

Uputstvo za rad s projektom
za
Nadzor vodovodne mreže

*Git Alliance
Zmaja od Bosne bb
Sarajevo
Bosna i Hercegovina*

Sarajevo, 30. April 2018

Sadržaj

Pregled promjena	1
1 Node i NPM	2
2 React Native	4
2.1 Expo	5
2.2 Editor	5
3 Backend	6
3.1 MongoDB	6
3.2 Pokretanje servera	9
3.3 Postman	9
4 API rute	10
4.1 /users	10
4.2 /users/id	10
4.3 /pipes	11
4.4 /pipes/id	11
4.5 /changes	12
4.6 /changes/id	12
4.7 Firebase	13

Pregled promjena

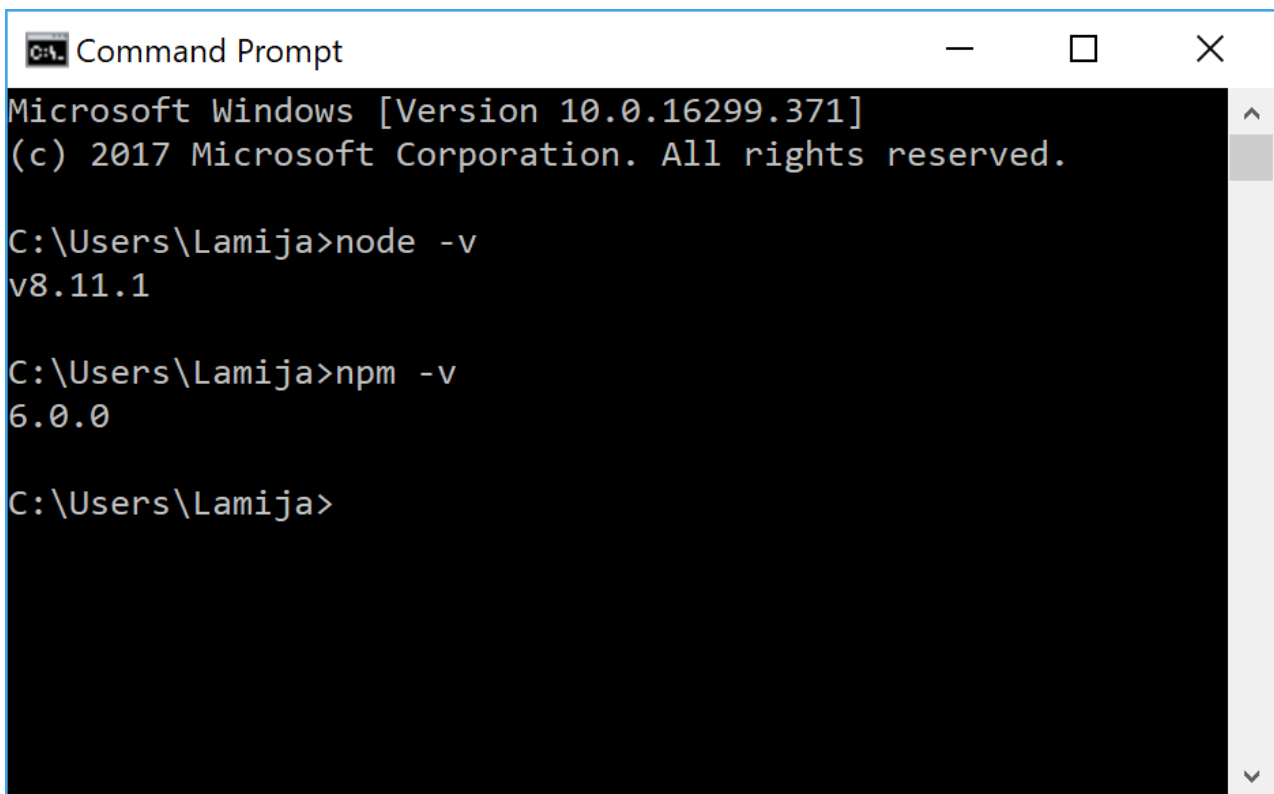
Verzija	Datum	Autor(ica)	Opis Promjena
1.0	30.04.2018	Git Alliance	Prva verzija dokumenta
2.0	14.05.2018	Git Alliance	MongoDB i backend

Poglavlje 1

Node i NPM

Za početak, potrebno je instalirati Node sa Node Package Managerom. Node je open source serversko okruženje, u čijem sklopu se nalazi i Node Package Manager - program koji pomaže pri instalaciji Node paketa. Node paket sadrži sve datoteke koje su potrebne za korištenje nekog Node modula. Moduli su JavaScript biblioteke koje se mogu koristiti u sklopu projekta.

