

Laboratorijska vežba 5

Firestore

Opis projekta

Projekat predstavlja aplikaciju koja služi za dodavanje POI-a (Points of Interest) na Firebase Firestore bazu podataka. Arhitektura aplikacije je slična vežbi 3 ali se umesto Room baze podataka koristi Firebase Firestore implementiran servis (naziv za klasu koja povezuje aplikaciju sa spoljnim izvorima, ne Android servis) kojom, uz korišćenje Flow-a dodaje i preuzima podatke sa Firestore-a. Potrebno je kreirati projekat na Firebase-u a zatim i Firestore bazu a zatim je povezati sa postojećim kodom aplikacije. Uputstvo za povezivanje se nalazi u nastavku pripreme za laboratorijsku vežbu.

Opis postojeće strukture

Implementirani su sledeći ekrani:

1. **PoiListScreen** – prikazuje listu Poi-a koji su preuzeti iz baze podataka. Klikom na dugme View prelazi se na ekran za prikaz sadržaja Poi-a.
2. **ViewPoiScreen** – prikazuje sadržaj Poi-a.
3. **AddPoiScreen** – omogućuje dodavanje novih Poi-a.

Implementirani su sledeći ViewModel-i:

1. **PoiViewModel** – komunicira sa servisom za pribavljanje podataka sa *Firestore*-a korišćenjem Flow-a.
2. **EditViewModel** – koristi se za prenos unetih podataka o Poi-u.

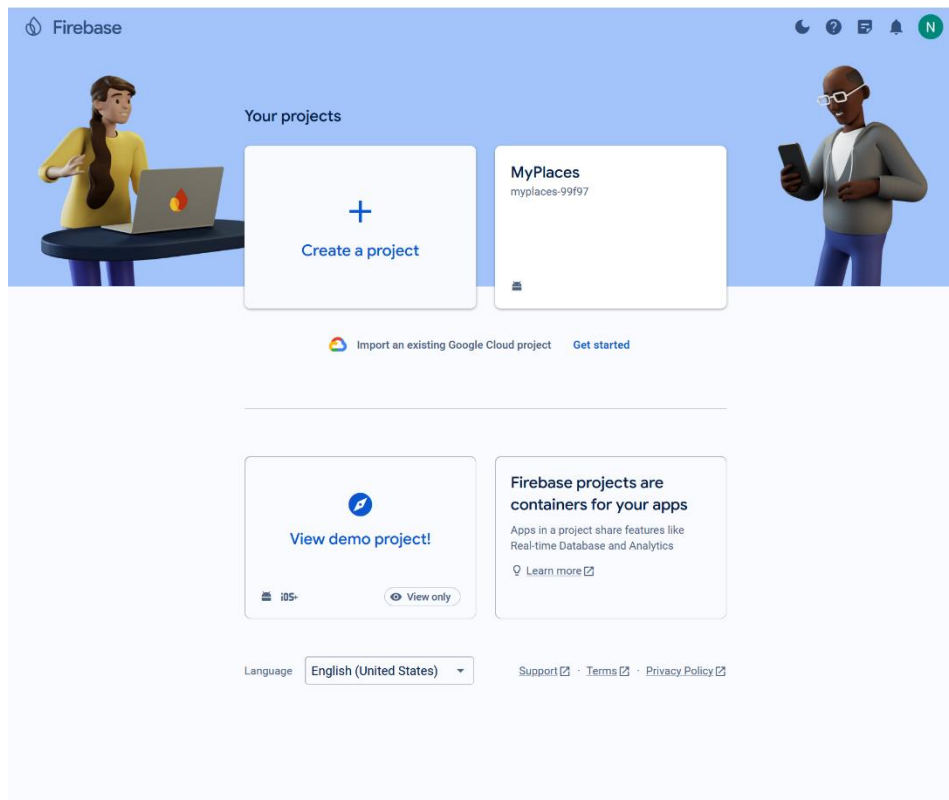
Model podataka je implementiran u data klasi Poi i sadrži ime i adresu Poi-a. Preostala dva podatka *created_at* i *id* su generisani od strane *Firestore*-a prilikom upisa.

