

Android

-

Firestore



Firebase

Peter Borovanský

KAI, I-18

MS-Teams: [2sf3ph4](#), [List](#), [github](#)

borovan 'at' ii.fmph.uniba.sk

•Room

- @Database
- @Entity
- @Dao

•Firebase

- Authentication
 - Email/Password, Google, FB, Twitter
- Realdatabase
- Storage
- Push notifications

Kam (inam) uložiť naše dáta



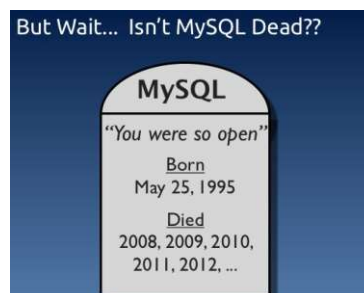
W - Windows
A - Apache
M - MySQL
P - PHP
WAMP - Server

- máme vlastný server – protokol najčastejšie http-https
 - najčastejšie (v bakalárkach) AMP – Apache-MYSQL-PHP, prip. java-servlets
 - tak trochu doba minulá...

[LinuxDays 2017:](#)

[MySQL sežere Vaše data](#)

[David Karban](#)



bude:

- lokálne
 - SQLite <https://developer.android.com/training/data-storage/sqlite>
 - Room <https://developer.android.com/topic/libraries/architecture/room>
- cloudové úložiská a ich služby
 - Relačné - tabuľkovo orientované
 - [Parse.com](#) <http://parseplatform.org/> (kúpil Facebook)
 - Nerelačné - noSQL – json
 - [firebase.com](#) (Google)





SQLite vs. Room

SQLite databáza:

- obsahovala veľa *boilerplate* kódu
- operácie, ktoré blokovali main-thread
- sql dotazy, ktoré sa konštruujú v run-time (nie sú predkompilované)

! **Caution:** Although these APIs are powerful, they are fairly low-level and require a great deal of time and effort to use:

- There is no compile-time verification of raw SQL queries. As your data graph changes, you need to update the affected SQL queries manually. This process can be time consuming and error prone.
- You need to use lots of boilerplate code to convert between SQL queries and data objects.

For these reasons, we **highly recommended** using the [Room Persistence Library](#) as an abstraction layer for accessing information in your app's SQLite databases.

od 2017 Android má Room, ktorá umožňuje:

- používať komponenty Room (kapt - anotácie @Entity, @Dao, @Database)
- udržiavať relácie medzi entitami (kľúče - keys)
- spúšťať DB operácie mimo main-thread, prirodzene pomocou corutín
- je to odporúčaná náhrada za bývalú SQLite (existuje v Androide od API-1)



Room

```
build.gradle (project)
plugins {
    kotlin("jvm") version "2.0.21"
    alias(libs.plugins.android.application) apply false
    alias(libs.plugins.kotlin.android) apply false
    id("com.google.devtools.ksp") version "2.0.21-1.0.27" apply false
}
```

```
build.gradle (module:app)
dependencies {
    val room_version = "2.6.1"
    ksp("androidx.room:room-compiler:$room_version")
    implementation("androidx.room:room-ktx:$room_version")
}
```

```
build.gradle (module:app)
plugins {
    alias(libs.plugins.android.application)
    alias(libs.plugins.kotlin.android)
    id("com.google.devtools.ksp")
}
```

Anotácie:

- **@Database** – abstraktná trieda RoomDatabase
 - Room.databaseBuilder() persistentná inštancia, dáta existujú aj po skončení procesu,
 - Room.inMemoryDatabaseBuilder() – dáta zmiznú, keď proces zanikne

- **@Entity** – tabuľky v SQL databáze

@Entity

```
data class Student ( ... )
```

- **@DAO** – data access object – metódy na prístup k databáze

@Dao

```
interface StudentDAO {
    @Query("SELECT * FROM Student WHERE name = :name")
    suspend fun getName(name: String): Student?
}
```



RoomDB

malý príklad

- vytvoríme aplikáciu na registrovanie študentov s funkciami:
 - signup/login/logout/delete
- v návrhovom vzore MVVM
- s použitím corutín

Najprv si:

- obohatíme build.gradle (module:app) o
- Room –používame plugin ksp (com.google.devtools.ksp)

```
build.gradle (module:app)
dependencies {
    val room_version = "2.6.1"
    ksp("androidx.room:room-compiler:$room_version")
    implementation("androidx.room:room-ktx:$room_version")
}
```

- plugins

```
plugins {
    alias(libs.plugins.android.application)
    alias(libs.plugins.kotlin.android)
    id("com.google.devtools.ksp")
}
```



@Entity

```
@Entity
data class Student (
    val isic          : String,
    val name          : String,
    val passwordHash  : Int,
    @ColumnInfo(passwordHash = "password_hash")
    val description   : String) {
    @PrimaryKey(autoGenerate = true)
    var id: Long = 0
}
```

mapovanie kotlinovskej položky `passwordHash` na stĺpec tabuľky `Student` s názvom `password_hash`

`autoGenerate` - autoincrement primary key – najčastejšie sa implementuje ako SEQUENCE (Oracle, Postgre), resp. MS-SQL

```
CREATE TABLE (
    ID_column INT NOT NULL IDENTITY(1,1) PRIMARY KEY
```

@Entity

foreignKeys/Embedded

```
@Entity
data class User(
    @PrimaryKey val userId: Long,
    val name: String,
    val age: Int
)
```

```
@Entity(foreignKeys = [
    ForeignKey(
        entity = User::class,
        parentColumns = ["userId"],
        childColumns = ["userOwnerId"],
        onDelete = CASCADE)
])
data class Library(
    @PrimaryKey
    val libraryId: Long,
    val title: String,
    val userOwnerId: Long
)
data class UserAndLibrary(
    @Embedded val user: User,
    @Relation(
        parentColumn = "userId",
        entityColumn = "userOwnerId"
    )
    val library: Library
)
```



@Dao

data access object

@Dao

```
interface StudentDAO {  
    @Insert(onConflict = OnConflictStrategy.REPLACE)  
    suspend fun insert(student: Student): Long  
  
    @Query("SELECT * FROM Student WHERE name = :name")  
    suspend fun getName(name: String): Student?  
  
    @Query("SELECT * FROM Student WHERE id = :id")  
    suspend fun getID(id: Long): Student?  
  
    @Query("SELECT * FROM Student WHERE isic = :isic")  
    suspend fun getISIC(isic: String): Student?  
  
    @Query("DELETE FROM Student WHERE id = :id")  
    suspend fun deleteID(id: Long)  
  
    @Insert  
    suspend fun insertAll(vararg students: Student)  
  
    @Delete  
    suspend fun delete(student: Student)  
}
```




@Database

```
@Database(entities = arrayOf(Student::class), version = 1)
abstract class StudentDatabaseEasy: RoomDatabase() {
    abstract fun studentDAO(): StudentDAO

    fun getInstance(context: Context) = Room.databaseBuilder(
        context.applicationContext,
        StudentDatabaseEasy::class.java,
        "studentdatabase"
    ).build()
}

signup:
coroutineScope.launch {
    if (db.getName(name) != null || db.getISIC(isic) != null) {
        withContext(Dispatchers.Main) {
            error.value = "Student already exists"
        }
    } else {
        val studentId = db.insert(
            Student(isic, name, password.hashCode(), desc))
    }
}
```



@Database

```
val coroutineScope = CoroutineScope(Dispatchers.IO)
```

```
fun login(name: String, password: String) {  
    coroutineScope.launch {  
        val student = db.getName(name)  
        if (student == null)  
            withContext(Dispatchers.Main) {  
                error.value = "Student not found"  
            }  
        else {  
            if (student.passwordHash == password.hashCode()) {  
                Status.login(student)  
                withContext(Dispatchers.Main) {  
                    logged.value = true  
                }  
            } else {  
                withContext(Dispatchers.Main) {  
                    error.value = "Password is incorrect"  
                }  
            }  
        }  
    }  
}
```

Rozsiahlejší príklad @Entity

```
@Entity(foreignKeys = [ForeignKey(
    entity = Company::class,
    parentColumns = ["id"],
    childColumns = ["company_id"],
    onDelete = ForeignKey.CASCADE)])
```

```
data class Employee (
    @ColumnInfo(name = "name")
    val name: String,

    @ColumnInfo(name = "company_id")
    val companyId : Int = 0)
{
    @PrimaryKey(autoGenerate = true)
    @ColumnInfo(name = "id")
    val employeeId : Int = 0
}
```

```
@Entity(tableName = "Department")
class Department(
    @ColumnInfo(name = "name")
    val name: String,

    @ColumnInfo(name = "company_id")
    val companyId : Int = 0
) {
    @PrimaryKey(autoGenerate = true)
    @ColumnInfo(name = "id")
    var id = 0
}
```

```
@Entity(tableName = "Company")
data class Company (
    @ColumnInfo(name = "name")
    val name: String,

    @ColumnInfo(name = "date_updated")
    @TypeConverters(DateConverter::class)
    val itemUpdatedDate: Date? = null,

    @Embedded
    private val location: Location? = null,

    @Embedded(prefix = "hq_")
    private val headLocation: Location? = null,

    @Ignore
    val picture: Bitmap? = null
) {
    @PrimaryKey
    @ColumnInfo(name = "id")
    val companyId = 0
}
```

```
@Entity(primaryKeys = ["id", "code"])
class Office {
    val id : Int = 0
    var code: String
}
```

Rozsiahlejší príklad @Dao

```
@Dao
interface EmployeeDao {
    @get:Query("SELECT * FROM Employee")
    val allEmployees: LiveData<List<Employee?>>?

    @RawQuery
    fun getAllEmployeesWithLimit(query: String?): List<Employee?>

    @Insert
    fun insertEmployee(employee: Employee?)
    @Insert(onConflict = OnConflictStrategy.REPLACE)
    fun insertAll(employees: List<Employee?>)

    @Update
    fun updateEmployee(employee: Employee?)

    @Delete
    fun deleteEmployee(employee: Employee?)
}
```

```
@Dao
abstract class DepartmentDao {
    @Insert(onConflict = OnConflictStrategy.REPLACE)
    abstract fun insertAll(departments: List<Department?>)

    @Insert
    abstract fun insert(product: Department?)

    @Delete
    abstract fun delete(product: Department?)

    @Transaction
    fun insertAndDeleteInTransaction(
        newDepartment: Department?,
        oldDepartment: Department?) {
        insert(newDepartment)
        delete(oldDepartment)
    }
}
```

```
@Dao
interface CompanyDao {
    @get:Query("SELECT * FROM Company")
    val allCompanies: LiveData<List<Company?>>?

    @get:Query("SELECT * FROM Company ORDER BY name")
    val allCompaniesOrdered: LiveData<List<Company?>>?

    @Insert
    fun insertCompany(company: Company?)

    @Query(
        "SELECT * FROM Company WHERE name LIKE :companyName")
    fun getCompanies(
        companyName: String?):
        LiveData<List<Company?>>?

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    fun insertAll(companies: List<Company?>)

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    fun insertAll(vararg companies: Company?)

    @Update
    fun updateCompany(company: Company?)

    @Update
    fun updateCompanies(vararg company: Company?)

    @Delete
    fun deleteCompany(company: Company?)

    @Delete
    fun deleteCompanies(vararg company: Company?)
}
```



