

Firebase

Plan of this lecture

- What is Firebase?
- Cloud database and access rights.
- How to integrate it to your app.
- Homework

What is Firebase?

Firebase is a fully managed *platform* for building *iOS, Android, and web apps* that provides *automatic data synchronisation, authentication services, messaging, file storage, analytics, and more*. Starting with Firebase is an efficient way to build or prototype mobile backend services.

<https://console.firebaseio.google.com>

What is Firebase?

- A set of tools which provides a full suite for app development.
- NoSQL database.
- Base on node.js.
- Real time syncing with multiple devices.
- Ability to create applications with no server-side programming.
- Backend as a Service



Access via
 REST or SDK





Build better apps



Cloud Firestore

Store and sync app data at global scale



ML Kit BETA

Machine learning for mobile developers



Cloud Functions

Run mobile backend code without managing servers



Authentication

Authenticate users simply and securely



Hosting

Deliver web app assets with speed and security



Cloud Storage

Store and serve files at Google scale



Realtime Database

Store and sync app data in milliseconds



Improve app quality



Crashlytics

Prioritize and fix issues with powerful, realtime crash reporting



Performance Monitoring

Gain insight into your app's performance



Test Lab

Test your app on devices hosted by Google



App Distribution BETA

Distribute pre-release versions of your app to your trusted testers



Grow your business



In-App Messaging BETA

Engage active app users with contextual messages



Google Analytics

Get free and unlimited app analytics



Predictions

Smart user segmentation based on predicted behavior



A/B Testing BETA

Optimize your app experience through experimentation



Cloud Messaging

Send targeted messages and notifications



Remote Config

Modify your app without deploying a new version



Dynamic Links

Drive growth by using deep links with attribution



Build better apps



Cloud Firestore

Store and sync app data at global scale



ML Kit

Machine learning for mobile devices



Cloud Functions

Serverless functions for backend services



Authentication

Authenticate users simply and securely



Cloud Pub/Sub

Centralized event delivery with open protocols



Cloud Storage

Store and serve files from storage



Realtime Database

Decentralized real-time data synchronization



Improve app quality



Crashlytics

Automate crash detection with detailed reporting



Performance Monitoring

Get insights into your app's performance and usage



New relic

Get deeper data for backend monitoring via New relic



App Distribution

Distribute your app to millions of users via direct download



Grow your business



In-app Messaging

Engage with users with personalized messages



Google Analytics

Understand user behavior across your app



AdMob

Tap into user monetization through rewarded ads



Ad Testing

Test different ad formats and configurations



Cloud Messaging

Deliver instant messaging and notifications



Remote Config

Adjust code with minimal downtime



Responseable

Allow users to respond directly from your notifications

Firebase setup

Go to <https://console.firebaseio.google.com>

Recent projects



Add project



Explore a demo project

 Create a project (Step 1 of 3)

Let's start with a name for
your project

Project name

CiklumDating

 ciklumdating

Continue

Enter name and
press on
“Continue”



Google Analytics for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, Predictions, and Cloud Functions.

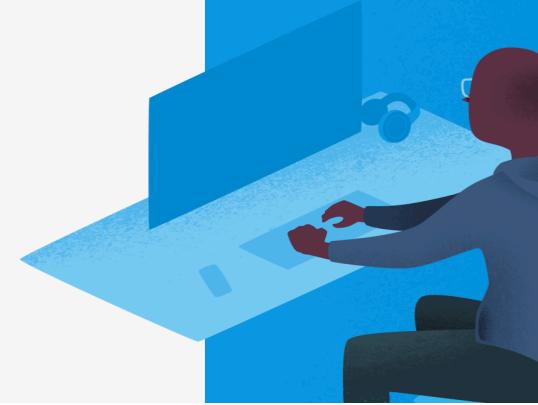
Google Analytics enables:

-  A/B testing [?](#)
 -  User segmentation & targeting across Firebase products [?](#)
 -  Predicting user behavior [?](#)
 -  Crash-free users [?](#)
 -  Event-based Cloud Functions triggers [?](#)
 -  Free unlimited reporting [?](#)
-  Enable Google Analytics for this project
Recommended

[Previous](#)

[Continue](#)

Just press
“Continue”