Instalacija se može naći [ovdje](#). Nakon pokretanja instalacije nije potrebno ništa naročito mijenjati, dakle dovoljno je samo pratiti korake.

A screenshot of a Windows Command Prompt window. The title bar reads "Command Prompt". The window content shows the following text: "Microsoft Windows [Version 10.0.16299.371]" followed by "(c) 2017 Microsoft Corporation. All rights reserved." The user has entered the command "node -v" and the output is "v8.11.1". Then, the user has entered "npm -v" and the output is "6.0.0". The prompt is now "C:\Users\Lamija>".

```
C:\Users\Lamija>node -v
v8.11.1

C:\Users\Lamija>npm -v
6.0.0

C:\Users\Lamija>
```

Slika 1.1: Komande za provjeru verzije Node i NPM.

Nakon instalacije, Node i NPM se koriste kroz konzolu. Na Windows operativnom sistemu to je Command Prompt (cmd.exe), na Linux operativnom sistemu to je Terminal. Da bismo provjerili da li je instalacija bila uspješna, možemo unijeti komande sa slike 1.1. Iz razloga što nove verzije izlaze na gotovo dnevnoj bazi, nije praktično specificirati u dokumentu koju verziju treba koristiti. Dovoljno je reći da bi se trebalo raditi sa

najnovijom dostupnom LTS (Long Term Support) verzijom.

Razlog instalacije NPM jeste korištenje React Native paketa koji će biti opisan u idućem poglavlju. Paketi u sklopu NPM se mogu instalirati globalno ili lokalno, pomoću sljedeće komande:

```
npm install -g <ime paketa>
```

Ukoliko u komandu dodamo '-g', paket će se instalirati globalno, u folder pod nazivom node_modules koji se nalazi u home folderu korisnika. Na Windowsu se taj folder nalazi na putanji 'C:\\Users\\Imekorisnika\\node_modules'. Na Linuxu se taj folder nalazi na putanji '~/node_modules'. Ukoliko izostavimo '-g' iz komande, paket će se instalirati u folderu u kojem se trenutno nalazimo, što je uobičajena praksa pri korištenju većine paketa.

Poglavlje 2

React Native

React Native je okruženje za razvoj aplikacija za iOS i Android. Budući da je projekat već kreiran i preuzet sa repozitorija, dovoljno je instalirati potrebne node module. Dakle, trebamo otvoriti command prompt u folderu u kojem se nalazi projekat (u našem slučaju, naziv foldera je NadzorVodovodneMreze) i pokrenuti komandu:

```
npm install
```

Nakon instaliranih modula, možemo pokrenuti aplikaciju komandom:

```
npm start
```

Ovom komandom se pokreće server za development na računaru, te u konzoli vidimo QR kod koji nam služi za otvaranje aplikacije na mobilnom uređaju. Da bismo pokrenuli aplikaciju na mobilnom uređaju, potreban nam je dodatni softver koji je opisan u nastavku.



Slika 2.1: Primjer izlaza komande npm start za React Native aplikaciju.

2.1 Expo

Expo je aplikacija za pokretanje React Native aplikacija koje se serviraju preko lokalne mreže sa računara. Sastavni je dio tzv. 'Live development' sredine, što znači da se aplikacija ponovo učita svaki put kada dođe do promjene u kodu na računaru. Za Android okruženje, može se instalirati na [ovom linku](#), za iOS okruženje može se instalirati na [ovom linku](#).

Nakon instalacije Expo aplikacije, možemo povezati novokreiranu aplikaciju sa mobilnim uređajem. Načini povezivanja su navedeni u samom izlazu komande za pokretanje.

2.2 Editor

Za razvoj aplikacije može se koristiti bilo koji text editor (Visual Studio Code, Sublime Text, Atom, Vim itd.).

