Univerzitet u Nišu

Elektronski fakultet



Sistemi za upravljanje bazama podataka

Database as a Service na primeru Firebase baze podataka

- Seminarski rad -

Mentor: Prof. Aleksandar Stanimirović Kandidat: Pavle Stojanović 1148

Jun 2021.

Contents

[Database as a service 3](#_Toc75695969)

[Firebase 4](#_Toc75695970)

[Firestore baza podataka 5](#_Toc75695971)

[Postavljanje Firestore baze podataka 6](#_Toc75695972)

[Struktura podataka Firestore-a 8](#_Toc75695973)

[Hijerarhija podataka 10](#_Toc75695974)

[Podkolekcije 11](#_Toc75695975)

[Literatura 13](#_Toc75695976)

# Database as a service

Baza podataka-kao-usluga (DBaaS), poznatija kao „Upravljane baze podataka“, uzela je maha kada je AWS predstavio svoju uslugu relacionih baza podataka 2009. godine. Od tada je postao najbrže rastući cloud servis sa nekim procenama koje projektuju tržište od 320 milijardi dolara do 2025. godine. Razlog za ovaj meteorski rast je dokazana vrednost baze podataka kao usluge u omogućavanju brzog izlaska na tržište poboljšanjem produktivnosti, standardizacije i sigurnosti podataka.

Sa Database as a service modelom, vlasnici aplikacija ne moraju sami instalirati i održavati bazu podataka. Umesto toga, provajder usluga baze podataka preuzima odgovornost za instaliranje i održavanje baze podataka, a vlasnici aplikacija plaćuju tu uslugu na osnovu korišćenja iste.

Ključne karakteristike Sa Database as a service modela:

* Self-service: DBaaS omogućava klijentima da bez previše znanja o bazama podataka koriste usluge iste.
* On-demand . Plaćate za ono što koristite, čime se ostvaruje ušteda.
* Dynamic. Pruža fleksibilnu platformu baze podataka koja se prilagođava trenutnim potrebama na osnovu dostupnih resursa.
* Sigurnost. Tim stručnjaka koji vam stoji na raspolaganju kontinuirano nadgleda vašu bazu podataka.
* Automatizacija. Automatizira administraciju i nadzor baze podataka.

DBaaS oslobađa vaše IT odeljenje stresa i u celini olakšava život vašoj kompaniji. Pored čiste pogodnosti, dodaje nivo složenosti vašoj bezbednosnoj strukturi proaktivnim upravljanjem vašim okruženjem. Imati tim stručnjaka na raspolaganju presudno je za pravilnu funkcionalnost, podešavanje i kontinuirano nadgledanje vaših baza podataka koje doprinose sveukupnoj sigurnosti.

Većina organizacija koje koriste DBaaS rešenja sposobne su uštedeti novac i resurse efikasnijim recikliranjem servera i skladištenja. Uz kvalifikovane administratore baza podataka koji rade na optimizaciji performansi sistema, zaposleni se sada mogu usredsrediti na svakodnevne IT operacije i povećati svoju produktivnost.

Važno je shvatiti da kao i druge tehnologije u oblaku, DBaaS ima dva primarna potrošača:

* IT organizacija koja upravlja i održava oblak
* Krajnji korisnik koji troši resurse u oblaku, obično programeri i DevOps.

IT organizacija primenjuje DBaaS rešenje omogućavajući krajnjim korisnicima (programeri i DevOps) da obezbede bazu podataka po svom izboru, na zahtev, iz kataloga podržanih baza podataka, koja može sadržavati i relacione i nerelacione baze podataka. IT organizacija može konfigurirati DBaaS da podržava specifična izdanja ovih softverskih naslova i može dalje ograničiti konfiguracije koje određeni korisnici mogu pružiti. Na primer, programerima može biti dozvoljeno da bazama podataka pružaju mali memorijski otisak koristeći tradicionalne diskove, dok bi DevOps mogao da obezbedi servere većeg kapaciteta sa SSD-ovima. I na kraju, IT organizacija može postaviti politike za standardne operacije baza podataka, kao što su sigurnosne kopije, DR i sigurnosne politike, kako bi se osiguralo da se podaci s vremena na vreme pravilno čuvaju kako bi se omogućio oporavak kada je to potrebno.