Ďalšie čítanie - Room

tutoriály - odporúčané

- **Migrate from kapt to KSP**
<https://developer.android.com/build/migrate-to-ksp>
- **Room Persistence Library: Introduction:**
<https://medium.com/@magdamiu/android-room-persistence-library-97ad0d25668e>
- **Room Persistence Library: Entity, Dao, Database:**
<https://medium.com/@magdamiu/android-room-components-5a7458b99191>
- **Room Persistence Library: Relations:**
<https://medium.com/@magdamiu/android-room-persistence-library-relations-75bbe02e8522>
- **Room Persistence Library: Queries and Migration Support:**
<https://magdamiu.medium.com/android-room-persistence-library-queries-and-migration-support-a9f21d2dc9d8>

Cvičenie

úvod si pozrite v cvičení

Vytvorte aplikáciu, ktorá slúži na **evidenciu známok študentov** s nasledujúcimi entitami (verím, že tomu zápisu rozumiete). Máte urobiť Room model a minimálne základne GUI, aby bolo jasné, že viete pracovať s Room.

Základná verzia na hodnotenie:

- vie pridať položku do troch tabuliek (delete nemusíte riešiť), Znamky sú statický číselník,
- zobrazuje počet študentov, predmetov a hodnotení.

Bonus: [1 bod] nejaký listview zobrazuje všetky hodnotenia, v ľub. poradí, bez filtrov,

Bonus: [1 bod] viete zmazať študenta/predmet, ktorý už má hodnotenie, CASCADE...

```
drop table Student;
create table Student (
    id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    isic varchar(20) not null,
    meno varchar(20),
    priezvisko varchar(30),
    CONSTRAINT sid_pk PRIMARY KEY (id),
    CONSTRAINT isic_pk UNIQUE (isic)
);
insert into Student (isic, meno, priezvisko)
    values ('123456789', 'Sansa', 'Starkova');
commit;
drop table predmet;
create table Predmet (
    id NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    kod varchar(20) not null,
    nazov varchar(80),
    CONSTRAINT pid_pk PRIMARY KEY (ID),
    CONSTRAINT kod_pk UNIQUE (kod)
);
insert into Predmet (kod, nazov)
    values ('1-AIN-472/12',
        'Vývoj mobilných aplikácií, zimný semester 2020/2021');
commit;
```

```
create table Znamka (
    ID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    text varchar(5),
    CONSTRAINT zid_pk PRIMARY KEY (ID),
    CONSTRAINT text_pk UNIQUE (text)
);
insert into Znamka (text) values ('A');
insert into Znamka (text) values ('B');
insert into Znamka (text) values ('C');
insert into Znamka (text) values ('D');
insert into Znamka (text) values ('E');
insert into Znamka (text) values ('Fx');
create table Hodnotenie (
    studentID NUMBER,
    predmetID NUMBER,
    znamkaID NUMBER,
    CONSTRAINT fk_student FOREIGN KEY (studentID)
        REFERENCES Student(ID),
    CONSTRAINT fk_predmet FOREIGN KEY (predmetID)
        REFERENCES Predmet(ID),
    CONSTRAINT fk_znamka FOREIGN KEY (znamkaID)
        REFERENCES Znamka(ID)
);
```

Cvičenie - C

Malá evidencia produktov pomocou bar-code scannera

EAN 8584004040108

Horalka

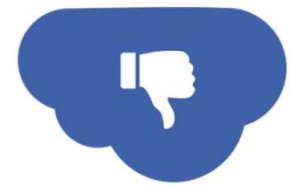


[See on Ebay](#)

Brand	Sedita
Manufacturer	I.D.C. Holding
EAN	8584004040108
Country	Slovakia
Last Scan	Nov 25 2020 at 11:17 PM
GS1 Name	I.D.C. HOLDING, odš.závod Pečivárne
GS1 Address	Drieňová 3 Bratislava SK
Description	No description for 8584004040108
Barcode	



8 584004 040108



Parse + facebook

Parse vs. Parse Server

- API pre komunikáciu mobilných/web aplikácií, ukladanie/zdieľanie dát
- podporoval PUSH notifikácie pomocou Google Cloud Messaging (GCM) vaša aplikácia dostane notifikáciu zo servera, ak iný užívateľ vyvolá event



[Docs](#) [Blog](#) [Community](#) [Back Us](#)

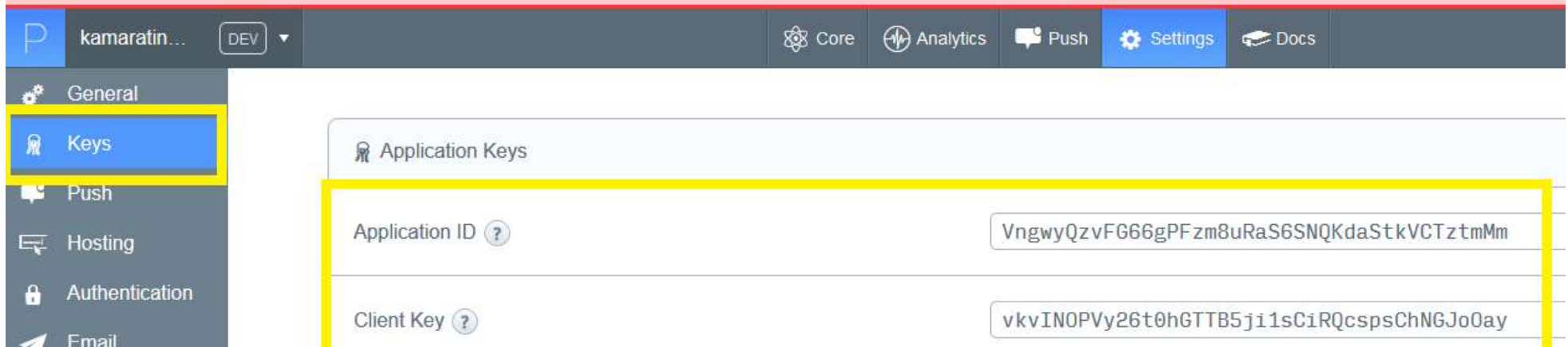
Parse.com kúpený FB 2013 (free) končí v 2017 ☹

The Complete Application Stack

Ponúka migráciu na open-source Parse Server

- s veľmi podobným API 😊           
- na vlastnom serveri, s infraštruktúrou Node.js + MongoDB + Python

The Parse hosted service will be retired on January 28, 2017. You need to migrate your data as soon as possible. Read more on our [blog](#)



The screenshot shows the Parse.com dashboard interface. At the top, there's a navigation bar with the Parse logo, a user profile 'kamaratin...', a 'DEV' dropdown, and icons for 'Core', 'Analytics', 'Push', 'Settings', and 'Docs'. On the left, a sidebar menu lists 'General', 'Keys', 'Push', 'Hosting', 'Authentication', and 'Email'. The 'Keys' section is selected and highlighted with a yellow box. The main content area, titled 'Application Keys', contains two fields: 'Application ID' with the value 'VngwyQzvFG66gPFzm8uRaS6SNQKdaStkVCTztmMm' and 'Client Key' with the value 'vkvINOPVy26t0hGTTB5j11sCiRQcspChNGJo0ay'. Both fields are highlighted with yellow boxes.

Alternatívy

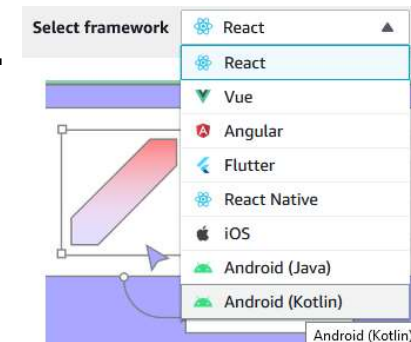
(k Parse Server)

Veci zadarmo sú (často) síce najlepšie, ale zase pomíjivé
Treba pozrieť konkurenciu:

- Amazon AWS – iOS, Android, Web, ReactNative, ...

<https://aws.amazon.com/amplify/>

- Authentication
- File Storage
- Analytics
- Notifications



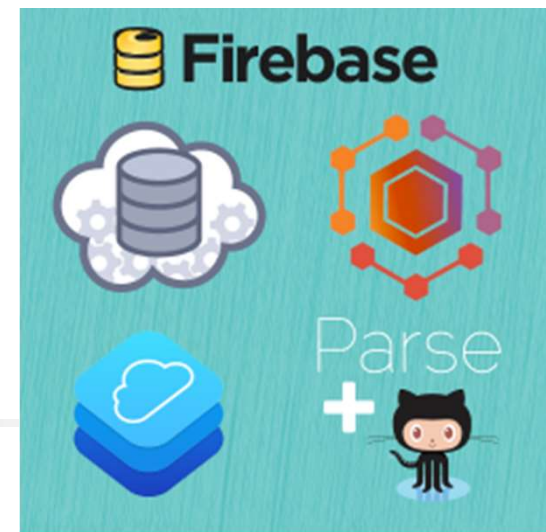
- MS Azure Mobile App SDK – iOS, Android, Xamarin, Cordova, ...

<https://docs.microsoft.com/en-us/azure/app-service-mobile/app-service-mobile-android-how-to-use-client-library>

- Databases
 - Azure SQL
 - Azure CosmosDB
 - Azure PostgreSQL
 - Azure for MariaDB
- Analytics

- Firestore – iOS, Android, Flutter, Web, ...