Poglavlje 3

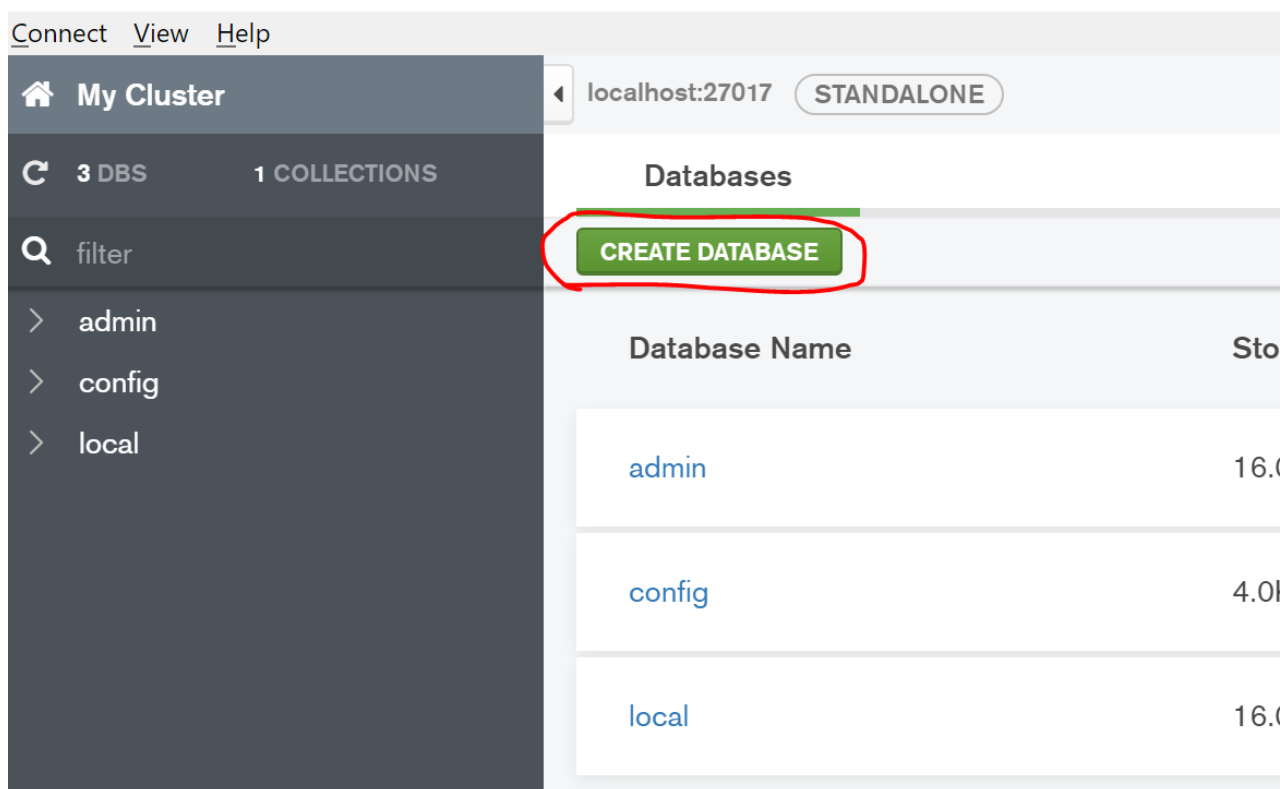
Backend

3.1 MongoDB

Potrebno je instalirati bazu MongoDB, koja se može preuzeti sa [ovog linka](#), verziju Community Server. Nakon uspješne instalacije, potrebno je kreirati direktorij u kojem će MongoDB spašavati podatke, pokretanjem command prompt-a, te unošenjem komande:

```
md \data\db
```

Sada možemo pokrenuti instaliranu bazu, navigirajući u folder C:\ProgramFiles\MongoDB\Server\3.6\bin i pokrenuti mongod.exe. Nakon ovoga, možemo pokrenuti MongoDB Compass koji je instaliran skupa sa MongoDB te se spojiti na lokalnu bazu koju smo pokrenuli u prethodnom koraku. Nakon spajanja na bazu, potrebno je kreirati konkretnu bazu u kojoj će se čuvati podaci aplikacije. To ćemo uraditi na sljedeći način: unutar MongoDB Compass, kliknemo na new database, kao na slici 3.1.