Tipično bi krajnji korisnik pristupio sistemu DBaaS putem portala koji nudi izbor različitih naslova baze podataka i u raznim različitim opcijama konfiguracije. Sa nekoliko klikova, korisnik određuje potrebnu bazu podataka i odgovarajuću konfiguraciju za obezbeđivanje. DBaaS sistem brzo obezbeđuje bazu podataka i vraća krajnju tačku koja može da traži, kao što su:

*mysql://192.168.15.243:3306/*

Korisnik to onda može koristiti direktno u aplikaciji. DBaaS sistem pruža jednostavne mehanizme za dodavanje korisnika, kreiranje šema i dodeljivanje dozvola različitim korisnicima kako zahteva aplikacija.

Razvoj robusnih i visokokvalitetnih mobilnih aplikacija zahteva puno posvećenosti, ali što je još važnije, moćne razvojne platforme. Firebase je jedna takva platforma koja je osvojila programere širom sveta.

# Firebase

Firebase pruža programerima mnoge mogućnosti za stvaranje visoko funkcionalnih i svestranih aplikacija na web, Android i iOS platformi. Iako u prostoru za razvoj aplikacija ima nekoliko kandidata, Firebase uvek razgovara sa najboljim trenutno dostupnim platformama.

Firebase je u osnovi kolekcija alata na koje se programeri mogu osloniti, kreirajući aplikacije i proširujući ih na osnovu potražnje.

Firebase ima za cilj da reši dva glavna problema za programere:

* Napraviti aplikaciju brzo
* Izbaciti i nadgledati aplikaciju

Programeri koji se oslanjaju na ovu platformu dobijaju pristup uslugama koje bi morali sami da razvijaju, a to im omogućava da se usredsrede na pružanje robusnih iskustava u primeni.

Neke istaknute funkcije platforme Google Firebase uključuju baze podataka, potvrdu identiteta, push poruke, analitiku, skladištenje datoteka i još mnogo toga.

Pošto su usluge hostovane u oblaku, programeri mogu nesmetano da izvrše skaliranje na zahtev bez ikakvih muka. Firebase je trenutno među najboljim platformama za razvoj aplikacija na koje se oslanjaju programeri širom sveta.

Evo nekoliko primera Firebase-ovih klijenata.

* Alibaba
* The Nev Iork Times
* Todoist
* Le Figaro
* eBai Motors

Evo deset najboljih prednosti korišćenja Firebase-a.

* Slobodan na početku
* Brzina razvoja
* Platforma za razvoj aplikacija od kraja do kraja
* Pokreće ga Google
* Programeri se mogu usredsrediti na frontend razvoj
* Serverless
* Nudi mogućnosti mašinskog učenja
* Generiše saobraćaj do vaših aplikacija
* Nadgledanje grešaka
* Sigurnost

Iako je fantastična platforma, Firebase takođe ima i svoje loše strane.

* Nije Open-Souce
* Firebase ne radi u mnogim zemljama
* Dostupne su samo NoSQL baze podataka
* Spori upiti
* Nisu sve usluge besplatne za pokretanje
* Nije jeftino i teško je predvideti cene
* Radi samo na Google Cloud-u
* Namenski serveri i podrška za preduzeća nisu opcija
* Ne pruža GraphQL API-je

# Firestore baza podataka

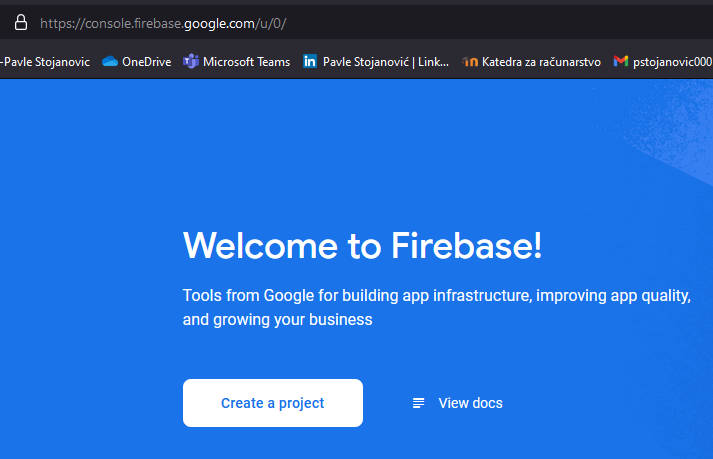
Cloud Firestore je fleksibilna, skalabilna baza podataka za razvoj IOS/Android ili web aplikacija. Poput Firebase baze podataka u stvarnom vremenu, ona sinhronizuje vaše podatke u klijentskim aplikacijama putem slušača u stvarnom vremenu i nudi offline podršku za mobilne i veb aplikacije, tako da možete da napravite responsive aplikacije koje rade bez obzira na kašnjenje mreže ili Internet vezu.