- realtime JSON oriented DB,
- PUSH notifikácie, analytics, REST API



Firebase

Startup Firebase(2011) kúpil Google(2014)



Ponúka služby/servisy:

- Authentication – FB/Google/Twitter/FirebaseEmail
- Realtime database – non sql store & synchronized database
- Cloud Firestore – realtime scalable database, works off-line
- File storage – secure upload & download files
- Test lab for Android – rôzne virtuálne zariadenia
- Firebase local emulator – WebApp/Android/iOS
- Cloud messaging – push notifikácie pomocou

Google Cloud Messaging/Firebase Cloud Messaging

- Cloud Functions – runs backend scripts on Google servers
- Crash reporting
- Analytics

...

<https://firebase.google.com/docs?authuser=0>



VMADemo

Analytics

DEVELOP

Authentication

Database

Storage

Hosting

Test Lab

Crash Reporting

GROW

Notifications

Remote Config

Dynamic Links

EARN

AdMob

Firebase Products

and development platforms

Build your app



Cloud Firestore

iOS



Firebase ML

iOS



Cloud Functions

iOS C++



Authentication

iOS C++



Hosting



Cloud Storage

iOS C++



Realtime Database

iOS C++

Improve app quality



Crashlytics

iOS



App Distribution

iOS



Performance Monitoring

iOS



Test Lab

iOS

Grow your business



Analytics

iOS C++



Extensions

iOS C++



Predictions

iOS C++



Firebase A/B Testing

iOS C++



Cloud Messaging

iOS C++



In-App Messaging

iOS



Remote Config

iOS C++



Dynamic Links

iOS C++



App Indexing

iOS

Firebase Console

prvý dotyk

× Create a project (Step 1 of 3)

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

Project name

The screenshot displays the Firebase Console interface. On the left is a dark sidebar with the 'Firebase' logo and a 'Project Overview' section. Below this, under 'Product categories', is a 'Build' section with a list of services: Authentication, App Check, Firestore Database, Realtime Database, Extensions, Storage, Hosting, Functions, Machine Learning, and Remote Config. The main content area has a blue header with 'testProject' and a dropdown arrow. Below the header is a white banner with an envelope icon and the text 'Receive email updates about new Firebase features, research, and events'. The main body has a large 'testProject' title with a 'Spark plan' button next to it. Below this is the heading 'Get started by adding Firebase to your app' and a row of icons for iOS, Android, and a code editor icon, followed by two circular icons with arrows. At the bottom of the main area is the text 'Add an app to get started'. The footer contains several links and a 'Create project' button.

testProject

Receive email updates about new Firebase features, research, and events

testProject Spark plan

Get started by adding
Firebase to your app

iOS+ Android </> ↻ ↺

Add an app to get started

User segmentation & targeting across
Firebase products

Event-based Cloud Functions triggers

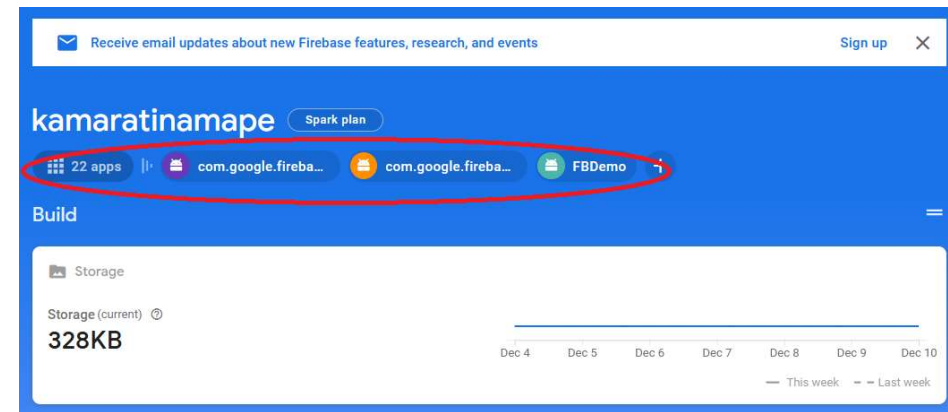
Free unlimited reporting

service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. [Learn more](#)

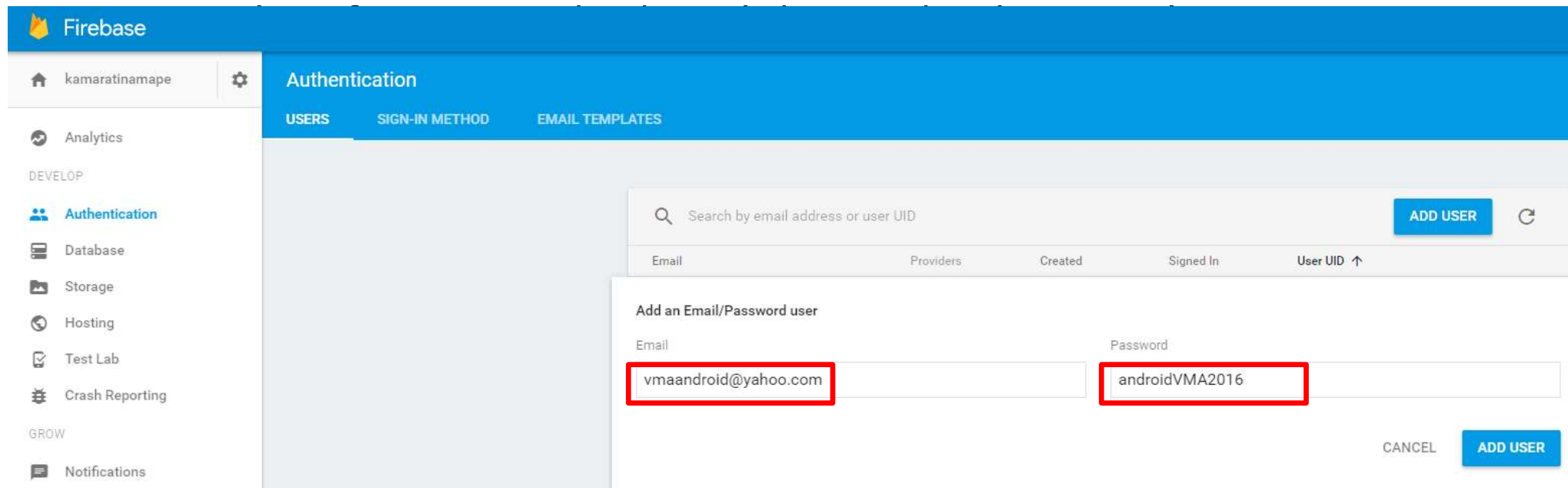
Previous Create project

Firebase Console

v tejto prezentácii



<https://console.firebase.google.com/project/kamaratinamape/database/data>



- ukážky sú robené pomocou tohoto účtu
- login a password vidíte v obrázku
- ale obávam sa, že to na inom zariadení bude chcieť potvrdenie tel.čísla ☹

<https://console.firebase.google.com/project/kamaratinamape/database/data>

Použite iný/váš účet
vmaandroid@yahoo.com
androidVMA2016



Autentifikácia

Sign-in metódy:

- vlastná autentifikácia cez email/password
 - môžete definovať viacero email-účtov (signup/signin)
- cez FB/Google/Twitter/GitHub/...

testProject ▼

Authentication

Authenticate and manage users from a variety of providers without server-side code

Get started

testProject ▼

Authentication

Users Sign-in method Templates Usage Settings

Sign-in providers

Get started with Firebase Auth by adding your first sign-in method

Native providers

Email/Password

Phone

Anonymous

Additional providers

Google

Game Center

Microsoft

Facebook

Apple

Twitter

Play Games

GitHub

Yahoo

Custom providers

OpenID Connect

SAML



SMS Multi-factor Authentication

Allow your users to add an extra layer of security to their account. two steps, using SMS. [Learn more](#)



Sign-in methods

<https://console.firebase.google.com/u/1/project/kamaratinamape/authentication/providers>

kamaratinamape ▾

[Go to docs](#)

Authentication

[Users](#) [Sign-in method](#) [Templates](#) [Usage](#) [Settings](#)

Sign-in providers

[Add new provider](#)

Provider	Status
Email/Password	Enabled
Google	Enabled
Facebook	Enabled
Twitter	Enabled
Anonymous	Enabled



Sign-in methods

<https://console.firebase.google.com/u/1/project/kamaratinamape/authentication/providers>

Configure provider (Step 2 of 2)

Facebook ☐ Enable

App ID

App secret

To complete set up, add this OAuth redirect URI to your Facebook app configuration. [Learn more](#)

`https://testproject-19224.firebaseio.com/_/auth/handler`

Cancel Save

Configure provider (Step 2 of 2)

Twitter ☐ Enable

API Key

API secret

To complete set up, add this callback URL to your Twitter app configuration. [Learn more](#)

`https://testproject-19224.firebaseio.com/_/auth/handler`

Email/Password ☒ Enable

Allow users to sign up using their email address and password. Our SDKs also provide email address verification, password recovery, and email address change primitives. [Learn more](#)

Firestore requires the passwords to be at least 6 characters

☐ Email link (passwordless sign-in) ☐ Enable

Google ☒ Enable

Google sign-in is automatically configured on your connected Apple and web apps. To set up Google sign-in for your Android apps, you need to add the [SHA1 fingerprint](#) for each app on your [Project Settings](#).

Update the [project-level setting](#) below to continue

Project public-facing name ?
project-451754775840

Project support email ?
prostrediahm@gmail.com

This will be the email address presented to users when they are authenticating with Google. It can be changed to your signed-in email or an email of a Google Group managed by you.