ZADATAK:

1. Logovati se na Firebase Consolu (korišćenjem google naloga) i kreirati projekat i bazu podataka prema uputstvu u nastavku. Dodati google-services.json fajl i ispratiti uputstvo do kraja kako bi aplikacija mogla da koristi *Firestore*.
2. Dodati funkciju u **StorageService** klasu za brisanje poi-a iz *Firestore* baze (delete).
3. Dovršiti implementaciju **PoiViewModel** klase dodavanjem funkcije za brisanje Poi-a (deletePoi) koja zove prethodno implementiranu funkciju StorageService klase. Obratiti pažnju na adekvatan poziv asinhronne funkcije.
4. Na ekranu *AddPoiScreen* dodati poziv funkcije za dodavanje novog poi-a.

Povezivanje na Firebase Firestore

1. Logovati se na *Firebase* konzolu pomoću Google naloga.



2. Kreirati projekat

× Create a project (Step 1 of 3)

Let's start with a name for your project[®]

Project name

POIs

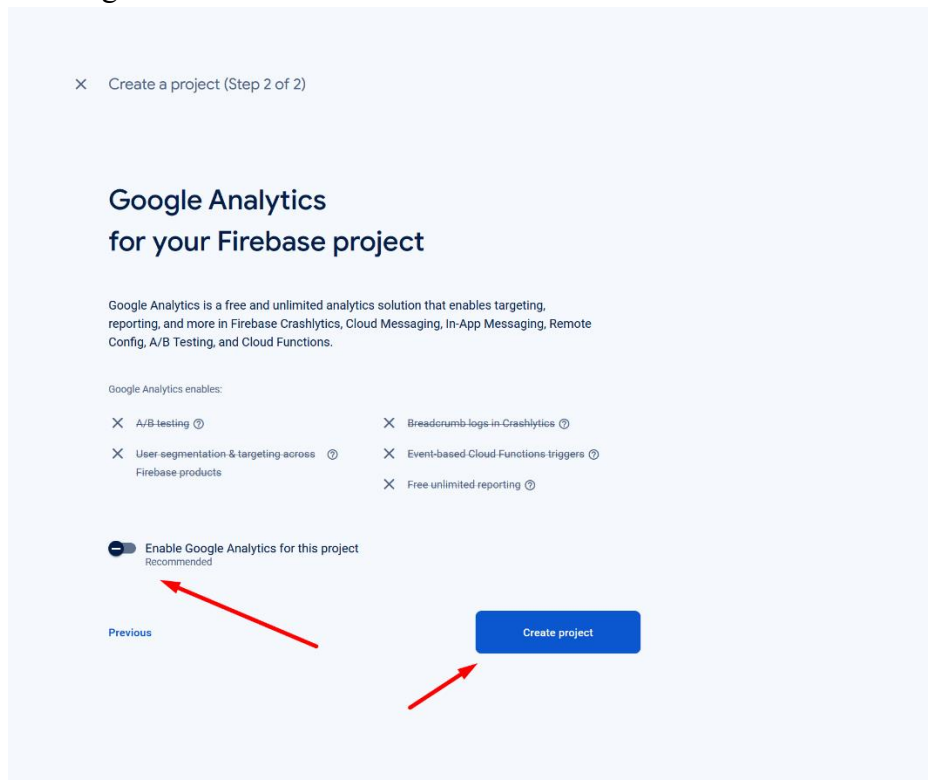
✎ pois-7cb66

We've updated our [terms of service](#)

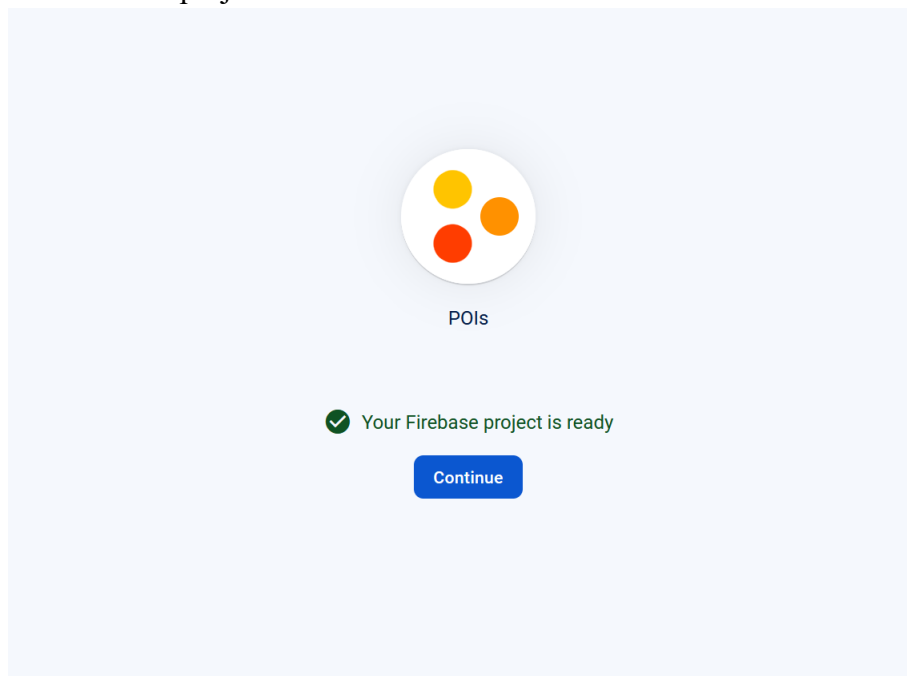
☒ I confirm that I will use Firebase exclusively for purposes relating to my trade, business, craft, or profession.

