠKOLA STRUKOVNIH STUDIJA ZA INFORMACIONE I KOMUNIKACIONE TEHNOLOGIJE

INTERNET TEHNOLOGIJE

Dokumentacija

FitEnd

<https://github.com/JankoSoskic/FitEnd>

SADRŽAJ

1.Uvod

1.0. Tema projekta

1.1. Korisceni programski jezici

1.2. Opis funkcionalnosti

2.Mapa Projekta

2.1. Organizaciona sema

2.2. Kontroleri i opis

2.2.1. Register

2.2.2. Login

2.2.3. Exercises

2.2.4. ExerciseType

2.2.5. UserPlans

2.2.6. UnosInicijalnihPodatka

2.2.7. Admin

2.3. Slika baze podataka

2.4. Slika strukture fajlova

2.3.0. Solution

2.4.1. Domen

2.4.2. Application

2.4.3. Implementation

2.4.4. API

Uvod

1.0. Tema projekta

Tema projekta je aplikacija za fitnes goals,

Korisnik moze da se loguje, I kao ulogovan korisnik, moze da napravi plan za mrsavljenje koji automatski racuna plan I program za sledecih N dana

1.1. Korisceni programski jezici

Projekat koristi c#, struktura celog projekta je maksimalno generalizovana.

Projekat je razvijen u visual studio-u, a za engine baze koriscen je Microsoft-ov sql server.

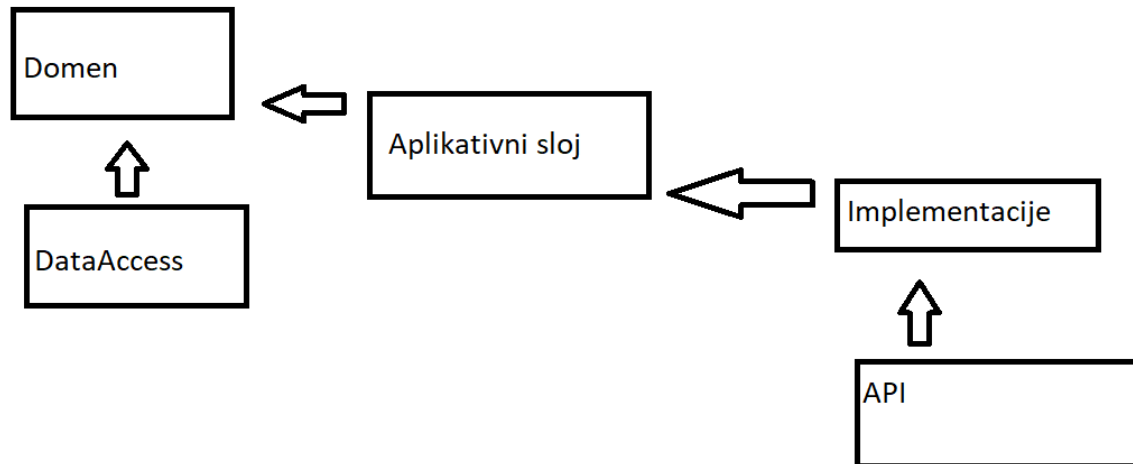
1.2. Opis funkcionalnosti

- Aplikacija je podeljena na 4. celine : Domenski , Aplikacioni , Implementacioni, dataAccess , I Api sloj.
- Domenski sloj je popunjen svim podacima kojima se moze baratati kroz aplikaciju.
- U aplikacionom sloju se nalaze sva moguca pravila koja se moraju ispuniti za isravni rad aplikacije.
- U implementacionom se nalaze zapravo sve implementacije aplikativnog sloja.
- data access je sloj gde se nalaze sva pravila za bazu podataka, I nacin podizanja baze.
- I api sloj kao vid dodirnih tacaka sa svim ostalim slojevima.
- Upotreba odgovarajucih statusnih kodova.
- Koriscen je globalni exception handler, zbog pucanja koda, ako se prepozna greska koja se desila, vraca se propratni statusni kod, ukoliko nije uhvacena, vraca se serverska greska.
- Upotreba JWT-a kao nacin registracije na aplikaciju.
- Iako se JWT koristi, on nije uslov za rad same aplikacije.
- Svaki EndPoint koji potencijalno menja bazu, se proverava upotrebom FluentValidator, I naravno vracanje propratnog statusnog koda.
- Upload fajla, upis u bazu fajl u vidu putanje do fajla, I prebacivanje fajla na server.
- Logovanje svakog moguceg pristupa stranici, upisa u tekstualni fajl je radjeno na ovom primeru.
- Iscitavanje tekstualnog fajla.
- Pretraga sa paginacijom.