Authentication

iOS </> C++

Autentifikácia

Sign-in metódy:

- vlastná autentifikácia cez email/password
 - môžete definovať viacero email-účov
- cez FB/Google/Twitter

SIGN-IN METHOD		EMAIL TEMPLATES
Sign-in providers		
Provider	Status	
Email/Password	Enabled	
Google	Enabled	
Facebook	Enabled	
Twitter	Enabled	
GitHub	Disabled	
Anonymous	Enabled	

Authentication

Users Sign-in method Templates Usage

pre začiatok a zoznámenie sa práce s FB vrelo odporúčam skúsiť niektoré aplikácie z balíka

GITHUB:

[quickstart-android-master](https://github.com/firebase/quickstart-android)
<https://github.com/firebase/quickstart-android>

Podprojekt Auth máte v VMA repozitóri

Search by email address, phone number, or user UID					Add user
Identifier	Providers	Created	Signed In	User UID ↑	
(anonymous)		Dec 1, 2016	Dec 1, 2016	1s70KNetLsYbmZ07ZE5qsHvah7q2	
vmaandroid@yahoo.com		Nov 22, 2016	Dec 13, 2018	4eWsnHhdRPVgUAtV200q7V1gN...	
—		Dec 1, 2016	Dec 2, 2016	5UoBQBMAcXQ9rj2060MsG80IOJ...	
(anonymous)		Dec 2, 2018	Dec 2, 2018	7v4jPhXKQBVz5zq86oUuhikAW1t1	
prostrediahm@gmail.com		Dec 7, 2019	Dec 7, 2019	861aiShu8QsZoz87Zv5Ms6cHU6Rl2	

Project:auth.zip

<https://github.com/firebase/quickstart-android>



Authentication

iOS C++

Project Setting

<https://console.firebase.google.com/u/0/project/kamaratinamape/settings/general/android:com.example.firebasedemo1>

Your apps

- meno projektu (support email)
- rovnaké API pre všetky Android apps (package)
- SHA1-certificatite – potrebné pre Google login
- z FB-konzoly si stiahnite google-services.json do projektu

Add app

google-services.json

com.google.firebase.quickstart.an...

com.google.firebase.quickstart.auth

com.google.firebase.quickstart.auth

FBAuth
com.google.firebase.quickstart.auth.j...

quickstart
com.google.firebase.quickstart.datab...

FBDatabase
com.google.firebase.quickstart.datab...

com.google.firebase.quickstart.fcm

FBMessaging
com.google.firebase.quickstart.fcm.ja...

App ID

1:539843735083:android:87fcb66cc6ca320a

App nickname

Add a nickname

Package name

com.google.firebase.quickstart.auth

SHA certificate fingerprints

Type

7a:94:75:11:dd:3d:57:2a:36:ed:2a:f2:76:13:a0:b8:68:0f:67:f1

SHA-1

Add fingerprint

google-services .json

```
{ "project_info": {  
  "project_number": "539843735083",  
  "firebase_url": "https://kamaratinamape.firebaseio.com",  
  "project_id": "kamaratinamape",  
  "storage_bucket": "kamaratinamape.appspot.com"  
},  
"client": [  
  { "client_info": {  
    "mobilesdk_app_id": "1:539843735083:android:e4c17d2977753b25",  
    "android_client_info": { "package_name": "sk.uniba.fmph.dai.borovan.fbdemo"  
    }  
  },  
  { "oauth_client": [{  
    "client_id": "539843735083-e4n6dg61g1npk7uka8ebf2rhcmg4t7v1.apps.googleusercontent.com",  
    "client_type": 3  
  }  
  ],  
  "api_key": [ { "current_key": "AIzaSyCbfmtNkbnhj1qanA051uSfQ11_PTjPa8" } ],  
  "services": {  
    "analytics_service": { "status": 1 },  
    "appinvite_service": { "status": 1, "other_platform_oauth_client": [] },  
    "ads_service": { "status": 2 }  
  }  
  ],  
  "configuration_version": "1"  
}
```

```
\quickstart-android-master\auth\app  
.  
..  
.gitignore  
app.iml  
auth-app.iml  
build  
build.gradle  
google-services.json  
proguard-rules.pro  
src  
58 bytes  
64 bytes free
```

Tento súbor potrebujete mať v projekte, stiahnite a do pod-adresára \app

Project:auth.zip

<https://github.com/firebase/quickstart-android>



Iná autentifikácia

Pozrite si [Facebook](#), resp. [Google](#) Login API

12:31

Firebase Authentication

Java
Run the Firebase Auth quickstart written in Java.
[OPEN](#)

12:31

Firebase Authentication

Kotlin
Run the Firebase Auth quickstart written in Kotlin.

GoogleSignInActivity
Use a Google Sign In credential to authenticate with Firebase.

FacebookLoginActivity
Use a Facebook Login credential to authenticate with Firebase.

TwitterLoginActivity
Use a Twitter Login credential to authenticate with Firebase.

EmailPasswordActivity
Use an email and password to authenticate with Firebase.

PasswordlessActivity
Use only an email to authenticate with Firebase.

PhoneAuthActivity
Use a phone number to authenticate with Firebase.

AnonymousAuthActivity
Sign in anonymously and then later upgrade to a full Firebase Auth user.

FirebaseUIActivity
Use FirebaseUI to authenticate with Firebase.



Authentication

iOS

Autentifikácia cez FaceBook

<https://firebase.google.com/docs/auth/android/facebook-login>



Facebook

Enable ☒

App ID

1286286781442318

App secret

663845afe79e444fafa94932a5d2cf7f

To complete set up, add this OAuth redirect URI to your Facebook app configuration. [Learn more](#)

https://kamaratinamape.firebaseio.com/_/auth/handler



CANCEL

SAVE

Autentifikácia cez FB/Twitter/GitHub API:

- predpokladá, že registrujete aplikáciu na FB/Twitter/Git developerskej konzole napr. <https://developers.facebook.com/apps/1286286781442318/settings/basic/>
- kde dostanete nejakú analógiu APP ID/Secret key
- tú/tie zapíšete do Firebase API vašej Firebase appky
- Firebase vám vygeneruje **google-services.json**, ktorý zakompilujete do .apk

Project:auth.zip

<https://github.com/firebase/quickstart-android>



Authentication

iOS

Facebook for developers

(dev konzola od FB)

facebook for developers Docs Tools Support My Apps Search developers.fb

FirebaseAuthApp APP ID: 1286286781442318 OFF Status: In D

Dashboard

Settings

Basic

Advanced

Roles

Alerts

App Review

PRODUCTS +

Facebook Login

App ID

1286286781442318

App Secret

663845afe79e444fafa94932a5d2cf7f

Reset

Display Name

FirebaseAuthApp

Namespace

App Domains

Contact Email

borovansky@gmail.com

Privacy Policy URL

https://kamaratinamape.firebaseio.com/__/auth/handler

Terms of Service URL

Terms of Service for Login dialog and App Details

<https://developers.facebook.com/apps/1286286781442318/settings/basic/>

Autentifikácia cez Twitter

<https://firebase.google.com/docs/auth/android/twitter-login>


<https://developer.twitter.com/en/apps/13160641>

Na Twitter developerskej konzole musíte registrovať aplikáciu, a získaťe Consumer API keys a Access token, ten prezradíte Firebase Console

Apps > [FirebaseAuthorisationDemo](#)

[App details](#) [Keys and tokens](#) [Permissions](#)

App details
Details and URLs

 **App icon**
App icon is default, click edit to upload.

App Name
FirebaseAuthorisationDemo

Description
asdfsdfsdfsdfsdfsdf

Website URL
<http://dai.fmph.uniba.sk/courses/VMA/android/>

Sign in with Twitter
Enabled

Callback URL
https://kamaratinamape.firebaseio.com/__/auth/handler

Project:auth.zip

<https://github.com/firebase/quickstart-android>

Autentifikácia cez Twitter

<https://developer.twitter.com/en/apps/13160641>

Apps > [FirebaseAuthorisationDemo](#)

App details

Keys and tokens

Permissions

Important notice about your access token and access token secret

To make your API integration more secure, we will no longer show your access token and access token secret beyond the first 10 characters. You will be able to regenerate it at anytime here, which will invalidate your current access token and secret. Please save this information. API keys, which will still be shown here as they are below. To learn more, [visit the Forums](#).

Keys and tokens

Keys, secret keys and access tokens management

Consumer API keys

k7YuSJH9qjJLeZn51N3TzCV0c (API key)

Xpz88UjjJTVQU61IkQWHxIOyiQIfkRJPu7qHAY0V83I1NCfCtu (API secret key)

Regenerate

Access token & access token secret

492951543-CZmXbG2ad1LICjIUs8H8RJo2ZmRIWLPWK4JLtzQW (Access token)

Sp8zOMrLSAw7mCarrLnpNwmshYFj5Eonwg9JsQG8DtSn9 (Access token secret)

Read-only (Access level)

Project:auth.zip

<https://github.com/firebase/quickstart-android>

Autentifikácia cez Twitter

<https://developer.twitter.com/en/apps/13160641>

API KEY a SECRET treba vložiť do Firebase Console pre Twitter Authentication



☒ Enable

API key

k7YuSJH9qjJLeZn51N3TzCV0c

API secret

Xpz88UjjJTVQU61IkQWHxIOyiQIfkRJPU7qHAY0V83l1NCfCtu

To complete set up, add this callback URL to your Twitter app configuration. [Learn more](#)

https://kamaratinamape.firebaseio.com/_/auth/handler



Cancel

Save

Project:auth.zip

<https://github.com/firebase/quickstart-android>



Authentication

iOS C++

Autentifikácia cez Google

<https://developers.google.com/android/guides/client-auth>

Musíte do Firebase projektu/aplikácie vložiť svoj SHA1 kľúč (viac bolo minule)
Google sign-in is automatically configured on your connected iOS and web apps. To set up Google sign-in for your Android apps, you need to add the [SHA1 fingerprint](#) for each app [Project Settings](#).

Android apps

com.google.firebase.quickstart.an...

com.google.firebase.quickstart.auth

FBAuth
com.google.firebase.quickstart.auth.j...

quickstart
com.google.firebase.quickstart.datab...

FBDatabase
com.google.firebase.quickstart.datab...

com.google.firebase.quickstart.fcm

FBMessaging
com.google.firebase.quickstart.fcm.ja...

Download the latest config file

[google-services.json](#)

This file contains configuration details such as keys and identifiers, for the services you just enabled.

App ID

1:539843735083:android:87fcb66cc6ca320a

App nickname

Add a nickname

Package name

com.google.firebase.quickstart.auth

SHA certificate fingerprints

Type

7a:94:75:11:dd:3d:57:2a:36:ed:2a:f2:76:13:a0:b8:68:0f:

Project:auth.zip

<https://github.com/firebase/quickstart-android>



Authentication

iOS C++

kamaratinamape

Spark plan

3 apps visible (max 3)

6 apps in project

		com.google.firebase.quickstart.ana...	
		com.google.firebase.quickstart.auth	
		quickstart com.google.firebase.quickstart.database	

Registrovanie appky

v project settings registrujete aplikáciu

Your project

Project name	kamaratinamape
Project ID	kamaratinamape
Cloud Firestore location	us-central
Web API Key	AIzaSyAcPPKWgkGfoDI-FAWNwQsJlj38EfsheWA

Public settings

These settings control instances of your project shown to the public

Public-facing name	kamaratinamape
Support email	vmaandroid@yahoo.com

Your apps

Android apps

- com.google.firebase.quickstart.an...
- com.google.firebase.quickstart.auth
- quickstart
com.google.firebase.quickstart.database

Add app

Download the latest config file

google-services.json

This file contains configuration details such as keys and identifiers, for the services you just enabled.