Continue

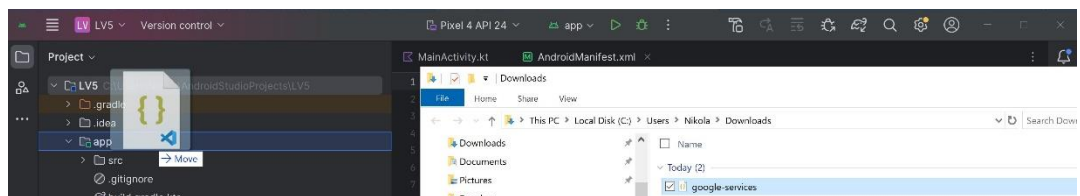
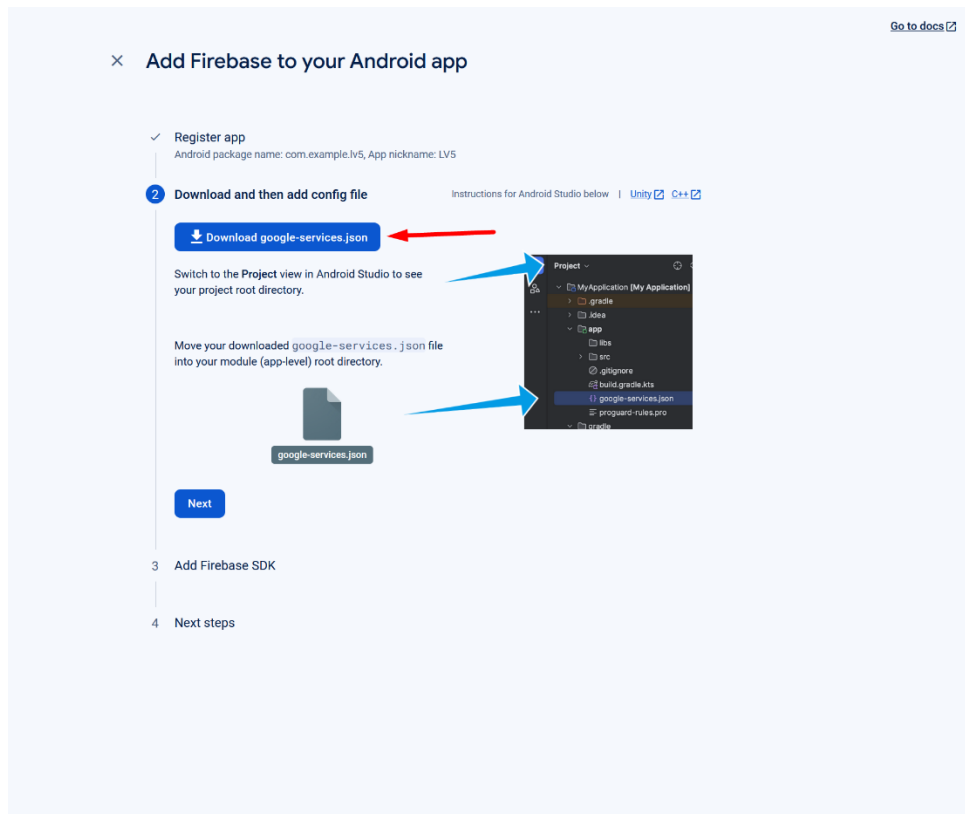
3. Onemogućiti analitiku i nastaviti



4. Sačekati dok projekat ne bude kreiran a zatim nastaviti.



5. Kliknuti na Android ikonicu kako bi Firebase bio dodat u Android projekat



8. Dodati odgovarajuće slogove u *Gradle* fajlove kako bi SDK bio adekvatno uključen. U slučaju ove vežbe proveriti da li je sve adekvatno uključeno.