Model podataka Cloud Firestore podržava fleksibilne, hijerarhijske strukture podataka. Čuvajte podatke u dokumentima, organizovanim u zbirke. Dokumenti mogu da sadrže i složene ugnežđene objekte pored podzbirki. Možete da koristite upite za preuzimanje pojedinačnih, određenih dokumenata ili za preuzimanje svih dokumenata u kolekciji koji se podudaraju sa vašim parametrima upita. Vaši upiti mogu da uključuju višestruko ulančane filtere i kombinuju filtriranje i sortiranje. Oni su takođe podrazumevano indeksirani, tako da su performanse upita proporcionalne veličini vašeg skupa rezultata, a ne skupu podataka. Kao i baza podataka u realnom vremenu, Cloud Firestore koristi sinhronizaciju podataka za ažuriranje podataka na bilo kom povezanom uređaju. Međutim, takođe je dizajniran da efikasno čini jednostavne, jednokratne preuzimanja upita.

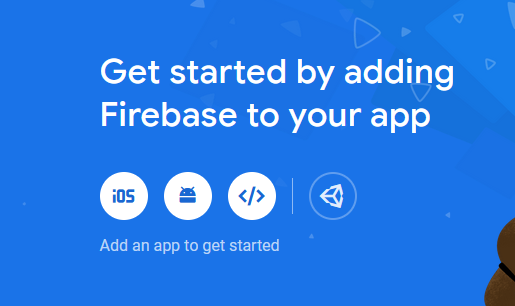
Cloud Firestore kešira podatke koje vaša aplikacija aktivno koristi, tako da aplikacija može da piše, čita, sluša i traži podatke čak i ako je uređaj van mreže. Kada se uređaj vrati na mrežu, Cloud Firestore sinhronizuje sve lokalne promene sa Cloud Firestore-om. Cloud Firestore donosi vam najbolje od moćne infrastrukture Google Cloud-a: automatsko kopiranje podataka u više regiona, jake garancije doslednosti, atomske batch operacije i stvarna podrška za transakcije. Dizajniran je da se nosi sa najtežim opterećenjima baze podataka iz najvećih svetskih aplikacija.

# Postavljanje Firestore baze podataka

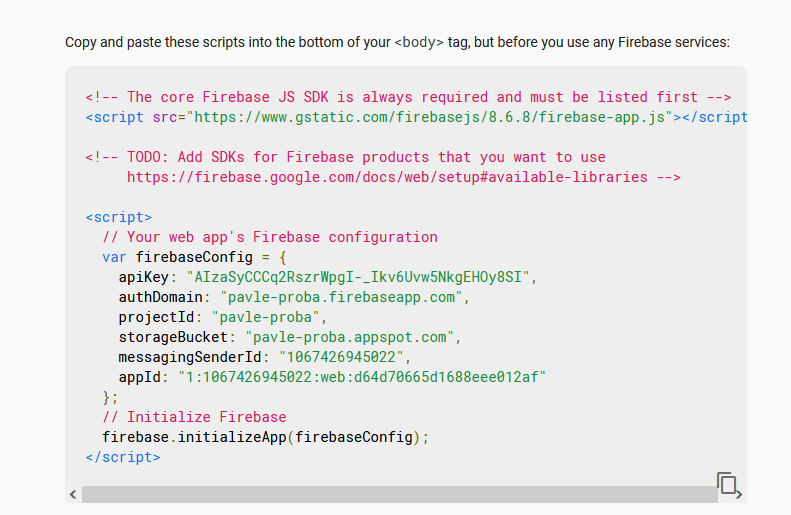
Za početak, poduzmimo nekoliko koraka za podešavanje Firebase baze podataka. Prvo, morate da napravite projekat u Firebase konzoli ako ga nemate.