- Konfigurisana cela baza podataka, I code first pristup rada sa bazom podataka.
- Mogucnost slanja mejla pri registraciji korisnika.
- Automtasko generisanje plana kako doci do zeljene tezine.
- Granulisanje privilegija po odredjenim Rolovima.
- Upotreba posebnog endpointa za Swagger.

Mapa projekta

2.1. Organizaciona sema



Svi endpointi su na api sloju

2.2. Kontroleri i opis

2.2.1. Register

Register kontroler ima jedan jedini endpoint, za registrovanje korisnika, kome se posalji podaci, gde je korisnik zatim registrovan u aplikaciji, I poslat mu je email da je uspela registracija.

2.2.2. Login

Login je takodje kontroler u kojem ima jedan jedini endpoint, na tom endpointu se posalju kredencijali za login, I ukoliko takva kombinacija postoji, pravi se JWT kljuc koji se moze posle prevesti u aktora aplikacije, ukoliko se taj kljuc posalje kroz zaglavlje http-a.

2.2.3. Exercises

Ovaj kontroler ovog puta ima malo vise akcija nego prosli, za pocetak, moze se raditi search svih vezba, ukoliko se proslede parametric za paginaciju podaci ce se slati po tim parametrima, ali ukoliko se ne posalju, uzece se podrazumevane vrednosti, I vratice se prva stranica paginacije. Takodje moze se na istom endpointu poslati opcioni parametar za tip vezbe, paginacija ce se automatski prilagoditi .

Zatim moze se pretraziti po nazivu (na drugom endpointu).

Admin na ovom endpointu moze odraditi nekoliko akcija, a to su ubacivanje nove vezbe, update vezbe, kao I brisanje vezbe. Naravno svuda se proverava input fluent validatorom, da li ugrozava bazu I vracanje odgovarajucih statusnih kodova. Takodje pri insert nove vezbe mogu se ubaciti jedna ili vise slika, koje bi trebalo da predstavljaju kako se vezbe pravilno rade, one se automatski prebacuju u wwwroot folder I putanja se zapisuje u bazu.

2.2.4. ExerciseType

Jedan standardni kontroler u kojem imaju najjobicnije CRUD operacije sa exercise type tabelom (insert, delete, i update samo admin moze).

2.2.5. UserPlans

Ovaj kontroler vec postaje jako zanimljiviji. Na ovom kontroleru postoji endpoint koji služi za kreiranje novog plana treniranja za mrsavljenje. Zbog zelje da bude sve sto je lakse moguće za korisnika, napravljen je sistem gde korisnik samo treba da upise koju kilazu zeli da ima (taj unos kilaze je proveren naravno), I do kada zeli da uspe u tom planu. Automatski ce se zatim napraviti plan kroz seriju matematickih operacija, kada radi vezbu koliko ponavljanja, koju vezbu, takodje je uracunat prosečni unos kalorija na dnevnom nivou, sto znaci da ishrana nece patiti (ako se naravno korisnik pridržava plana) . Ovde takodje postoji endpoint za brisanje plana, I pretragu plana. Ali ovde se I nalazi update tezine korisnika, iako se te informacije nalaze u drugoj tabeli.