×

Add Firebase to your Android app

Go to docs

✓ Register app

Android package name: com.example.lv5, App nickname: LV5

✓ Download and then add config file

3 Add Firebase SDK

Instructions for Gradle | Unity | C++

★ Are you still using the `buildscript` syntax to manage plugins? Learn how to [add Firebase plugins](#) using that syntax.

1. To make the `google-services.json` config values accessible to Firebase SDKs, you need the Google services Gradle plugin.

☒ Kotlin DSL (build.gradle.kts) ☐ Groovy (build.gradle)

Add the plugin as a dependency to your **project-level** `build.gradle.kts` file:

Root-level (project-level) Gradle file (<project>/build.gradle.kts):

```
plugins {  
    // ...  
  
    // Add the dependency for the Google services Gradle plugin  
    id("com.google.gms.google-services") version "4.4.1" apply false  
}
```

2. Then, in your **module (app-level)** `build.gradle.kts` file, add both the `google-services` plugin and any Firebase SDKs that you want to use in your app:

Module (app-level) Gradle file (<project>/<app-module>/build.gradle.kts):

```
plugins {  
    id("com.android.application")  
    // Add the Google services Gradle plugin  
    id("com.google.gms.google-services")  
    ...  
}  
  
dependencies {  
    // Import the Firebase BoM  
    implementation(platform("com.google.firebase:firebase-bom:33.0.0"))  
  
    // TODO: Add the dependencies for Firebase products you want to use  
    // When using the BoM, don't specify versions in Firebase dependencies  
    // https://firebase.google.com/docs/android/setup#available-libraries  
}
```

By using the Firebase Android BoM, your app will always use compatible Firebase library versions. [Learn more](#)

3. After adding the plugin and the desired SDKs, sync your Android project with Gradle files.

Previous

Next

LV LV5 Version control

Pixel 4 API 24 app

MainActivity.kt AndroidManifest.xml build.gradle.kts (-app) build.gradle.kts (LV5)

Project

LV5 C:\Users\Nikola\AndroidStudioProjects\LV5

gradle

idea

app

src

gitignore

build.gradle.kts

google-services.json

proguard-rules.pro

gradle

gitignore

build.gradle.kts

gradle.properties

gradlew

gradlew.bat

local.properties

settings.gradle.kts

External Libraries

Scratches and Consoles

1 // Top-level build file where you can add configuration options common to all sub-projects/modules.

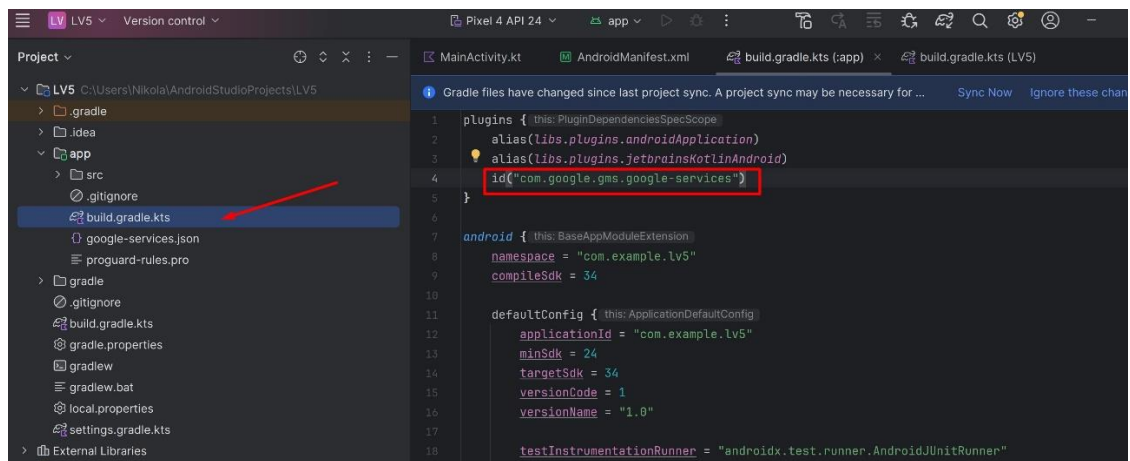
2 plugins { this: PluginDependenciesSpecScope