Configure Google Analytics

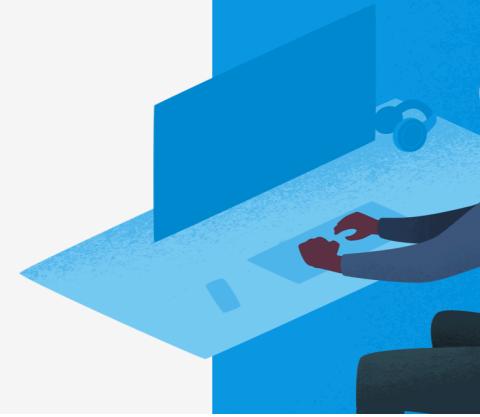
New Google Analytics account name

Cancel Save

After you choose a Google Analytics account and link it to your property, data exported from your Google Analytics property into Google Data Studio is subject to the Google Analytics terms of service. [Learn more.](#)

Create project

Select account
or create a new
one and then
press on
“Create project”



When project setup will finish you should see next picture:

[Project Overview](#)

Develop

[Authentication](#)[Database](#)[Storage](#)[Hosting](#)[Functions](#)[ML Kit](#)

Quality

[Crashlytics](#)[Performance](#)[Test Lab](#)[App Distribution](#)

Analytics

[Dashboard](#)[Events](#)[Conversions](#)[Audiences](#)[Funnels](#)[Extensions](#)

Spark

Free \$0/month

[Upgrade](#)

CiklumDating

Spark plan

Get started by adding Firebase to your app



Add an app to get started



Store and sync app data in milliseconds



Authentication



Cloud Firestore

First of all we need to setup firebase
for iOS and android projects.

For iOS project we need to get it bundle id. Open **Runner.xcworkspace** and navigate to **general** tab of project settings.



Runner

Flutter

Runner

- GoogleService-Info.plist
- Main.storyboard
- Assets.xcassets
- LaunchScreen.storyboard
- Info.plist

Supporting Files

- GeneratedPluginRegistrant.h
- GeneratedPluginRegistrant.m
- AppDelegate.swift
- Runner-Bridging-Header.h

Products

Pods

Frameworks

Pods

Runner

General

Signing & Capabilities

Resource Tags

Info

Build Settings

Build Phases

PROJECT

TARGETS

Runner

Identity

Display Name: Runner

Bundle Identifier: com.ciklum.mobile.datingApp

Version: 1.0.0

Build: 1

Deployment Info

Target	Device
iOS 8.0	<input checked="" type="checkbox"/> iPhone
	<input checked="" type="checkbox"/> iPad
	<input type="checkbox"/> Mac (requires macOS 10.15)

Main Interface: Main

Device Orientation

- Portrait
- Upside Down
- Landscape Left
- Landscape Right

CiklumDating

Spark plan

Get started by adding
Firebase to your app



Add an app to get started

Select iOS first



 Add Firebase to your iOS app

1 Register app

iOS bundle ID 

com.ciklum.mobile.datingApp

App nickname (optional) 

Flutter dating app

App Store ID (optional) 

123456789

Register app

2 Download config file

3 Add Firebase SDK

4 Add initialization code

5 Run your app to verify installation

Enter bundle id,
name and press
Register app



x Add Firebase to your iOS app

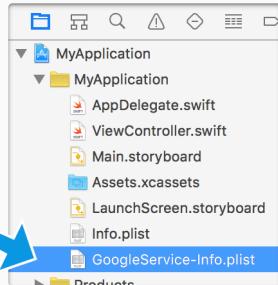
1 Register app

iOS bundle ID: com.ciklum.mobile.datingApp, App nickname: Flutter dating app

2 Download config file

[Download GoogleService-Info.plist](#)

Move the GoogleService-Info.plist file you just downloaded into the root of your Xcode project and add it to all targets.



Previous

Next

3 Add Firebase SDK

4 Add initialization code

5 Run your app to verify installation

Press on download, then add “GoogleService-Info.plist” to your Xcode project.

Press on “Next” and skip all other steps (press next on step 3 and 4, on step 5 press “Skip this step”).