Slika 3.1: Kreiranje baze (korak 1)

Zatim je potrebno unijeti sljedeće podatke u formu:

Create Database

Database Name

Collection Name

☐ Capped Collection 

Before MongoDB can save your new database, a collection name must also be specified at the time of creation. [More Information](#)

CANCELCREATE DATABASE

Slika 3.2: Kreiranje baze (korak 2)

Ovime je baza postavljena, te ćemo ju populirati naknadno kroz nekoliko API poziva, nakon što pokrenemo server.

3.2 Pokretanje servera

Nakon uspješne instalacije i pokretanja MongoDB, možemo pokrenuti i backend projekat. Ukoliko je backend tek preuzet sa git repozitorija, ne sadrži u sebi node modules. Da bismo instalirali potrebne zavisnosti, moramo otvoriti command prompt u folderu u kojem se nalazi backend projekat (u našem slučaju Backend) i pokrenuti komandu:

```
npm install
```

u istom folderu u kojem se nalazi projekat backenda. Nakon ovoga, za pokretanje backenda dovoljno je unijeti komandu:

```
npm run start
```

nakon čega bi trebalo da vidimo izlaz "RESTful API server started on: 3000", što znači da API server sada sluša na portu 3000. Ovo je bitno jer React Native aplikacija treba komunicirati sa ovim API.

3.3 Postman

U svrhe testiranja API, kao i populiranja baze, potrebno je instalirati Postman, koji će nam omogućiti da šaljemo GET, POST, PUT i DELETE zahtjeve na backend. Instalaciju za Postman je moguće naći na [ovom linku](#).

Napomena: nakon kreiranja, baza je inicijalno prazna, pa je preporučljivo putem API poziva napuniti bazu nekim podacima.

Poglavlje 4

API rute

U nastavku će biti opisane rute za API pozive na serveru. Rute se nalaze na lokalnom serveru koji smo prethodno konfigurisali, dakle na <https://localhost:3000>.

4.1 /users

- **GET**: dobavljanje kolekcije svih korisnika
- **POST**: kreiranje jednog korisnika

Tijelo ovog zahtjeva je u JSON formatu, te sadrži sljedeća polja:

```
"username": "primjerKorisnika",  
"password": "primjerPasswordaKorisnika"
```

4.2 /users/id

- **GET**: dobavljanje korisnika sa naznačenim id
- **PUT**: promjena podataka korisnika sa naznačenim id

Tijelo ovog zahtjeva je u JSON formatu, te sadrži sljedeća polja:

```
"username": "primjerKorisnikaNovo",  
"password": "primjerPasswordaKorisnikaNovo"
```

- **DELETE**: brisanje korisnika sa naznačenim id

4.3 /pipes

- **GET**: dobavljanje kolekcije svih cijevi
- **POST**: kreiranje jedne cijevi

Tijelo ovog zahtjeva je u JSON formatu, te sadrži sljedeća polja:

```
"location": "lokacija"
"water_level": "vodostaj, broj između 0 i 1"
"critical_pipe_count": "pozitivan broj"
```

4.4 /pipes/id

- **GET**: dobavljanje cijevi sa naznačenim id
- **PUT**: promjena podataka o cijevi sa naznačenim id

Tijelo ovog zahtjeva je u JSON formatu, te sadrži sljedeća polja:

```
"location": "lokacija"
"water_level": "vodostaj, broj između 0 i 1"
"critical_pipe_count": "pozitivan broj"
```

- **DELETE**: brisanje cijevi sa naznačenim id

4.5 /changes

- **GET**: dobavljanje kolekcije svih promjena
- **POST**: kreiranje jedne promjene

Tijelo ovog zahtjeva je u JSON formatu, te sadrži sljedeća polja:

```
"author": "username korisnika koji je napravio izmjenu"
  "change_name": "naziv promjene"
  "change_date": "datum promjene"
  "location": "lokacija"
"water_level": "vodostaj, broj između 0 i 1"
"critical_pipe_count": "pozitivan broj"
```

4.6 /changes/id

- **GET**: dobavljanje promjene sa naznačenim id
- **PUT**: promjena podataka promjene sa naznačenim id