Zatim jednostavno kliknite na dugme „Napravi bazu podataka“ da biste aktivirali uslugu u Firebase konzoli. Firebase nudi jednu bazu podataka po projektu besplatno, ograničenu na veličinu podataka od 1 GB sa dnevnim obimom čitanja od 50 000, pisanjem od 20 000 i brisanjem od 20 000. Ova besplatna ponuda je dovoljna da programeri pokrenu svoju inicijativu bez potrebe za plaćanjem infrastrukturne usluge. Mozete izabrati bazu podataka za IOS, Android ili Web aplikaciju.



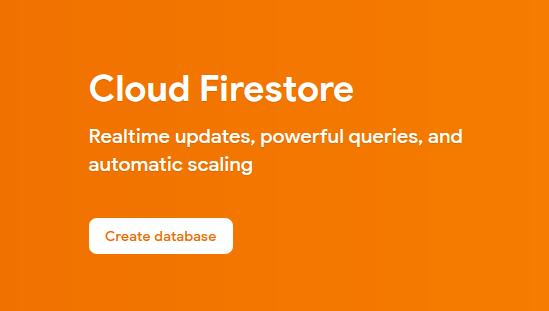
Nakon odabira firebase Vam generiše Firebase config koji koristite za pristup firebase-u iz vase aplikacije.



Dalje, možete odabrati lokaciju baze podataka. Ako nemate bilo kakav specifičan zahtev, podrazumevana opcija je da odaberete centar za više regija kako biste maksimizirali dostupnost i trajnost baze podataka jer će se vaši podaci replicirati u centre podataka u različitim regionima. Obratite pažnju na ovo jer ne možete promeniti centar podataka nakon stvaranja baze podataka.

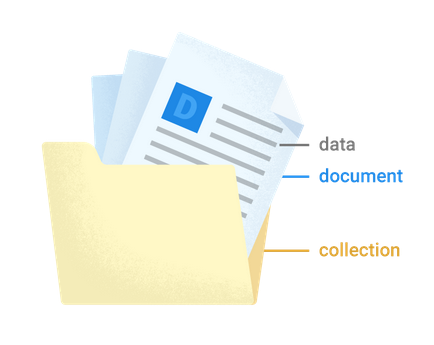
Sada je sve postavljeno i baza podataka je sada spremna za upotrebu! Zatim, pogledajmo podešavanje šeme podataka.

Firebase baza podataka ima 2 opcije - Realtime Database i Cloud Firestore. Generalno, Firestore je pogodan za većinu svrha jer nudi funkcije na visokom nivou, dok je baza podataka u realnom vremenu pozadinska baza podataka za aplikaciju sa specifičnom potrebom za pristup API-ju nižeg nivoa.



# Struktura podataka Firestore-a

Firestore je u osnovi baza podataka zasnovana na dokumentima, omogućava čuvanje čitave gomile dokumenata u kolekciji, a svaki dokument može imati različit skup atributa. Značajno se razlikuje od tradicionalne relacione baze podataka da šema podataka treba da bude fiksna i dobro definisana. Stoga, ako ste navikli da radite na relacionoj bazi podataka, možda cete morati da prilagodite svoj način razmišljanja.

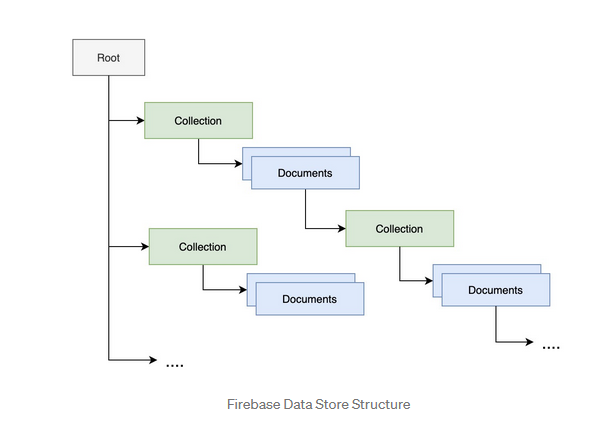


Firestore se sastoji od kolekcije dokumenata. Podaci uskladišteni u svakom dokumentu predstavljaju par ključ-vrednost. Takođe, podržava ugnežđenu strukturu tako da možemo čuvati kolekciju dokumenata unutar svakog dokumenta.