The same steps for android application. But instead of bundle id we need applicationId. It's located at ***android/app/build.gradle*** (need to find applicationId)

EXPLORER

OPEN EDITORS

build.gradle android/app

FLUTTER_EXAM_APP-MASTER



> .vscode

FLUTTER_EXAM_APP-MASTER

FLUTTER_EXAM_APP-MASTER

> app

> src

build.gradle

{...} google-services.json

> gradle

build.gradle

gradle.properties

local.properties

settings.gradle

> build

> ios

> lib

flutter-plugins

.gitignore

.metadata

.packages

{...} analysis_options.yaml

pubspec.lock

{...} pubspec.yaml

READMD.md

build.gradle

android > app > build.gradle

```
31 |         sourceSets {  
32 |             main.java.srcDirs += 'src/main/kotlin'  
33 |         }  
34 |  
35 |         lintOptions {  
36 |             disable 'InvalidPackage'  
37 |         }  
38 |  
39 |         defaultConfig {  
40 |             // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html).  
41 |             applicationId "com.ciklum.mobile.dating_app"  
42 |             minSdkVersion 16  
43 |             targetSdkVersion 28  
44 |             versionCode flutterVersionCode.toInt()  
45 |             versionName flutterVersionName  
46 |             testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"  
47 |         }  
48 |  
49 |         buildTypes {  
50 |             release {  
51 |                 // TODO: Add your own signing config for the release build.  
52 |                 // Signing with the debug keys for now, so `flutter run --release` works.  
53 |                 signingConfig signingConfigs.debug  
54 |             }  
55 |         }  
56 |     }  
57 |  
58 |     flutter {  
59 |         source '../..'  
60 |     }  
61 |  
62 |     dependencies {  
63 |         implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"  
64 |         testImplementation 'junit:junit:4.12'  
65 |         androidTestImplementation 'com.android.support.test:runner:1.0.2'  
66 |         androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'  
67 |     }  
68 | }
```

× Add Firebase to your Android app

✓ Register app

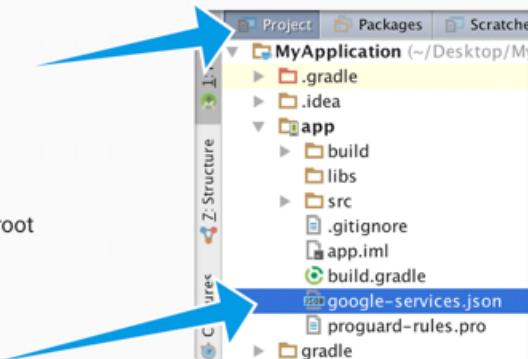
Android package name: com.ciklum.mobile.dating_app, App nickname: Flutter dating app

2 Download config file

[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 just downloaded into your Android app module root directory.



Previous

Next

Download ***google-services.json*** and place it to ***android/app*** folder. Then press on next and skip other steps (press next on step 3 and skip on step 4).

Now we are ready to setup
Authentication and **Database**.
Let's start with Authentication.



Firebase

Project Overview



Develop

Authentication

Database

Storage

Hosting

Functions

ML Kit

Quality

Crashlytics

Performance

Test Lab

App Distribution

CiklumDating ▾

CiklumDating

Spark plan

2 apps |



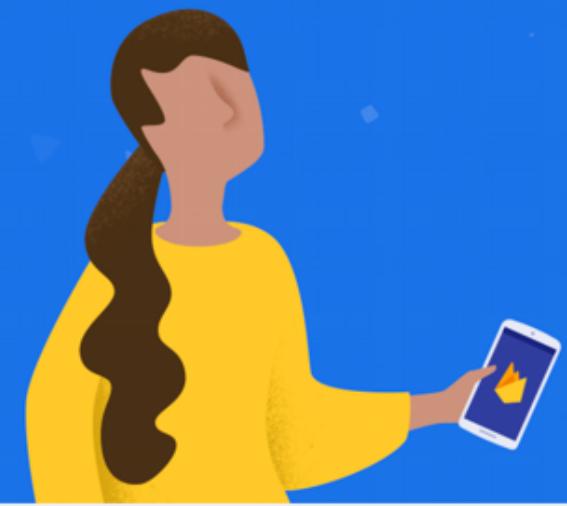
iOS Flutter datin...



Flutter datin...



Select Authentication at Develop menu.



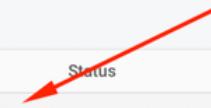


Authentication

Users **Sign-in method** Templates Usage

Sign-in providers

Provider	Status	
Email/Password	Disabled	
Phone	Disabled	
Google	Disabled	
Play Games	Disabled	
Game Center <small>Beta</small>	Disabled	
Facebook	Disabled	
Twitter	Disabled	
Github	Disabled	
Yahoo	Disabled	
Microsoft	Disabled	
Apple <small>Beta</small>	Disabled	



Select Sing-in
method tab and
press on Email/
Password

Sign-in providers

Provider	Status
Email/Password	<input checked="" type="checkbox"/> Enable
Email link (passwordless sign-in)	<input type="checkbox"/> Enable
Phone	Disabled

Then enable only top switcher and press on “Save”

We finish with authentication. Then we need to setup Database.