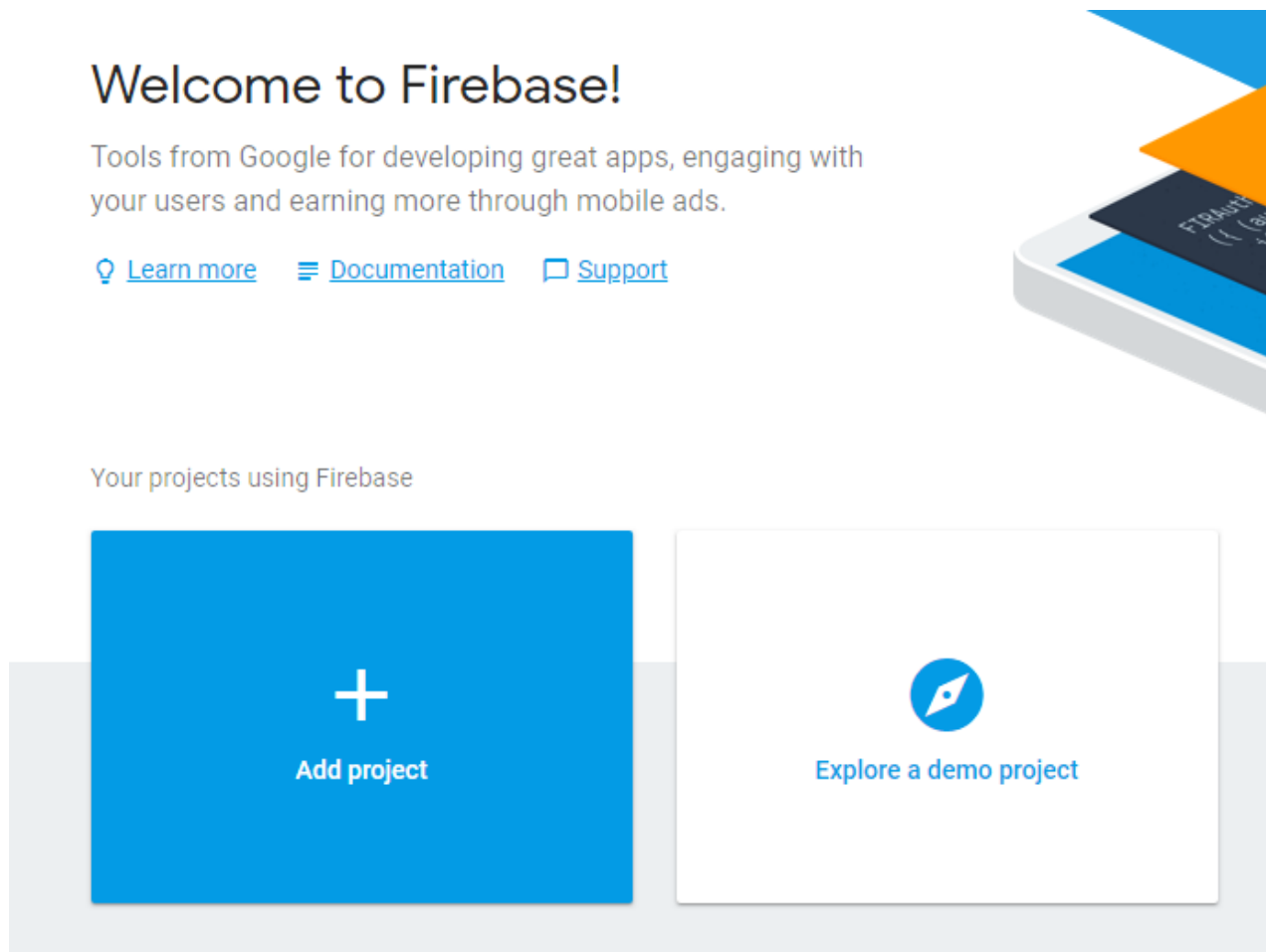
Tijelo ovog zahtjeva je u JSON formatu, te sadrži sljedeća polja:

```
"author": "username korisnika koji je napravio izmjenu"
  "change_name": "naziv promjene"
  "change_date": "datum promjene"
  "location": "lokacija"
"water_level": "vodostaj, broj između 0 i 1"
"critical_pipe_count": "pozitivan broj"
```

- **DELETE**: brisanje promjene sa naznačenim id

4.7 Firebase

Kako bi se mogle konfigurirati push notifikacije za korisnike, potrebno je prvo konfigurirati *Google Firebase*. Proces započinje ulaskom na sljedeći link: [Firebase Console](#), gdje je potrebno odabrati novi projekat.