3 alias(libs.plugins.androidApplication) apply false

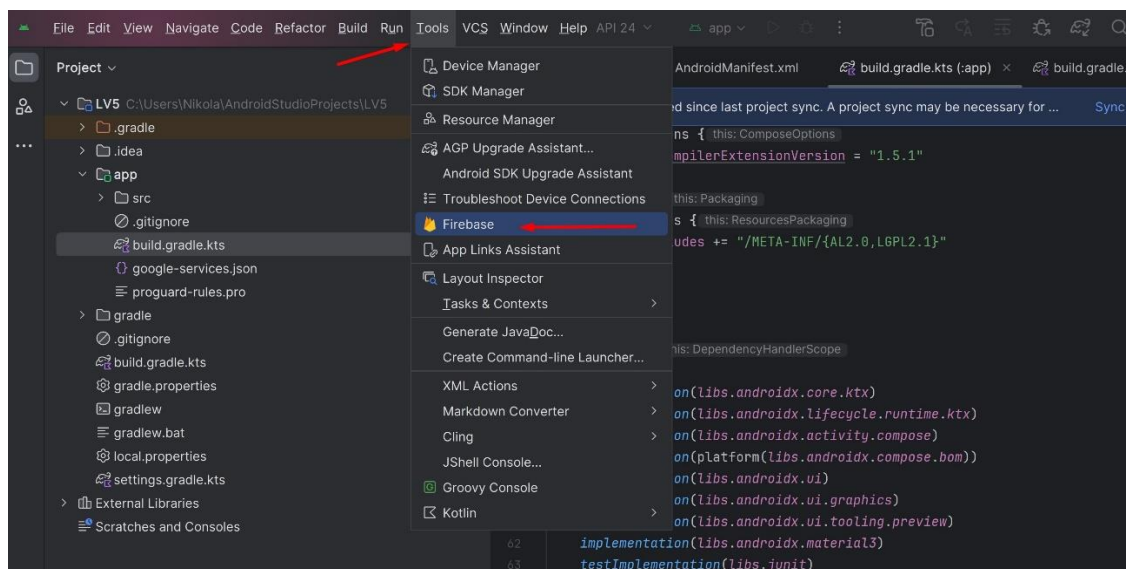
4 alias(libs.plugins.jetbrainsKotlinAndroid) apply false