App ID

1:539843735083:android:a9b2141120408d37

na konfigurovaný google-services.json stiahnete a nakopírujete do vášho projektu, do ...app/

... a skompilujete

Registrovanie appky

1 Register app

! This project is limited to 30 apps. You can register up to 8 more apps. [Learn more](#)

Android package name ?

com.example.fbdemo1

App nickname (optional) ?

My Android App

Debug signing certificate SHA-1 (optional) ?

75:11:dd:3d:57:2a:36:ed:2a:f2:76:13:a0:b8:68:0f:67:f1

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

Register app

✓ Register app

Android package name: com.example.fbdemo1

2 Download and then add config file

Instructions for Android Studio below | [Unity](#) [C++](#)

Download google-services.json

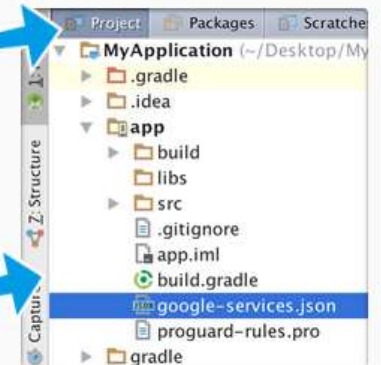
Switch to the **Project** view in Android Studio to see your project root directory.

Move your downloaded `google-services.json` file into your module (app-level) root directory.



google-services.json

Next



Registrovanie appky

1 Register app

! This project is limited to 30 apps. You can register up to 8 more apps. [Learn more](#)

Android package name ?

com.example.fbdemo1

App nickname (optional) ?

My Android App

Debug signing certificate SHA-1 (optional) ?

75:11:dd:3d:57:2a:36:ed:2a:f2:76:13:a0:b8:68:0f:67:f1

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

Register app

✓ Register app

Android package name: com.example.fbdemo1

2 Download and then add config file

Instructions for Android Studio below | [Unity](#) [C++](#)

Download google-services.json

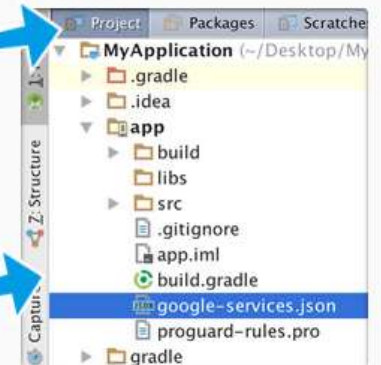
Switch to the **Project** view in Android Studio to see your project root directory.

Move your downloaded `google-services.json` file into your module (app-level) root directory.



google-services.json

Next



Registrovanie appky

3 Add Firebase SDK

Instructions for Gradle | [Unity](#) [C++](#)

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

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

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

```
buildscript {
    repositories {
        // Make sure that you have the following two repositories
        google() // Google's Maven repository
        mavenCentral() // Maven Central repository
    }
    dependencies {
        ...
        // Add the dependency for the Google services Gradle plugin
        classpath 'com.google.gms:google-services:4.3.13'
    }
}

allprojects {
    ...
    repositories {
        // Make sure that you have the following two repositories
        google() // Google's Maven repository
        mavenCentral() // Maven Central repository
    }
}
```

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

☐ Java ☒ Kotlin

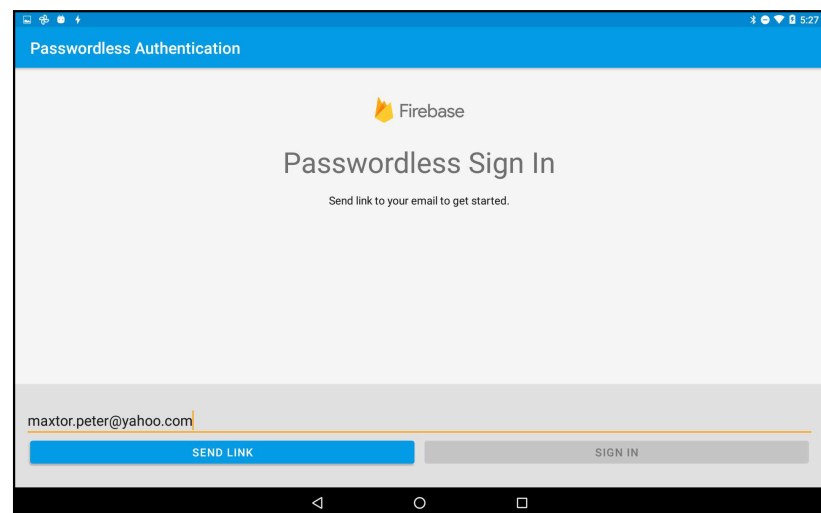
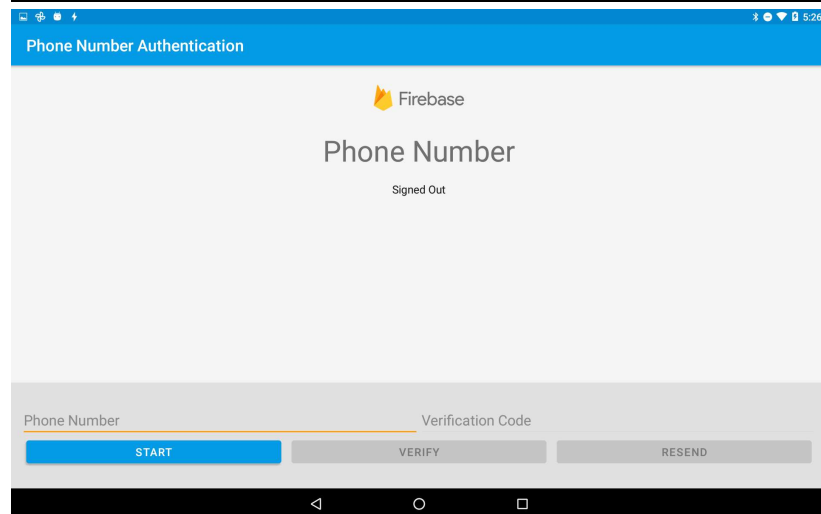
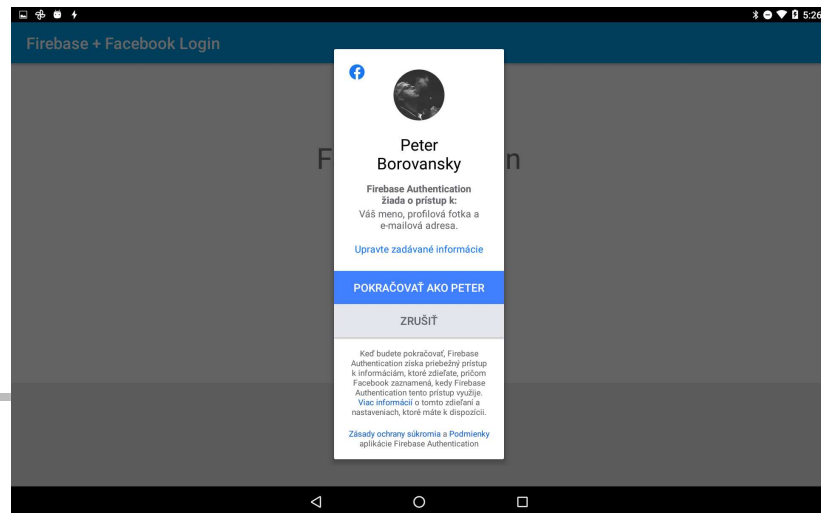
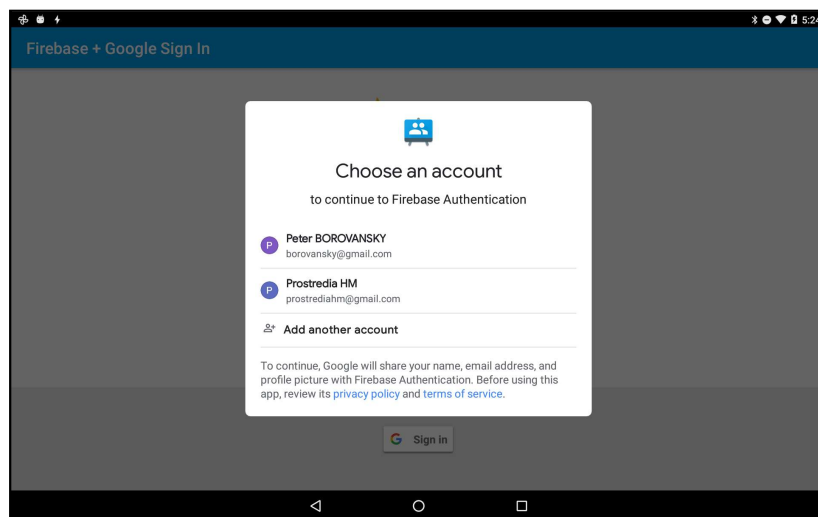
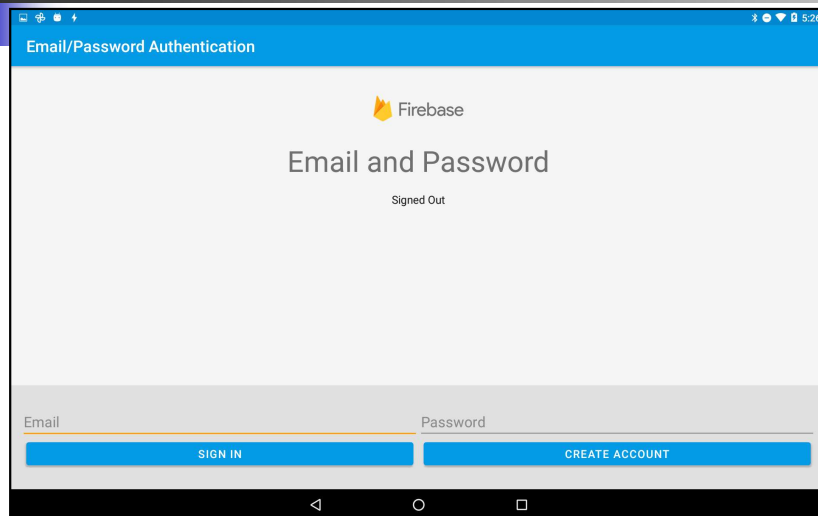
Module (app-level) Gradle file (`<project>/<app-module>/build.gradle`):

```
plugins {
    id 'com.android.application'
    // Add the Google services Gradle plugin
    id 'com.google.gms.google-services'
    ...
}

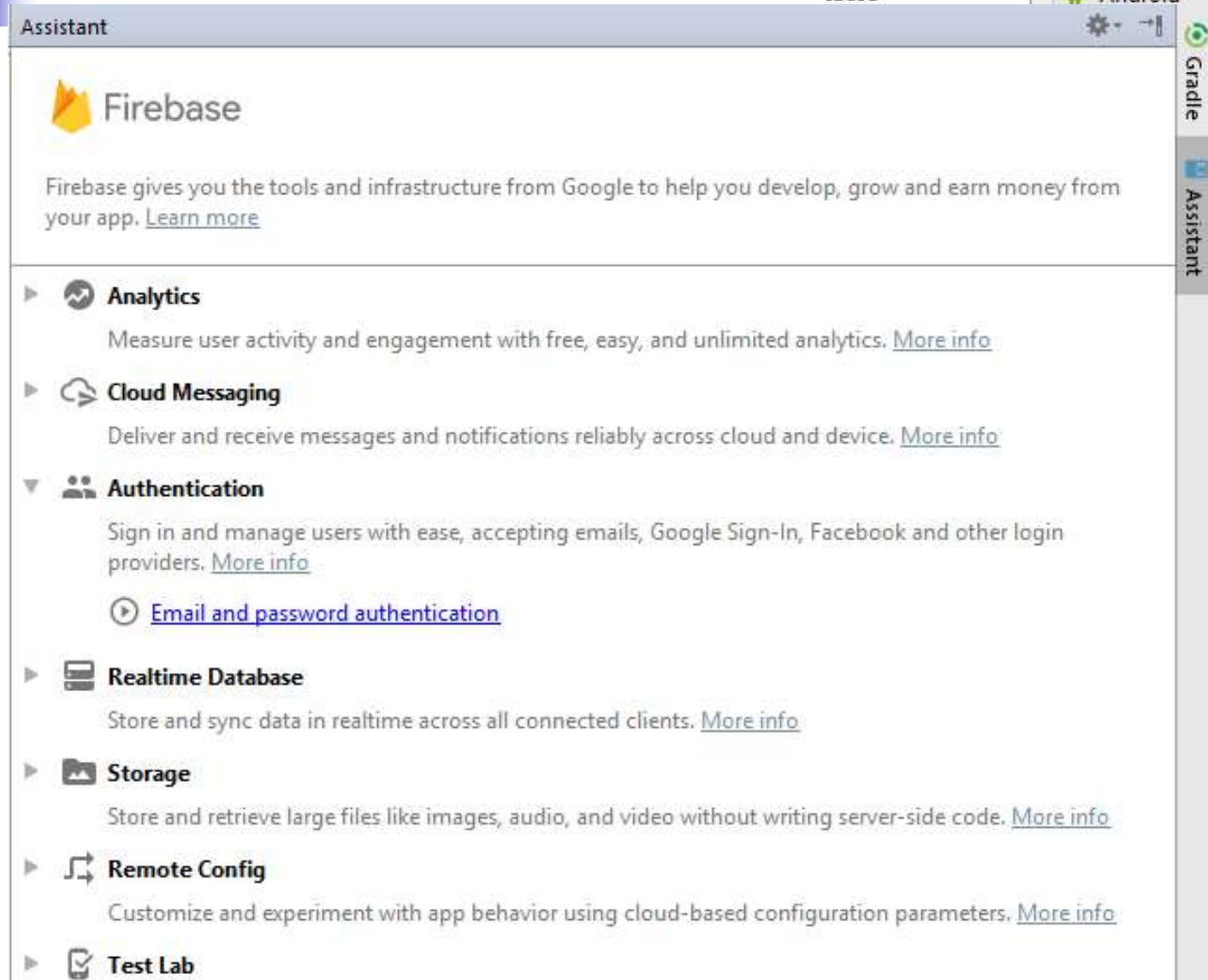
dependencies {
    // Import the Firebase BoM
    implementation platform('com.google.firebase:firebase-bom:31.1.1')

    // TODO: Add the dependencies for Firebase products you want to use
    // When using the BoM, don't specify versions in Firebase dependencies
    implementation 'com.google.firebase:firebase-analytics-ktx'

    // Add the dependencies for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}
```




