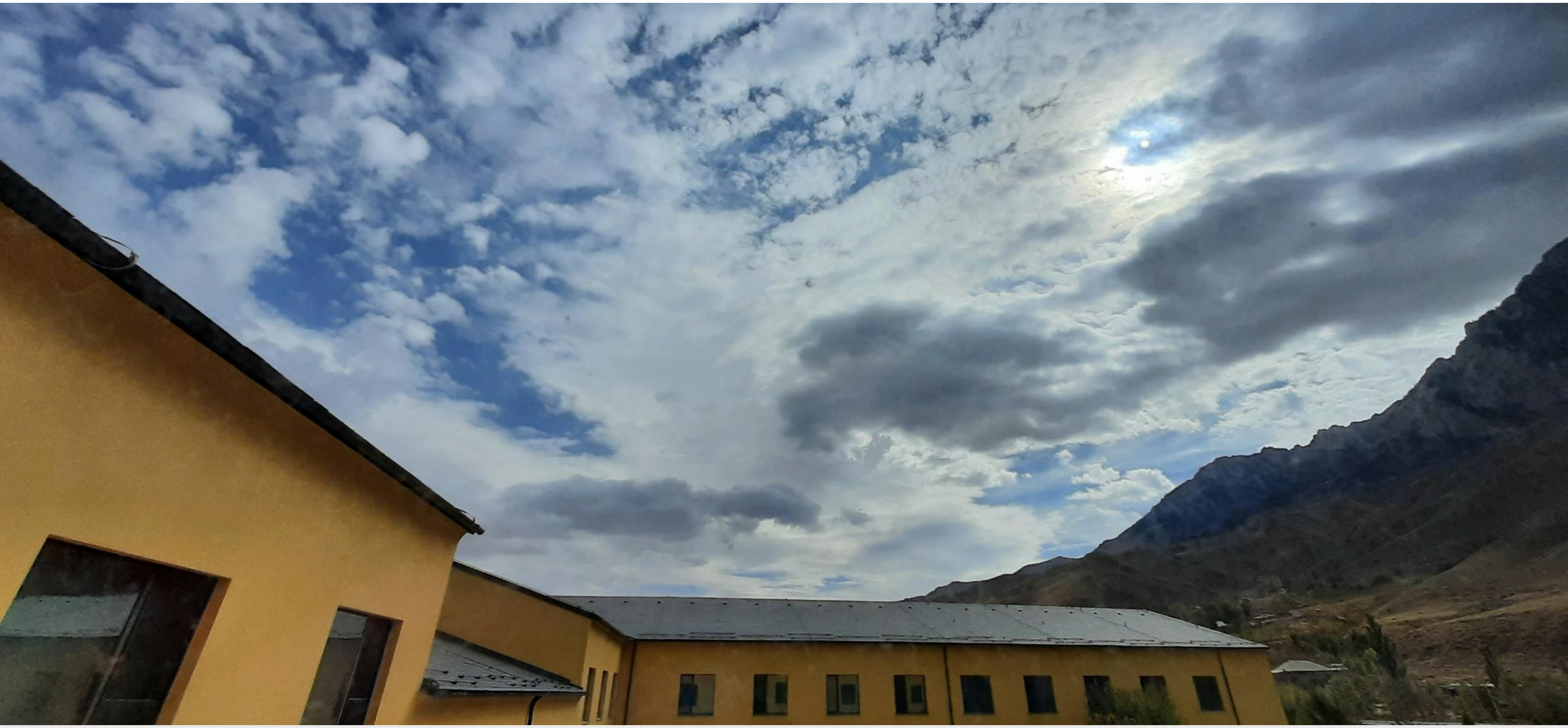
Adding Firebase to Android app

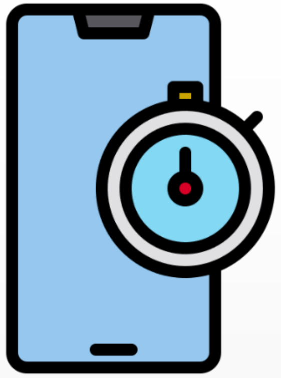
Dmytro Zubov, PhD
dmytro.zubov@ucentralasia.org

Naryn, Kyrgyzstan, 11:03am, Sept 29, 2022



Lessons learnt last time

- MQTT Messenger: Java Android app
- Types of variables in Java
- Types of modifiers in Java
- Java types
- Compound expressions in Java
- Bitwise operators in Java



What we gonna discuss today?