Slika 4.1: Konfigurisanje Firebase-a (korak 1)

Nakon unošenja neophodnih podataka o projektu projekat se kreira.

Add a project

×

Project name

NadzorVodovodneMreze ▼

Project ID ⓘ

nadzorvodovodnemreze ✎

Country/region ⓘ

Bosnia & Herzegovina ▼

By default, your Analytics data will enhance other Firebase features and Google products. You can control how your analytics data is shared in your settings at any time. [Learn more](#)

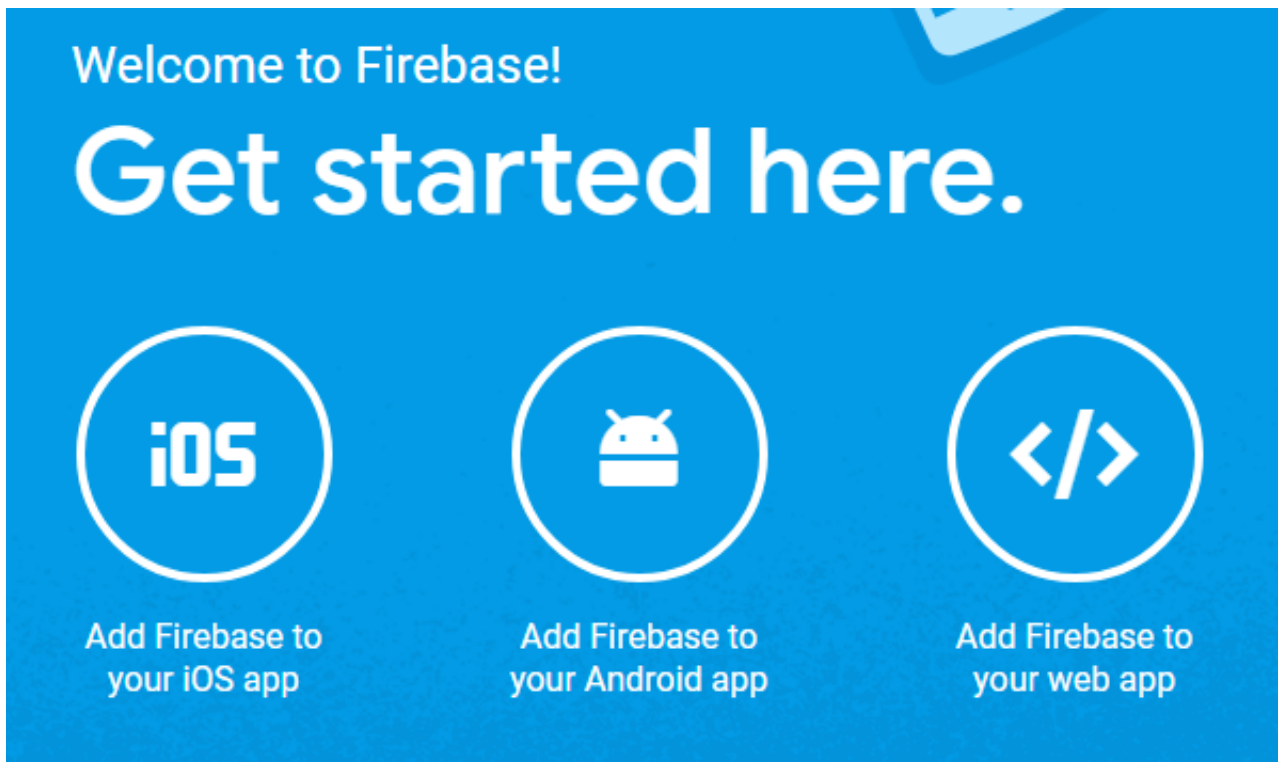
By proceeding and clicking the button below, you agree that you are using Firebase services in your app and agree to the applicable [terms](#).

CANCEL

CREATE PROJECT

Slika 4.2: Konfigurisanje Firebase-a (korak 2)

Potrebno je odabrati *Add Firebase to your Android app* kako bi se počela konfiguracija *remote push* notifikacija.