Firebase podpora v AS











The screenshot shows the Android Studio IDE with the 'Tools' menu open, highlighting the 'Firebase' option. Below the menu, the 'Assistant' panel is visible, displaying the Firebase logo and a list of services available for integration into an Android application.

Assistant

 **Firebase**

Firebase gives you the tools and infrastructure from Google to help you develop, grow and earn money from your app. [Learn more](#)

- ▶  **Analytics**
Measure user activity and engagement with free, easy, and unlimited analytics. [More info](#)
- ▶  **Cloud Messaging**
Deliver and receive messages and notifications reliably across cloud and device. [More info](#)
- ▼  **Authentication**
Sign in and manage users with ease, accepting emails, Google Sign-In, Facebook and other login providers. [More info](#)
 - ▶  [Email and password authentication](#)
- ▶  **Realtime Database**
Store and sync data in realtime across all connected clients. [More info](#)
- ▶  **Storage**
Store and retrieve large files like images, audio, and video without writing server-side code. [More info](#)
- ▶  **Remote Config**
Customize and experiment with app behavior using cloud-based configuration parameters. [More info](#)
- ▶  **Test Lab**



Authentication

iOS C++

Firestore v AS

Assistant: **Firestore**



Firestore gives you the tools and infrastructure from Google to help you develop, grow and earn money from your app. [Learn more](#)

Analytics

Measure user activity and engagement with free, easy, and unlimited analytics. [More info](#)

Authentication

Sign in and manage users with ease using popular login providers like Google, Facebook, and others. You can even use a custom authentication system. [More info](#)

[Authenticate using Google](#)

[Authenticate using Google \[KOTLIN\]](#)

[Authenticate using Facebook Login](#)

[Authenticate using Facebook Login \[KOTLIN\]](#)

[Authenticate using a custom authentication system](#)

[Authenticate using a custom authentication system \[KOTLIN\]](#)

Realtime Database

Store and sync data with this cloud-hosted NoSQL database. Data is synced across all clients in realtime and remains available when your app goes offline. [More info](#)

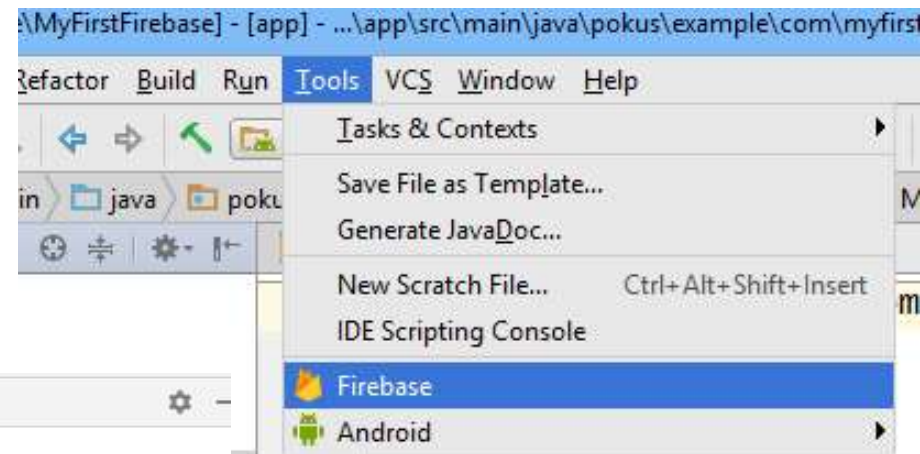
Cloud Firestore

Store and sync your app data with this flexible, scalable NoSQL cloud-hosted database. [More info](#)

Cloud Storage for Firestore

Store and retrieve large files like images, audio, and video without writing server-side code. [More info](#)

Cloud Functions for Firestore





Autentifikačný kit v AS

Assistant

← Firebase > Authentication

Email and password authentication

You can use Firebase Authentication to let your users sign in with their email addresses and passwords, and to manage your app's password-based accounts. This tutorial helps you set up an email and password system and then access information about the user.

[Launch in browser](#)

- 1 Connect your app to Firebase**
[Connect to Firebase](#)
- 2 Add Firebase Authentication to your app**
[Add Firebase Authentication to your app](#)

To use an authentication provider, you need to enable it in the [Firebase console](#). Go to the Sign-in Method page in the Firebase Authentication section to enable Email/Password sign-in and any other identity providers you want for your app.
- 3 Listen for auth state**

Declare the `FirebaseAuth` and `AuthStateListener` objects.

```
private FirebaseAuth mAuth;
```

```
private FirebaseAuth.AuthStateListener mAuthListener;
```

In the `onCreate()` method, initialize the `FirebaseAuth` instance and the `AuthStateListener` method so you can track whenever the user signs in or out.

```
mAuth = FirebaseAuth.getInstance();
```

Ponúka vám to step-by-step návod na vytvorenie rôznych typov Firebase aplikácií

Add Firebase Authentication to your app

Performing this action will make the following changes to your project.

```
build.gradle (project-level)
```

```
Add Firebase Gradle buildscript dependency
classpath 'com.google.gms:google-services:4.0.1'
```

```
app/build.gradle
```

```
Add Firebase plugin for Gradle
apply plugin: 'com.google.gms.google-services'
```

build.gradle will include these new dependencies:

```
compile 'com.google.firebase:firebase-auth:16.0.1:15.0.0'
```

This will also enable the firebase-core library which includes Firebase Analytics. [Learn more](#)

[Accept Changes](#) [Cancel](#)

1) Connect your app to Firebase

Success!

You've signed in to Android Studio.

To continue, go back to Android Studio.