5 id("com.google.gms.google-services") version "4.4.1" apply false

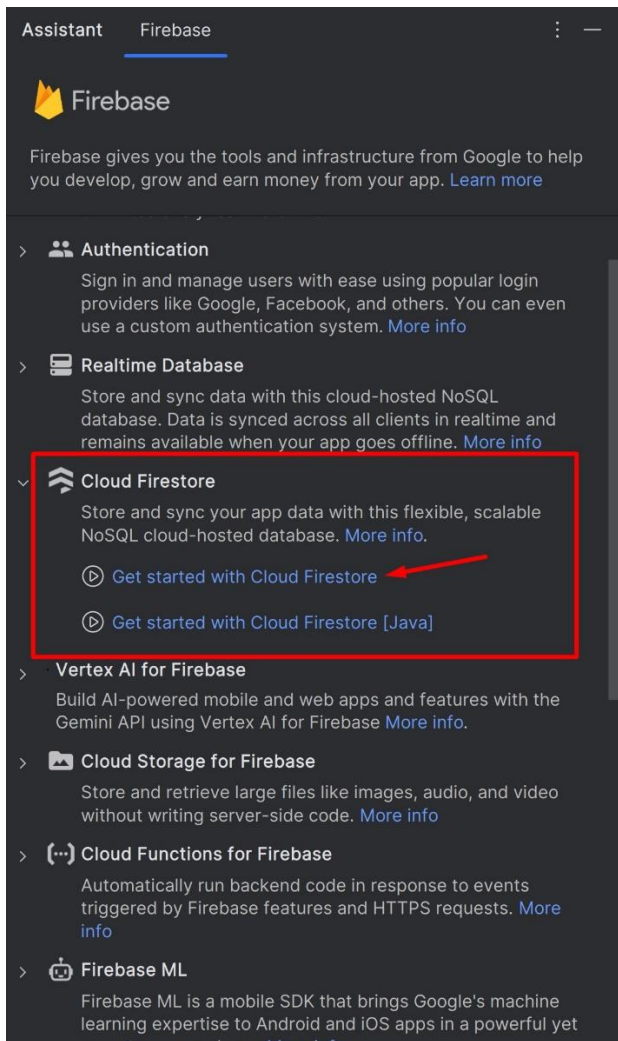
6 }



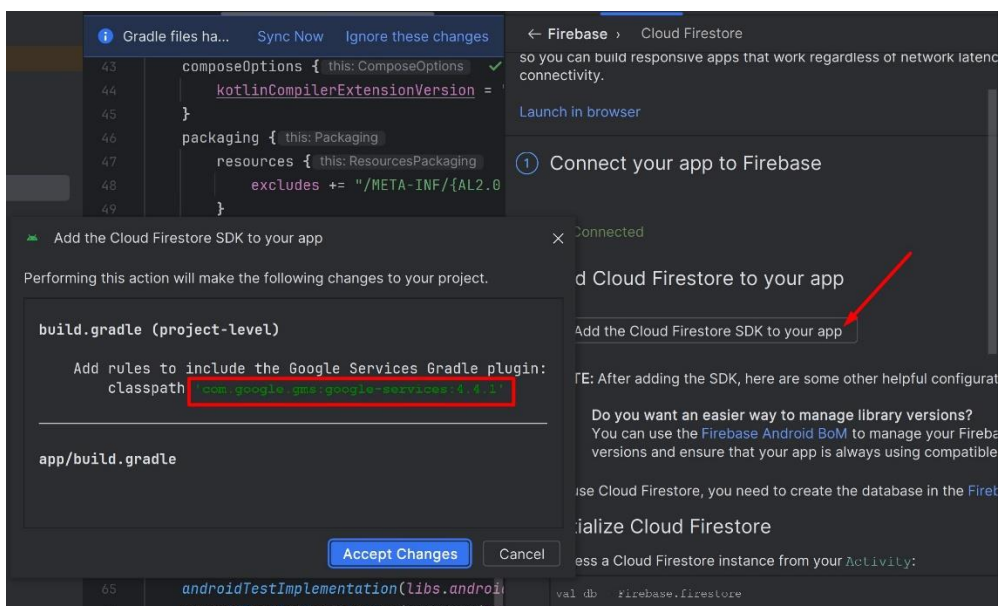
9. Otvoriti *Firebase* alat u *Android Studio*-u i proveriti da li je sve adekvatno povezano



10. I pokrenuti integraciju sa *Firestore*-om.



11. Pratiti upustva *Firebase* asistenta



12. Dodati *Firestore* biblioteku

The image shows an IDE window with a list of dependencies for an Android project. The line `implementation(libs.firebase.firestore)` is highlighted with a red box. To the right, a sidebar displays a four-step guide for adding and using Cloud Firestore:

- 2 Add Cloud Firestore to your app**
Add the Cloud Firestore SDK to your project.
NOTE: After adding the SDK, here are some things you need to do:
 - Do you want an easier way to manage your data? You can use the [Firebase Android Extensions](#) versions and ensure that your app is up to date.To use Cloud Firestore, you need to create a new collection and a document.
- 3 Initialize Cloud Firestore**
Access a Cloud Firestore instance from your app.

```
val db = Firebase.firestore
```
- 4 Add data**
Cloud Firestore stores data in *documents*. Firestore creates collections and documents. You do not need to explicitly create a new collection and a document.

13. Nastaviti na *Firebase* konzolu.


The image shows the 'Add Firebase to your Android app' wizard. It has a progress bar with four steps: 'Register app', 'Download and then add config file', 'Add Firebase SDK', and 'Next steps'. The 'Next steps' step is currently selected and numbered 4. Below the progress bar, the text says 'You're all set!' and provides links to documentation and sample apps. At the bottom, there is a 'Previous' link and a 'Continue to console' button.

14. Pokrenuti Cloud Firebase API

Google Cloud

POIs

Product details



Google Cloud Firestore API

Google Enterprise API

Easily store and sync app data at global scale

TRY THIS API

OVERVIEW

DOCUMENTATION

RELATED PRODUCTS

Overview

Cloud Firestore is a NoSQL document database that simplifies storing, syncing, and querying data for your mobile and web apps at global scale. Its client libraries provide live synchronization and offline support, while its security features and integrations with the Firebase and Google Cloud platforms accelerate building truly serverless apps.