Slika 4.3: Konfigurisanje Firebase-a (korak 3)

Prvo je potrebno izvršiti specifikaciju imena paketa za slanje *push* notifikacija, na sljedeći način:

The screenshot shows the 'Add Firebase to your Android app' dialog box, which is a three-step process. Step 1, 'Register app', is currently active. The dialog has a blue header with the title and a close button. Below the header is a progress bar with three steps: 1 (Register app), 2 (Download config file), and 3 (Add Firebase SDK). The main area contains three input fields: 'Android package name' with the value 'com.nvm.android', 'App nickname (optional)' with the value 'Freemium Android App', and 'Debug signing certificate SHA-1 (optional)' with a placeholder value. At the bottom right are 'CANCEL' and 'REGISTER APP' buttons. The 'REGISTER APP' button is highlighted in blue. Below the buttons, it says 'in project NadzorVodovodneMr...'.

Add Firebase to your Android app ✕

1 ————— 2 ————— 3

Register app **Download config file** **Add Firebase SDK**

Android package name ⓘ

com.nvm.android

App nickname (optional) ⓘ

Freemium Android App

Debug signing certificate SHA-1 (optional) ⓘ

00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00

Required for Dynamic Links, Invites and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

CANCEL REGISTER APP

in project NadzorVodovodneMr...

Slika 4.4: Konfigurisanje Firebase-a (korak 4)

Nakon izvršene registracije aplikacije, potrebno je downloadovati *google-services.json* datoteku te je smjestiti u *android* direktorij aplikacije, prema sljedećem uputstvu:

Add Firebase to your Android app

1

2


3

Register appDownload config fileAdd Firebase SDK

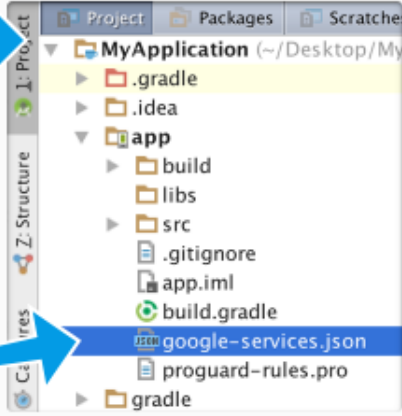
Android Studio instructions

Alternatives: [Unity](#) [C++](#)

- [Download google-services.json](#)
- Switch to the **Project** view in Android Studio to see your project root directory.
- Move the *google-services.json* file you have just downloaded into your Android app module root directory.



google-services.json



Slika 4.5: Konfigurisanje Firebase-a (korak 5)

Potrebno je izvršiti modifikaciju određenih datoteka u *android* folderu aplikacije, na sljedeći način:

Add Firebase to your Android app

1

2

3

Register appDownload config fileAdd Firebase SDK

Gradle instructions

Alternatives: [Unity](#) [C++](#)

The Google services plugin for [Gradle](#) loads the `google-services.json` file that you just downloaded. Modify your `build.gradle` files to use the plugin.

1. Project-level `build.gradle` (`<project>/build.gradle`):

```
buildscript {
    dependencies {
        // Add this line
        classpath 'com.google.gms:google-services:3.2.0'
    }
}
```
2. App-level `build.gradle` (`<project>/<app-module>/build.gradle`):

```
dependencies {
    // Add this line
    compile 'com.google.firebase:firebase-core:12.0.1'
}
...
// Add to the bottom of the file
apply plugin: 'com.google.gms.google-services'
```

includes Analytics by default ⓘ
3. Finally, press "Sync now" in the bar that appears in the IDE:

Gradle files have changed since last sync

[Sync now](#)

FINISH

Slika 4.6: Konfigurisanje Firebase-a (korak 6)

Nakon toga neophodno je izvršiti sljedeću komandu:

```
npm install react-native-fcm --save
```

Slika 4.7: Konfigurisanje Firebase-a (korak 7)

Zatim je potrebno dodati liniju `google()` u `build.gradle` datoteku projekta:

```
allprojects {  
    repositories {  
        mavenLocal()  
        jcenter()  
        google()  
        maven {  
            // All of React Native (JS, Obj-C sources, Android binaries) is  
            url "$rootDir/../node_modules/react-native/android"  
        }  
    }  
}
```

Slika 4.8: Konfigurisanje Firebase-a (korak 8)

Ovime je završena konfiguracija *Firebase* za željenu aplikaciju.