

Spirala 3

Uvod

Na ovoj spirali ćemo koristiti *Firestore* i njegov web API (web servis). Firestore je baza podataka koja je orijentisana na dokumente. Nema redove i tabele kao tipična relacionalna baza podataka. Podaci su organizovani u kolekcije i dokumente. *Dokumenti* predstavljaju skup podataka organizovanih kao ključ i vrijednost. *Kolekcija* je skup dokumenata. Firestore nudi REST web API za rad sa firestore bazom podataka. Pomoću API-a moguće je kreirati, dohvatati, ažurirati i brisati dokumente i kolekcije. Da bi mogli pristupiti bazi podataka koju napravite na firestore-u potreban vam je ključ za pristup (**access token**). Ovaj ključ je string kojeg trebate slati uz svaki zahtjev kako bi mogli raditi sa vašom privatnom bazom podataka. Za dobijanje ključa za pristup potrebno je da napravite račun za servis (**service account**). Service account su korisnički podaci koje vaša aplikacija ima i sa kojim može tražiti pristup firestore bazi podataka.

Pristupati bazi **isključivo** kroz HTTP/REST API (web servis) [API dokumentacija](#). Zabranjeno je korištenje firestore biblioteke.

Primjer, dohvaćanje svih kvizova. Poslati GET zahtjev, na url:

[https://firestore.googleapis.com/v1/projects/PROJEKATID/databases/\(default\)/documents/kviz?access_token=TOKEN](https://firestore.googleapis.com/v1/projects/PROJEKATID/databases/(default)/documents/kviz?access_token=TOKEN)

PROJEKATID - predstavlja id projekta kojeg možete vidjeti u Firebase konzoli (npr. rmatest-6f71c)

TOKEN - predstavlja access token (string). Način kako je moguće dobiti token opisan je u nastavku.

Koraci koje je potrebno napraviti prije implementacije zadatka dati su u nastavku.

1. Napraviti projekat na Firebase-u:

[Firebase konzola](#)

2. Napraviti na Firebase-u bazu podataka (zbog sigurnosti obavezno postavite opciju 'Start in locked mode') i kolekcije za:

Naziv kolekcije	Polja koja dokumenti u kolekciji trebaju imati
Kvizovi	naziv (string), idKategorije (string, id dokumenta kategorije), pitanja (array, elementi su id-evi dokumenta pitanja)
Pitanja	naziv (string), indexTacnog (number), odgovori (array stringova)
Kategorije	naziv (string), idIkonice (number)
Rangliste	nazivKviza (string), lista (mapa - ključ je pozicija, a vrijednost druga mapa sa ključem ime igrača i vrijednost procenat tačnih odgovora)

3. Napraviti service account za projekat i dodijelite mu neko ime; U drugom koraku odaberite rolu 'Cloud Datastore User'. Dobijeni json fajl snimiti u raw/ resource folder i postavite mu naziv secret.

[Kreiranje service accounta](#)

4. Dobiti access token za pristup bazi (koristeći Google API klijent), više detalja na linku ispod: [Dokumentacija OAuth2 Service Account](#)
5. Nakon što ste pročitali dokumentaciju na sljedeći način možete dobiti access token unutar android aplikacije:

- Dodati biblioteke:

```
implementation 'com.google.api-client:google-api-client:1.22.0'  
implementation 'com.google.api-client:google-api-client-android:1.22.0'
```

- Kod za dohvaćanje tokena za pristup bazi:

```
InputStream is = getResources().openRawResource(R.raw.secret);  
GoogleCredential credentials = GoogleCredential.fromStream(is).  
createScoped(Lists.newArrayList("https://www.googleapis.com/auth/datastore"));  
  
credentials.refreshToken();  
String TOKEN = credentials.getAccessToken();
```

Zadaci

Korištenjem AsyncTask klase i ostalog znanja stečenog na vježbama implementirajte sljedeće zadatke:

Zadatak 1. [1.8B] Pri kreiranju novog kviza u firestore kolekciju dodati novi dokument za taj kviz. Kada se kviz edituje snimiti sve promjene i u njegovom dokumentu u firestore bazi. Nakon dodavanja/izbacivanja pitanja iz kviza, ažurirajte listu pitanja u dokumentu kviza na Firebase-u. Ako se kreira novo pitanje upišite ga u kolekciju Pitanja.

Zadatak 2. [1.8B] U moguća pitanja učitajte sva pitanja koja nisu dodana na kviz, a nalaze se u Firebase kolekciji Pitanja.

Zadatak 3. [1.8B] Nakon kreiranja kategorije upišite je u Firebase kolekciju Kategorije. Ako kategorija već postoji ispišite grešku "Unesena kategorija već postoji!" i ne dozvolite kreiranje te kategorije.

Zadatak 4. [1.8B] Napravite da se nakon promjene kategorije i pri prvom pokretanju aplikacije kvizovi učitavaju iz kolekcije Kvizovi sa Firebase-a. Dohvatite samo one kvizove koji su u odabranoj kategoriji.

Zadatak 5. [1.8B] Nakon završetka kviza pitajte korisnika da unese ime (putem AlertDialog) i nakon toga prikažite rang listu za taj kviz. Za prikazivanje rang liste napravite fragment RangLista. U ovom fragmentu se treba prikazati listview u kojem svaki red predstavlja jedan element iz niza lista dokumenta RangLista sa Firebase-a. U elementu prikažite redni broj pozicije, ime igrača i procenat tačnih odgovora. Ovaj fragment ubacite u mjesto gdje se nalazio fragment Pitanja.

Rok za rad na spirali je 31.05.2019 do 23:59. Nakon završetka roka na vašim repozitorijima dobit ćete projekte pet drugih studenata. Za ove projekte ćete trebati napraviti code review koji će biti ocijenjen sa maksimalno 1B. Detaljnija uputstva za review dobit ćete pravovremeno.

Pitanja vezana za postavku spirale postavljajte na c2 forumu.