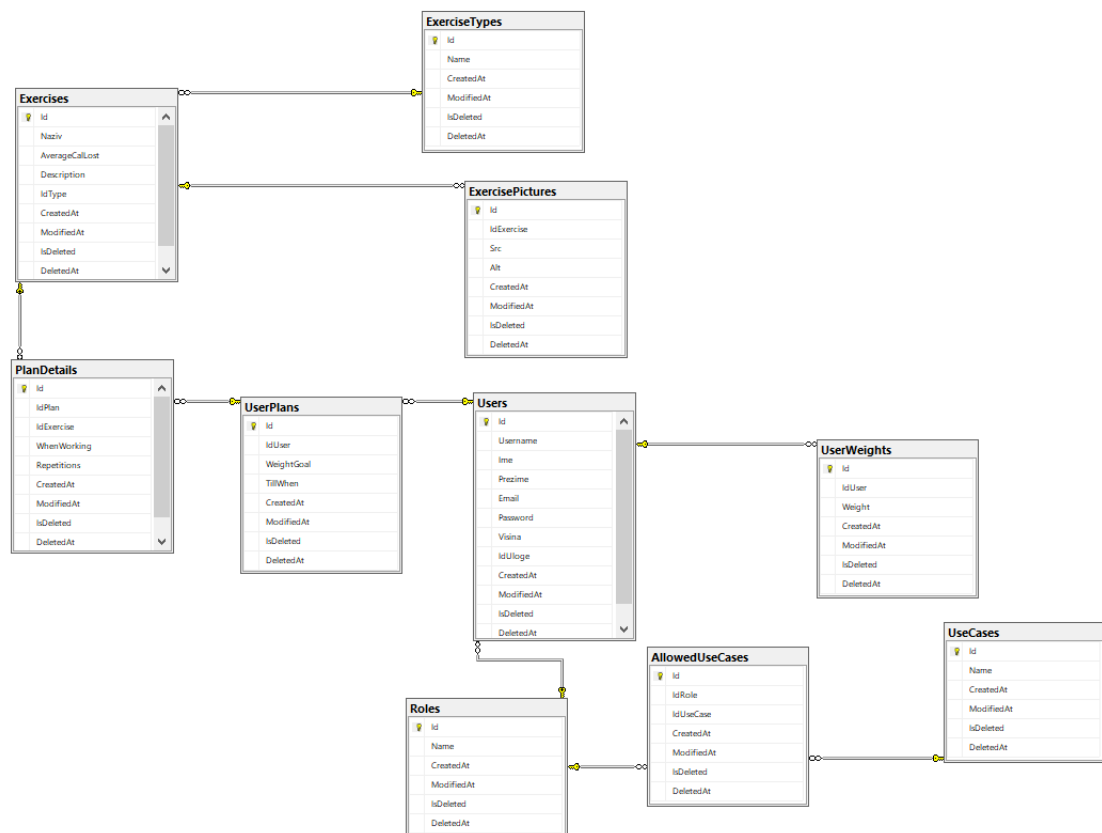
2.2.6. UnosInicijalnihPodatka

Ovaj kontroler ima jedan jedini endpoint koji služi kao seeder baze podataka, sve informacije koje su relevantne za inicijalno pokretanje aplikacije (Vezbe, Tipovi vezba, slike vezba, Uloge, korisnici itd....)

2.2.7. Admin

Poslednji ali ne I najmanje vazan kontroler, ovde se nalazi nekoliko endpointa koji su bitni za admina, mogu se dohvatiti svi rolovi, svi korisnici, specifični korisnik po imenu trazen, moze se updatovati Role korisnika, kao I da se izbrise neki korisnik. Takodje postoji endpoint koji moze pročitati informacije iz tekstualnog fajla (log fajla).

