

Projekt - Spirala 3

Tema aplikacije za projekat iz ovog predmeta biće mobilna aplikacija za fakultetske kvizove. Nakon završetka projekat aplikacija bi trebala biti upotrebljiva i sadržavati upotrebu svih tehnika i tehnologija koje budemo radili na predmetu.

Zadatak 1 [10B]

Koristeći web servis ([link](#)) implementirajte metode i repozitorije koji su navedeni u nastavku i povežite ih sa postojećim korisničkim interfejsom vaše aplikacije. Sve metode ovih repozitorija trebaju se izvršavati u okviru korutina koristeći IO dispatcher. Sve klase u nastavku trebaju se nalaziti u paketu **ba.etf.rma21.projekat.data.repositories**. Početna aktivnost treba imati naziv **MainActivity** i nalaziti se u paketu **ba.etf.rma21.projekat**.

Klase koje je potrebno implementirati:

ApiConfig - klasa koja čuva base url na web servis, defaultni je:

<https://rma21-etf.herokuapp.com>

- `postaviBaseURL(baseUrl:String):Unit` - postavlja base url za web servise, ukoliko se ova metoda ne pozove koristi se defaultni url

AccountRepository

- `postaviHash(acHash:String):Boolean` - postavlja (lokalno, ne putem web servisa) hash studenta koji će biti korišten u drugim repozitorijima, vraća true ukoliko je hash postavljen, false inače.
- `getHash():String` - vraća hash studenta koji je postavljen

PitanjeKvizRepository

- `getPitanja(idKviza:Int):List<Pitanje>` - vraća sva pitanja na kvizu sa zadanim id-em

TakeKvizRepository

- `zapocniKviz(idKviza:Int):KvizTaken` - kreira pokušaj za kviz, vraća kreirani pokušaj ili null u slučaju greške
- `getPocetiKvizovi():List<KvizTaken>` - vraća listu pokušaja ili null ukoliko student nema niti jedan započeti kviz

OdgovorRepository

- getOdgovoriKviz(idKviza:Int):List<Odgovor> - vraća listu odgovora za kviz, praznu listu ako student nije odgovarao ili nije upisan na zadani kviz
- postaviOdgovorKviz(idKvizTaken:Int,idPitanje:Int,odgovor:Int):Int - funkcija postavlja odgovor sa indeksom odgovor na pitanje sa id-em idPitanje u okviru pokušaja sa id-em idKvizTaken. Funkcija vraća ukupne bodove na kvizu nakon odgovora ili -1 ukoliko ima neka greška u zahtjevu

KvizRepository

- getAll():List<Kviz> - vraća listu svih kvizova
- getById(id:Int):Kviz - vraća jedan kviz koji ima zadani id ili null ako kviz ne postoji
- getUpisani():List<Kviz> - vraća listu svih kvizova za grupe na kojima je student upisan

PredmetIGrupaRepository

- getPredmeti():List<Predmet> - vraća sve predmete
- getGrupe():List<Grupa> - vraća sve grupe
- getGrupeZaPredmet(idPredmeta:Int):List<Grupa> - vraća grupe na predmetu sa id-em idPredmeta
- upisiUGrupu(idGrupa:Int):Boolean - upisuje studenta u grupu sa id-em idGrupa i vraća true ili vraća false ako nije moguće upisati studenta
- getUpisaneGrupe():List<Grupa> - vraća grupe u kojima je student upisan

Napomene

- Rok za rad na spirali 3 je **03.06.2021** do **23:59**. Rješenje se treba nalaziti na vašem privatnom repozitoriju (na master/default branchu). Prethodnu spiralu postavite na branch spirala2 (iz master brancha napravite branch spirala2 i onda se vratite na master da radite treću spiralu)
- Testovi se nalaze na repozitoriju (**biće postavljeni naknadno**)
- **Nakon implementacije zadataka početni testovi moraju prolaziti**
- Implementirajte sve u skladu sa **MVVM** arhitekturom. U slučaju da ne koristite MVVM arhitekturu možete ostvariti maksimalno 80% bodova spirale.
- Spirala se pregleda korištenjem testova. Ispoštujte konvenciju imenovanja datu u postavci. Određeni testovi su sakriveni i izvršit će se tokom pregledanja spirale.
- Minimalni SDK je 7.0.
- Pitanja vezana za postavku spirale postavljajte kroz forumsku diskusiju na c2: <https://c2.etf.unsa.ba/mod/forum/view.php?id=81064>