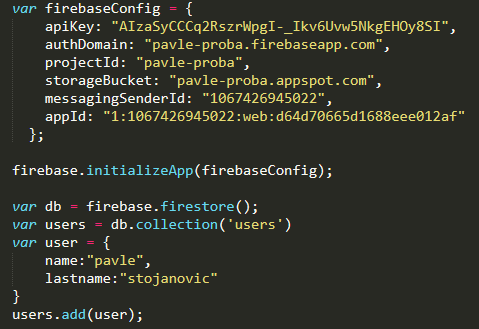
**Dokument = Red**

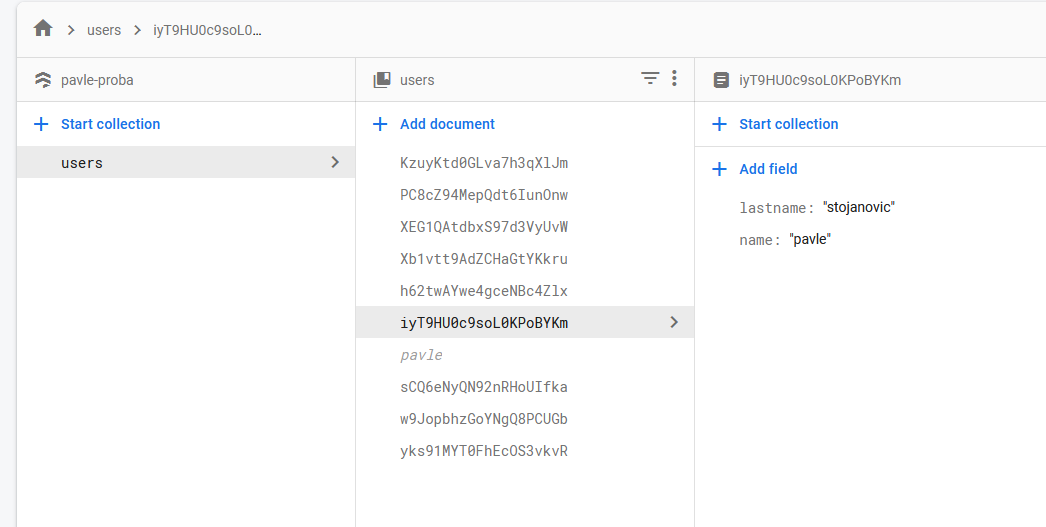
**Kolekcija = Tabela**

Dijagram u nastavku ilustruje kako izgleda struktura podataka:



Primer u nastavku ilustruje dodavanje dokumenta.