[Project Overview](#)

Develop

[Authentication](#)[Database](#)[Storage](#)[Hosting](#)[Functions](#)[ML Kit](#)

Quality

[Crashlytics](#)[Performance](#)[Test Lab](#)[App Distribution](#)

Analytics

CiklumDating

Spark plan

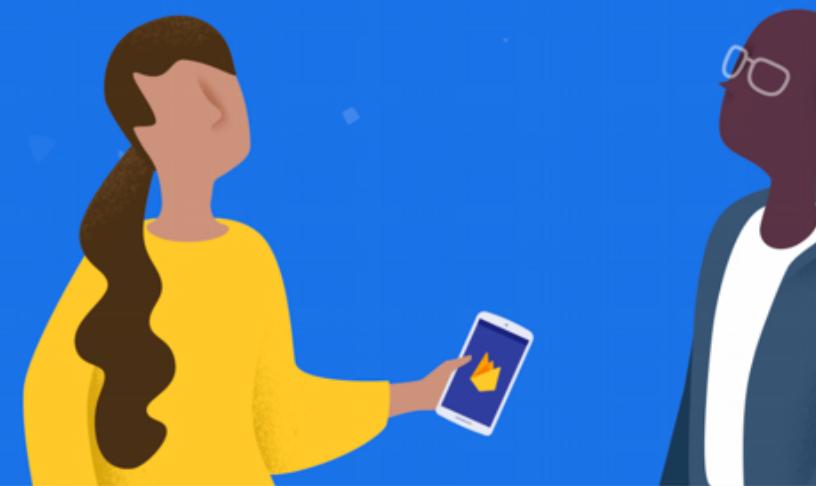
2 apps |

iOS Flutter datin... |

Flutter datin... |

+

Select Database from Develop menu



Waiting for Analytics data



Firebase

CiklumDating ▾

Go to docs

Project Overview



Develop

Authentication

Database

Storage

Hosting

Functions

ML Kit

Quality

Crashlytics

Performance

Cloud Firestore

Realtime updates, powerful queries, and automatic scaling

Create database

Press on "Create database"



Create database

1 Secure rules for Cloud Firestore

2 Set Cloud Firestore location

After you define your data structure, you will need to write rules to secure your data.

[Learn more](#)

Start in **production mode**

Your data will be private by default. Client read/write access will only be granted as specified by your security rules.

Start in **test mode**

Your data will be open by default to enable quick setup. Client read/write access will be denied after 30 days if security rules are not updated.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
        request.time < timestamp.date(2020, 1, 2);
    }
  }
}
```

! Anyone with your database reference will be able to read or write to your database for 30 days

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

Cancel

Next

Select “Start in test mode” (we will setup rules later) and press on “Next”

Create database



Secure rules for Cloud Firestore

2

Set Cloud Firestore location

Your location setting is where your Cloud Firestore data will be stored.

⚠ After you set this location, you cannot change it later. Also, this location setting will be the location for your default Cloud Storage bucket.

[Learn more](#)

Cloud Firestore location

nam5 (us-central)



Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

Cancel

Done

Select desired location and press on “Done”

That's all for initial setup. Let's switch to our app.

We need to add firebase libraries as dependencies at *pubspec.yaml*.

`firebase_core: ^0.4.0+9`

`firebase_auth: ^0.15.0+1`

`cloud_firestore: ^0.12.9+5`

Now we are ready to start development.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /users/{user}/favorites/{document=**} {
      allow read, write: if request.auth.uid == user;
    }
  }
}
```

Apply rule for cloud firestore

Links

- <https://console.firebaseio.google.com>
- <https://codelabs.developers.google.com/codelabs/flutter-firebase>
- <https://firebase.google.com/docs/flutter/setup>
- <https://firebase.google.com/docs/auth>
- <https://firebase.google.com/docs/firestore>
- <https://firebase.google.com/docs/firestore/security/get-started>

Homework

- Setup own firebase project.
- Add session restoration when open application.
- Add tab with favourite users.