Explore Google services you can now use in your Android app:



Firebase



Google Cloud Platform

Connect to Firebase

Firebase

☒ Create new Firebase project [What's this?](#) Signed in as vmaandroid@yahoo.com [Sign out](#)

☐ Choose an existing Firebase or Google project

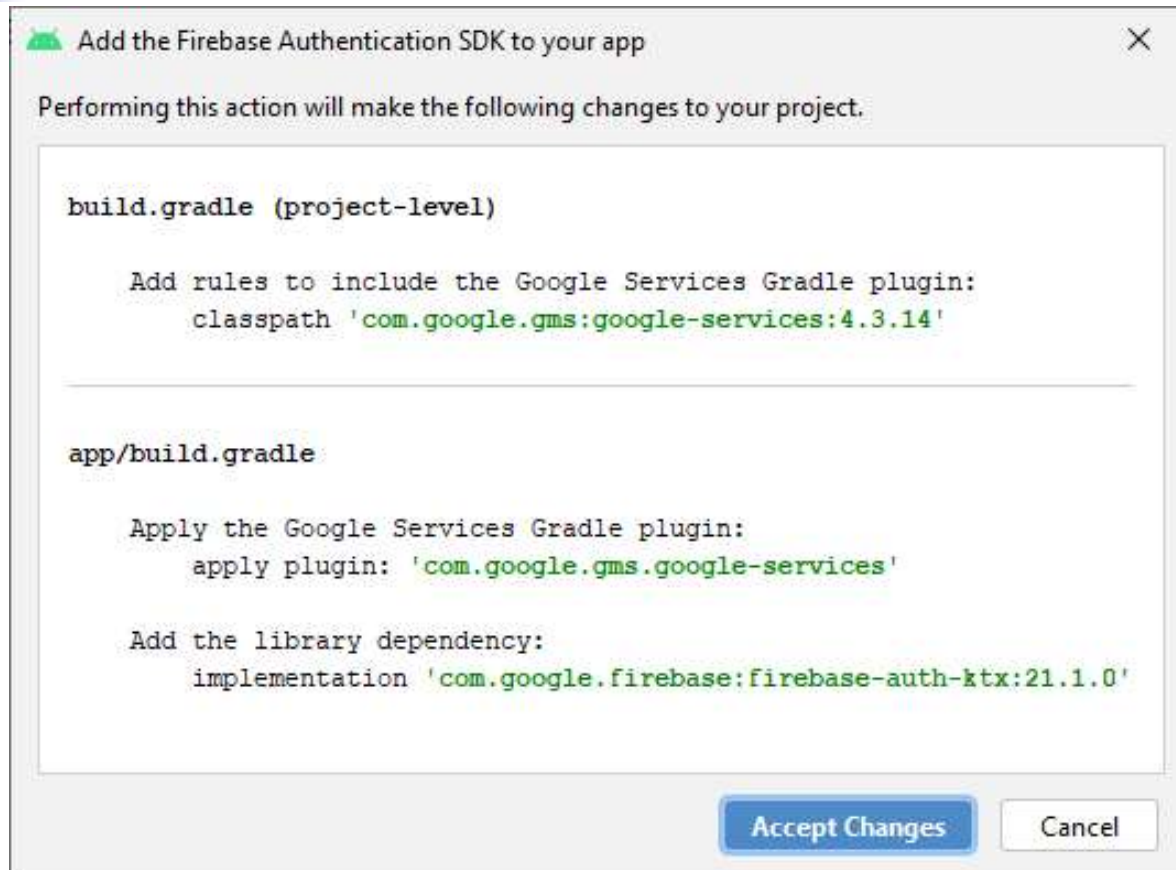
FBDemo	1 Android app(s) connected
kamaratinamape	11 Android app(s) connected

Country/region [What's this?](#)

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

Project:FBDemo1

2) Add Firebase Auth to your app





build.gradle (app)

```
dependencies {  
    ...  
    implementation 'com.google.firebase:firebase-auth-ktx:21.1.0'  
}
```

```
buildscript {  
    dependencies {  
        classpath 'com.google.gms:google-services:4.4.0'  
    }  
}  
plugins {  
    id 'com.android.application' version '7.3.1' apply false  
    id 'com.android.library' version '7.3.1' apply false  
    id 'org.jetbrains.kotlin.android' version '1.7.20' apply false  
}
```

```
plugins {  
    id 'com.android.application'  
    id 'kotlin-android'  
    id 'com.google.gms.google-services'  
}
```



Firebase Email Authentication

sign-up new user

```
lateinit var mAuth: FirebaseAuth
. . .
mAuth.createUserWithEmailAndPassword(email, password)
    .addOnCompleteListener(this, object:
        OnCompleteListener<AuthResult> {
            override fun onComplete(task: Task<AuthResult>) {
                if (task.isSuccessful()) { // Sign in success, update
                    Log.d(TAG, "createUserWithEmail:success")
                    val user = mAuth.currentUser
                } else { // If sign in fails,
                    Log.w(TAG, "createUserWithEmail:failure",
                        task.getException())
                }
            }
        })
```



Firestore Email Authentication

sign-in an existing user

```
lateinit var mAuth: FirebaseAuth
...
mAuth.signInWithEmailAndPassword(email, password)
    .addOnCompleteListener(this) { task ->
        if (task.isSuccessful) { // Sign in success
            Log.d(TAG, "signInWithEmail:success")
            val user = mAuth.currentUser
        } else { // If sign in fails
            Log.w(TAG, "signInWithEmail:failure", task.exception)
        }
    }
}
```




Firestore Email Authentication

get user info

```
val user = FirebaseAuth.getInstance().currentUser
if (user != null) { // Name, email address, and profile photo Url
    val name = user.displayName
    val email = user.email
    val photoUrl: Uri? = user.photoUrl
    val emailVerified = user.isEmailVerified
    val uid = user.uid
    infoTV.setText("$name, $email, $uid")
}
```

FirestoreDemo1

miki@sme.sk

qqqqqqq

SIGN IN

INFO

SIGNUP

null, miki@sme.sk, CWlvuMztroOnSvRxJZqxZmKGKNb2

Project:FirestoreDemo1

Firestore ako databáza

- noSQL databázy
- **Realtime Database** (efektívna pre mobilné app, synchronizácia)
 - ukladanie a synchronizácia v reálnom čase so všetkými pripojenými klientami
 - všetky dáta sú jeden veľký json dátový strom (JSON Tree)
 - existuje dávno, je stabilná, **regionálne** má veľmi slušnú latenciu
- **Cloud Firestore** (novinka)
 - Realtime updates, powerful queries, automatic scaling
 - dáta sú v kolekciách, hierarchicky organizované, sub-kolekcie, ...
 - novinka, beta r.2018, scalability, prepojenie cez viaceré dátové centrá



Obe podporujú offline support pre mobilných klientov (Android, iOS, web)
- zmeny počas off-line sa ukladajú do cache a synchronizujú, keď on-line

Firestore Database

nosql databáza - rules

- Default, no access, only FB console

```
service cloud.firestore {  
  match /databases/{database}/documents {  
    match /{document=**} {  
      allow read, write: if false;  
    }  
  }  
}
```

- Public, anyone can...

```
service cloud.firestore {  
  match /databases/{database}/documents {  
    match /{document=**} {  
      allow read, write: if true;  
    }  
  }  
}
```

- FB/Google/Git Authenticated only

```
service cloud.firestore {  
  match /databases/{database}/documents {  
    match /{document=**} {  
      allow read, write: if request.auth != null;  
    }  
  }  
}
```

Simulated read allowed

```
service cloud.firestore {  
  match /databases/{database}/documents {  
    match /{document=**} {  
      allow read, write: if request.auth != null;  
    }  
  }  
}
```

Rules Playground

Simulation type

get

Location

/databases/(default)/documents

collection/users

Authenticated



All fields are optional

Provider ⓘ

google.com

Run

Realtime Database

nosql databáza - rules

Default, no access, only FB console

```
// These rules don't allow anyone read or write access to your database
{
  "rules": {
    ".read": false,
    ".write": false
  }
}
```

Public, anyone can...

```
// These rules give anyone, even people who are not users of your app,
// read and write access to your database
{
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

FB/Google/Git Authenticated only

```
{
  "rules": {
    "users": {
      "$uid": {
        ".read": "$uid === auth.uid",
        ".write": "$uid === auth.uid"
      }
    }
  }
}
```

<https://kamaratinamape.firebaseio.com/>

★ Default security rules require users to be authenticated

kamaratinamape

```
4eWsnHhdRPVgUAitV200q7V1gNOI3: "treti status"
- pikatchus
  address: "treti status"
  lati: 48
  longi: 17
  name: "treti"
  time: 1480539003061
- statuses
  druhy: "druhy status"
  prvy: "prvy status"
  tretí: "treti status"
```

Realtime Database

DATA

RULES

USAGE

BACKUPS

Firestore Email Authentication

★ Default security rules require users to be authenticated

```
1 {
2   "rules": {
3     ".read": "auth != null",
4     ".write": "auth != null"
5   }
}
```

Realtime Database

nosql databáza - rules

Panel view **Query builder**

Run Clear

Query scope Path ⓘ
Collection ▾ /users

Where born >= number 1915
[Add where condition](#)

Order by born ascending
[+ Add to query](#)

Document ID	born
Jcu8pq0pijBsQZKHxab0	1915
bg4ZfwhM6Q2PY5RfWpK1	1915
fQ9OVH49aAVd1O3311k0	1915
I8KGCcjVz6Eiap1Q9SjS	1919

Realtime Database

zápis dát

```
databaseReference = FirebaseDatabase
    .getInstance()
    .reference
```

→ **databaseReference**.child("pikatchus")
object
 .setValue(Pokemon(name, address, lati, longi))

```
class Pokemon : Serializable
```

```
globalState == mutableMapOf<String,String>()
```

```
globalState[name]=status // prvy -> prvy status
```

→ **databaseReference**.child("statuses").**setValue**(globalState)
kolekcia

```
val user = firebaseAuth.currentUser
```

```
if (user != null) { // ak je user nalogovaný
```

→ **databaseReference**.child(user.uid).setValue(status)
uid
}

<https://kamaratinamape.firebaseio.com/>

★ Default security rules require users to be authenticated

```
kamaratinamape
├── 4eWsnHhdRPVgUAtV200q7V1gNOI3 "treti status"
├── pikatchus
│   ├── address: "treti status"
│   ├── lati: 48
│   ├── longi: 17
│   ├── name: "treti"
│   └── time: 1480539003061
└── statuses
    ├── druhy: "druhy status"
    ├── prvy: "prvy status"
    └── tretí: "tretí status"
```



Realtime Database

čítanie dát - synchronizácia

Pri akejkol'vek zmene sa zavolá listener

→
po zápise

```
databaseReference.addValueEventListener (
```

```

    object: ValueEventListener() {
        override fun onDataChange(dataSnapshot: DataSnapshot) {
            for (child in dataSnapshot.children) { % toto sú zmeny
                val o = child.value % (child.key, child.value)
                val str = "changed " + child.key + " is: "
                val o = child.value % zmenil sa (key,o=value)
                if (o is Pokemon) % zmenil sa objekt
                    statusMemo.append("Pokemon has ")
                Log.d(TAG, child.key)
                editTextMemo.append(
                    "changed ${child.key} is: ${o.toString()}\n")
            }
        }
    }
}


```




→
object