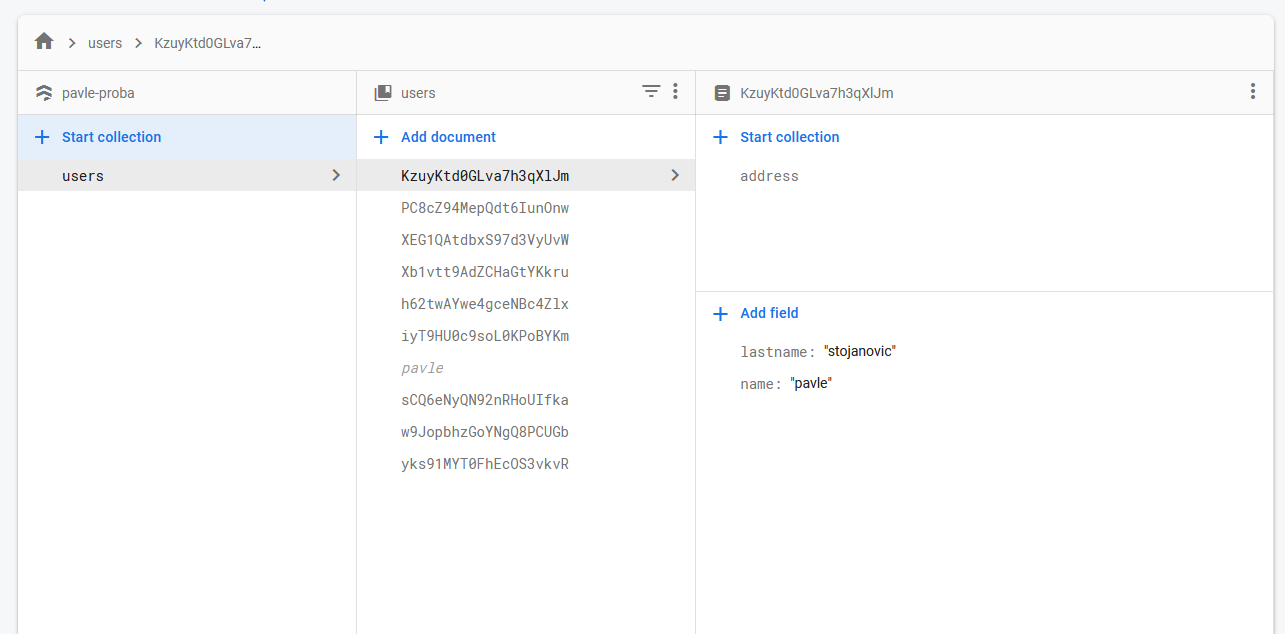
Cloud Firestore je schemaless, tako da imate potpunu slobodu nad poljima koja stavljate u svaki dokument i koje tipove podataka čuvate u tim poljima. Svi dokumenti u istoj kolekciji mogu sadržati različita polja ili u njima čuvati različite vrste podataka. Međutim, dobra ideja je da koristite ista polja i tipove podataka u više dokumenata, tako da možete lakše da ih tražite. Imena dokumenata u kolekciji su jedinstvena. Možete dati svoje ključeve, kao što su korisnički ID, ili možete da dozvolite Cloud Firestore-u da automatski kreira nasumične ID-ove za vas. Ne morate „kreirati“ ili „brisati“ kolekcije. Nakon što kreirate prvi dokument u kolekciji, kolekcija postoji. Ako izbrišete sve dokumente iz kolekcije, ona više ne postoji.

Svaki dokument u Cloud Firestore-u se jedinstveno identifikuje prema lokaciji u bazi podataka. Prethodni primer je pokazao da je dokument alovelace u users kolekciji. Možete stvoriti referencu na ovu lokaciju.

*const ref = db.doc('users/{id});*

## Hijerarhija podataka

Da biste razumeli kako hijerarhijske strukture podataka funkcionišu u Cloud Firestore-u, razmotrite primer primer korisnika za koga se vezuju adrese:

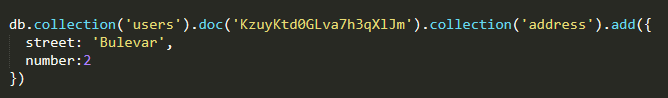


## Podkolekcije

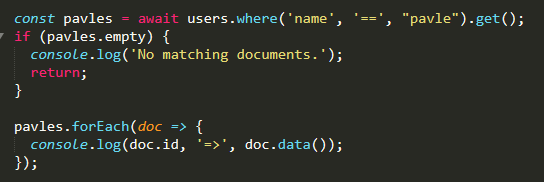
Najbolji način za skladištenje adresa u ovom scenariju je korišćenje podkolekcija (subcollections). Podkolekcija je kolekcija povezana sa određenim dokumentom.

Uočite obrazac kolekcija-dokument. Ne možete referencirati kolekciju u kolekciji ili dokument u dokumentu.

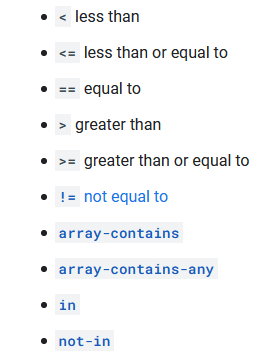
Sintaksa za dodavanje kolekcije unutar dokumenta:



Firestore omogućava kreiranje upita. Navodi se operator properti I vrednost. Sintaksa za upite data je u nastavku.



Operatori koje podržava firestore dati su u nastavku.



# Literatura

* <https://firebase.google.com/docs/build>
* <https://firebase.google.com/docs/firestore>
* https://fireship.io/lessons/firebase-quickstart/