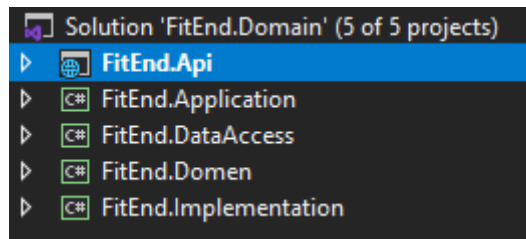
2.3. Slika baze podataka



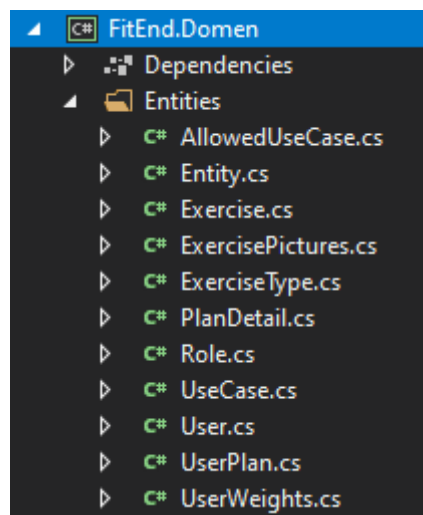
Baza je radjena Code-first pristupom

2.4. Slika strukture fajlova

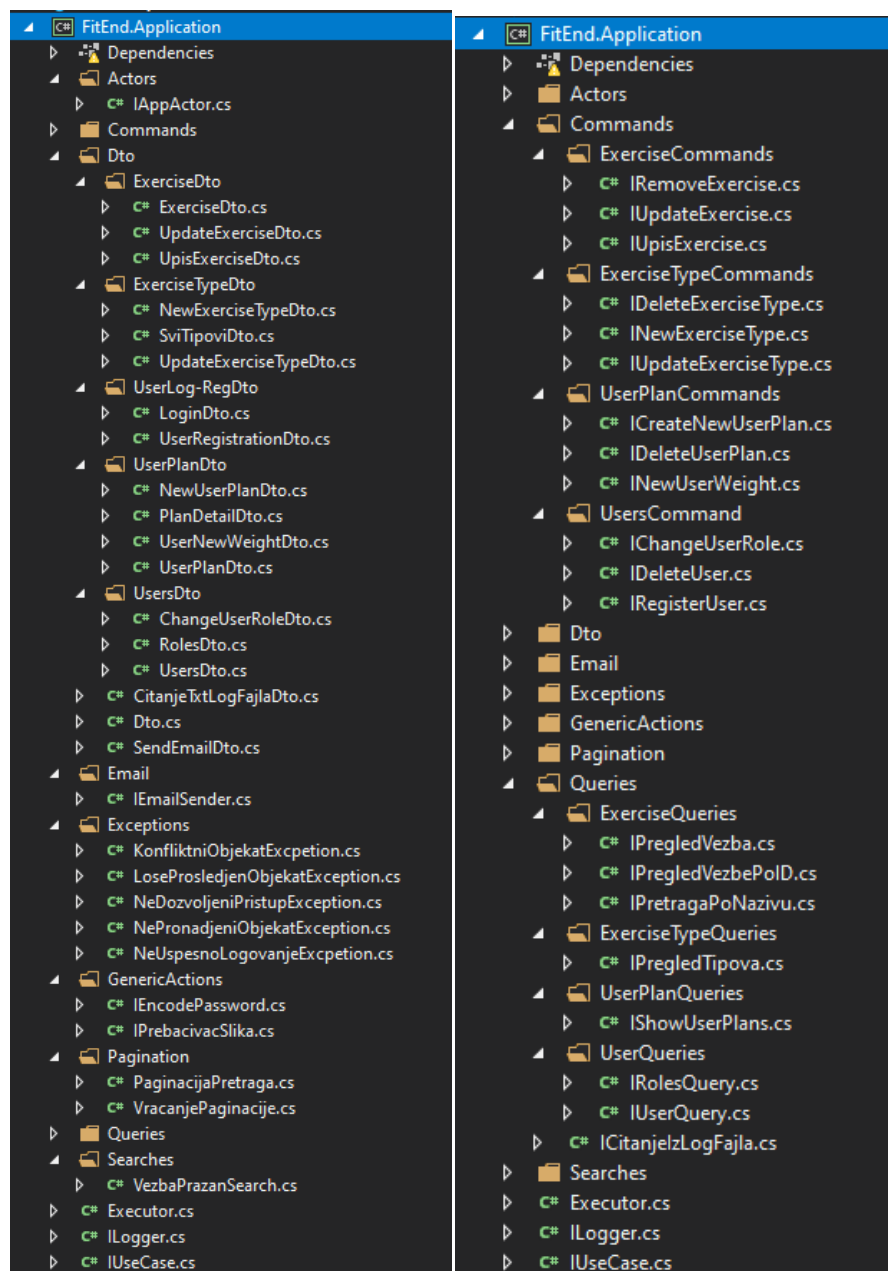