```

kamaratinamape
├── 4eWsnHhdRPVgUAtV200q7V1gNOI3: "treti status"
├── pikachus
│   ├── address: "treti status"
│   ├── lati: 48
│   ├── longi: 17
│   ├── name: "treti"
│   └── time: 1480539003061
└── statuses
    ├── druhy: "druhy status"
    ├── prvy: "prvy status"
    └── tretí: "tretí status"

```

Realtime Database
iOS   


Firestore DB

vyskúšajte si hotovú appku

Facebook YouTube LinkedIn WhatsApp Telegram


Firestore Database

RECENTMY POSTSMY 1

 vmaandroid


ako zajtra?

rano

 vmaandroid

super


fff

 vmaandroid

caute

mam sa dobre

pre začiatok odporúčam
skúsiť aplikácie z balíka
GITHUB:
[quickstart-android-master](#)



Database

..... peter: 'super'

- user-posts

- 4eWsnHhdRPVgUAtV200q7V1gNOI3

- -LSjZahGLiNYENME_DoI

..... author: 'vmaandroid'
..... body: 'mam sa dobre'
..... starCount: 0
..... title: 'caute'
..... uid: '4eWsnHhdRPVgUAtV200q7V1gNOI3'

- -LSj_LgHehm5T5hH3QWn

..... author: 'vmaandroid'
..... body: 'fff'
..... starCount: 1
+ stars
..... title: 'super'
..... uid: '4eWsnHhdRPVgUAtV200q7V1gNOI3'

+ -LSz_lMq8oBo6Vo4dyFJ

- iC07AtCq0haDIA3r6Sv0bTuyIhr1

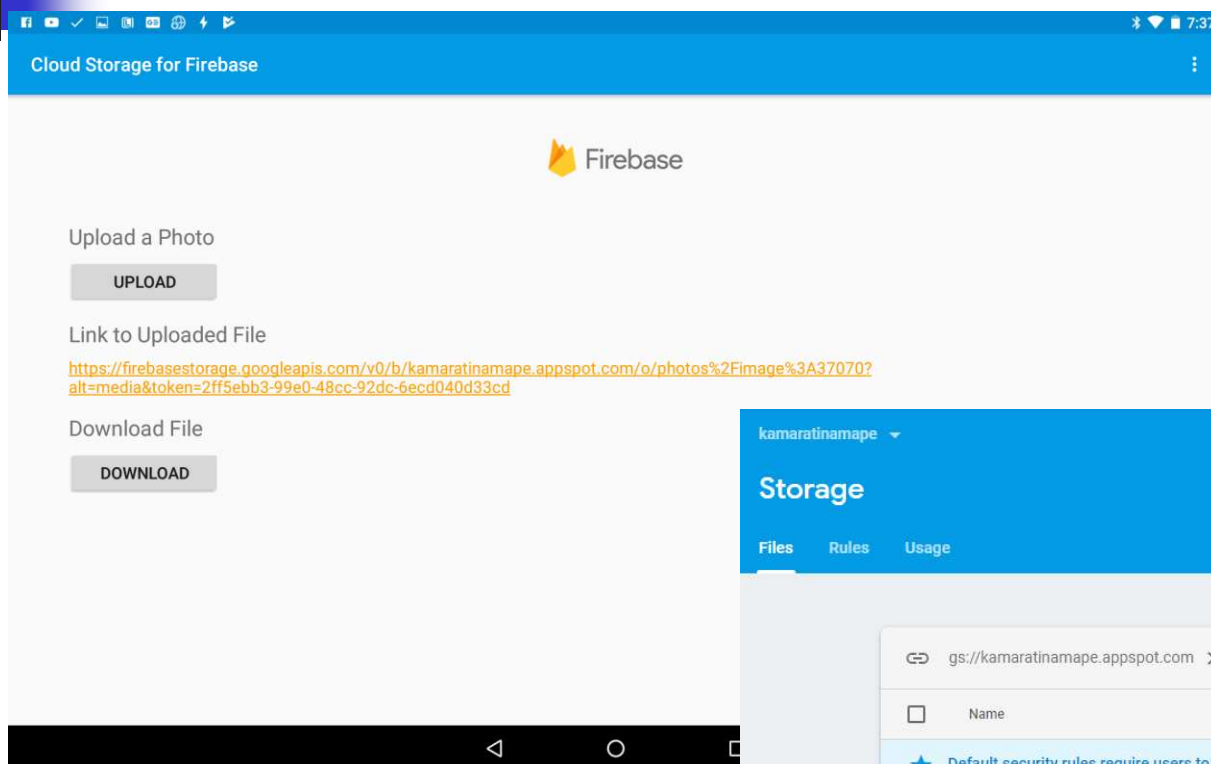
- -LSzaHe0sjuuGcpu7ybR

..... author: 'java'
..... body: 'World'
..... starCount: 0
..... title: 'Hello'
..... uid: 'iC07AtCq0haDIA3r6Sv0bTuyIhr1'

Project:database.zip

Firebase Storage

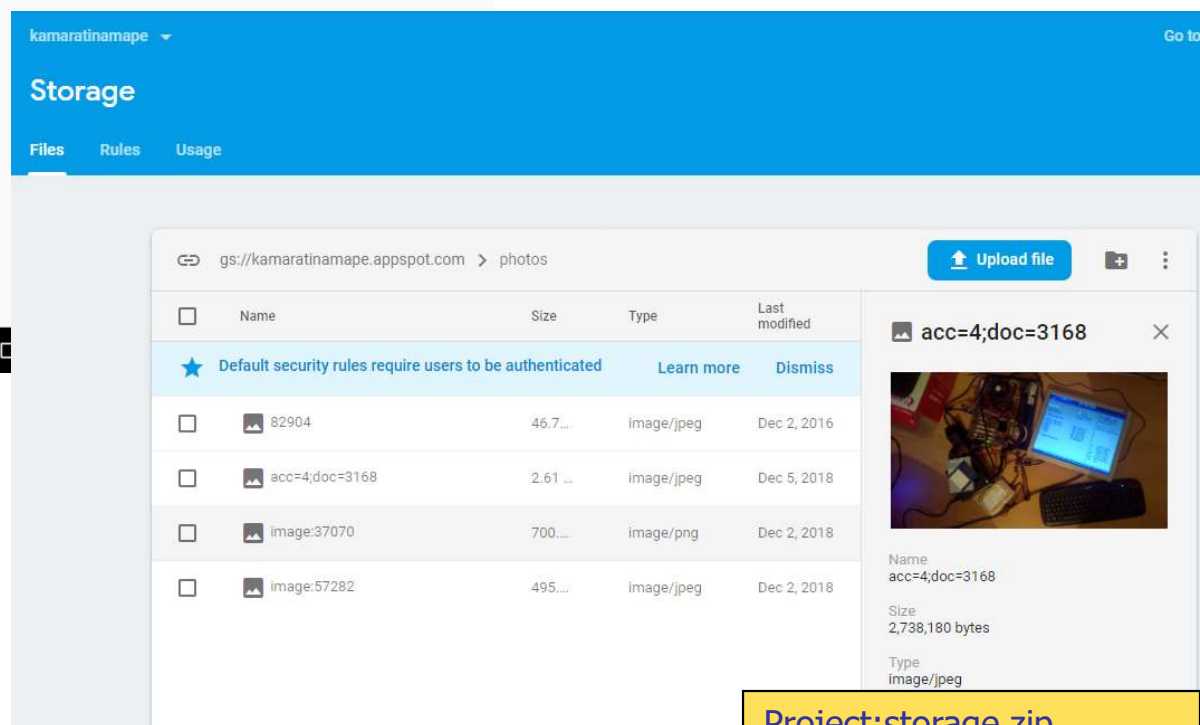
vyskúšajte si hotovú appku



pre začiatok odporúčam
skúsiť aplikácie z balíka

GITHUB:

[quickstart-android-master](https://github.com/firebase/quickstart-android-master)



Project:storage.zip



Firebase Cloud Messaging

(vyskúšajte si)

- Firebase Messaging – push notifikácie
- cross-platform (iOS, Android, ...)
- payload of up to 4kB to client app.

pre začiatok odporúčam
skúsiť aplikácie z balíka

GITHUB:

[quickstart-android-master](https://github.com/firebase/quickstart-android-master)

- Notification messages – when app is in the **background**

```
{"message":{  
  "token":"chLzRZ59Svk:APA91bGEy41ulMs3qQnThxYL6VWJAOu61pIHWkGTUEHQe4rWlyWL9yutLHxiwmgYdstis7T54I68yKhWZj95TnKXjUynd4rt2oLQ1gPAaIa249g2-h4MKSg7Xkgie8uCVPx8sbB_itLr",  
  "notification":{  
    "title":"Notification Test",  
    "body":"test"  
  }  
}}
```

- Data messages – key/value pairs received in a callback function.

```
{"message":{  
  "token":"chLzRZ59Svk:APA91bGEy41ulMs3qQnThxYL6VWJAOu61pIHWkGTUEHQe4rWlyWL9yutLHxiwmgYdstis7T54I68yKhWZj95TnKXjUynd4rt2oLQ1gPAaIa249g2-h4MKSg7Xkgie8uCVPx8sbB_itLr",  
  "data":{  
    "Nick": "Peter",  
    "body": "teacher",  
    "Room": "I-18"  
  }  
}}
```

Firebase Cloud Messaging

(vyskúšajte si)

Device Token

InstanceID Token:

chLzRZ59Svk:APA91bGEy41uIMs3qQnThxYL6VWJAOu61pIHWkGTUEHQe4rWlyWL9yutLHxiwmgYdstis7T54I68yKh
wZj95TnKXjUynd4rt2oLQ1gPAaIa249g2-h4MKSg7Xkgie8uCVPx8sbB_itLr

Sending a test message from Firebase Console (app is in background!)

1 Notification

Notification title (optional) ⓘ

Enter optional title

Test on device

You can test this campaign by entering or selecting the [FCM registration tokens](#) ⓘ of your development device below.

Add an FCM registration token

Recently Used ⓘ

☒ chLzRZ59Svk:APA91bGEy41uIMs3qQnThxYL6VWJAOu61pIHWkGT... ⌚ 🗑

Cancel Test

pre začiatok odporúčam
skúsiť aplikácie z balíka

GITHUB:
[quickstart-android-master](#)