Additional details

Type: SaaS & APIs

Last product update: 4/30/22

15. Izabрати Native mode

Google Cloud

POIs

Search (/) for resources, docs, products, and more

Search

Create database

1 Select your Firestore mode

Select between Native or Datastore mode

2 Configure your database

Select your Firestore mode

You can switch modes only if the database is empty.

☐ Native mode (recommended)

Recommended for all servers, mobile apps, and web apps

☒ Datastore mode

Use Datastore mode if your app requires the Datastore API

COMPARE MODES

CONTINUE

Pricing summary **FREE TIER**

This database has free-tier quota. Each project may have a single database named "(default)" which qualifies it for free-tier quota. Once you've exhausted this quota, you're billed based on operations, storage, and network usage¹. Location affects rates. [Learn more](#)

SHOW DETAILS

16. Potrebno je izmeniti prava pristupa iz Firebase konzole

Google Cloud

POIs

Search (/) for resources, docs, products, and more

Search

📦

1

?

:

N

Firestore

All databases > DATABASE (default)

Database

Firestore Studio

Indexes

Import/Export

Disaster Recovery

Time-to-live (TTL)

Security Rules

Insights

Usage

Key Visualizer

Security Rules

Last updated May 23, 2024 at 1:46:39 AM GMT+2

Security Rules help protect and secure your Firestore database by defining what mobile and web clients that connect to your database can access. [Learn more](#)

View your rules

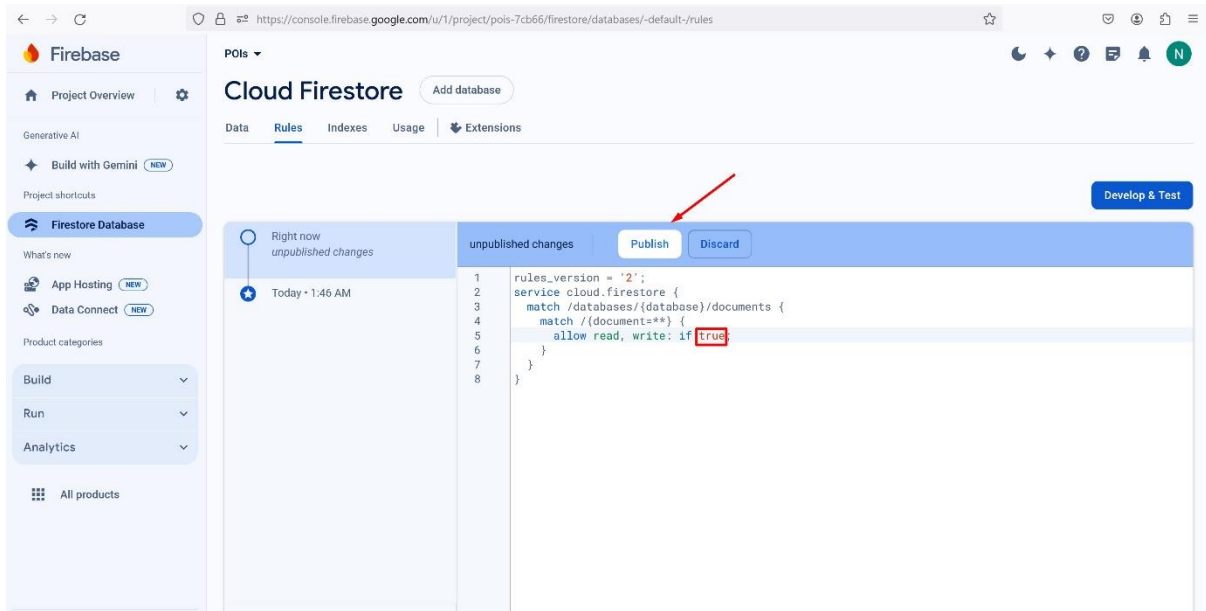
Rules only apply to mobile and web clients connecting to your database. Cloud IAM securely manages your internal server to database connections. [Learn more](#)

REFRESH

You can edit or write your own rules using the Firebase console or command-line interface. [Learn more](#)

```

1 rules_version = '2';
2 service cloud.firestore {
3   match /databases/{database}/documents {
4     match /{document=**} {
5       allow read, write: if false;
6     }
7   }
8 }
```



17. Počnite da koristite Firestore u svojoj aplikaciji.