2.4.0. Solution



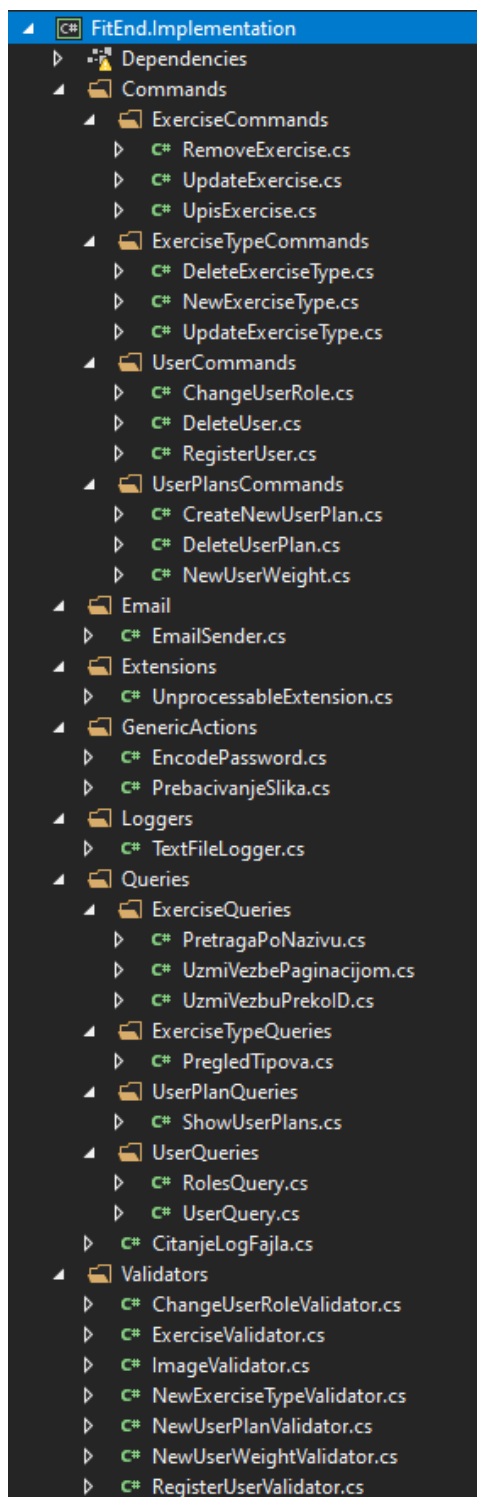
2.4.1. Domen



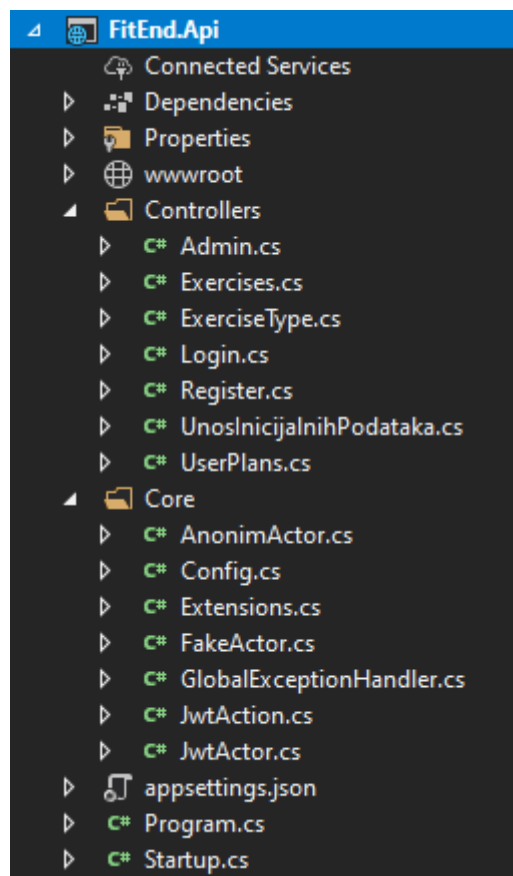
2.4.2. Application



2.4.3. Implementation



2.4.4. API



Hvala na citanju

Kraj.