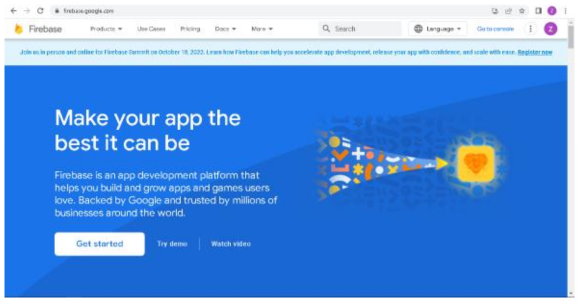
- What is Firebase?
- History of Firebase
- Why use Firebase?
- General architecture
- Services
- Adding Firebase Realtime Database to Android app

What is Firebase?

https://www.youtube.com/watch?v=vAoB4VbhRzM

- Firebase in 100 Seconds

What is Firebase?



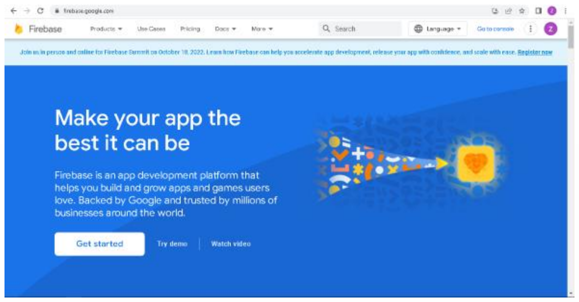
Join us in person and online for Firebase Summit on October 18, 2022. Learn how Firebase can help you accelerate app development, release your app with confidence, and scale with ease. Register now

Make your app the best it can be

Firebase is an app development platform that helps you build and grow apps and games users love. Backed by Google and trusted by millions of businesses around the world.

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

What is Firebase?



Join us in person and online for Firebase Summit on October 18, 2022. Learn how Firebase can help you accelerate app development, release your app with confidence, and scale with ease. Register now

Make your app the best it can be

Firebase is an app development platform that helps you build and grow apps and games users love. Backed by Google and trusted by millions of businesses around the world.

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

What is Firebase?

https://www.youtube.com/watch?v=dRYnm_k3w1w&feature=emb_logo

- Getting started with Firebase on Android

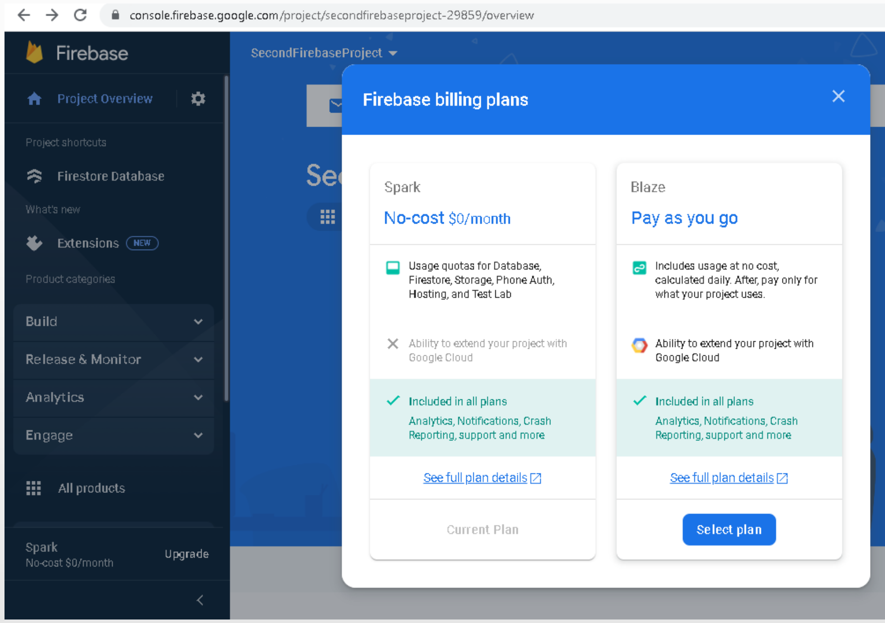
CHEMIN DE FER DU NORD
NORD EXPRESS
DEUTSCHE REICHSBAHN GES. = POLSKIE KOLEJE PAŃSTWOWE
BRUXELLES
LIEGE BERLIN VARSOVIE RIGA
DES WAGONS - LITS

History of Firebase

- Founded in 2011 by Andrew Lee and James Tamplin
- Initial product was a real-time database
- Over time it becomes a full suite for app development
- Acquired by Google in 2014

Why use Firebase?

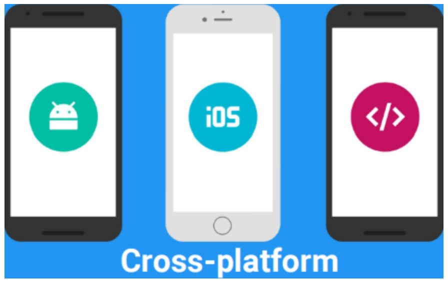
- Create app without backend server
- No need extra money spent for backend server. **<u>Ok, sometimes, we need :(</u>**
- Push notification
- NoSQL database so it is faster
- Auto backup
- and many more...


console.firebase.google.com/project/secondfirebaseproject-29859/overview

Firebase billing plans

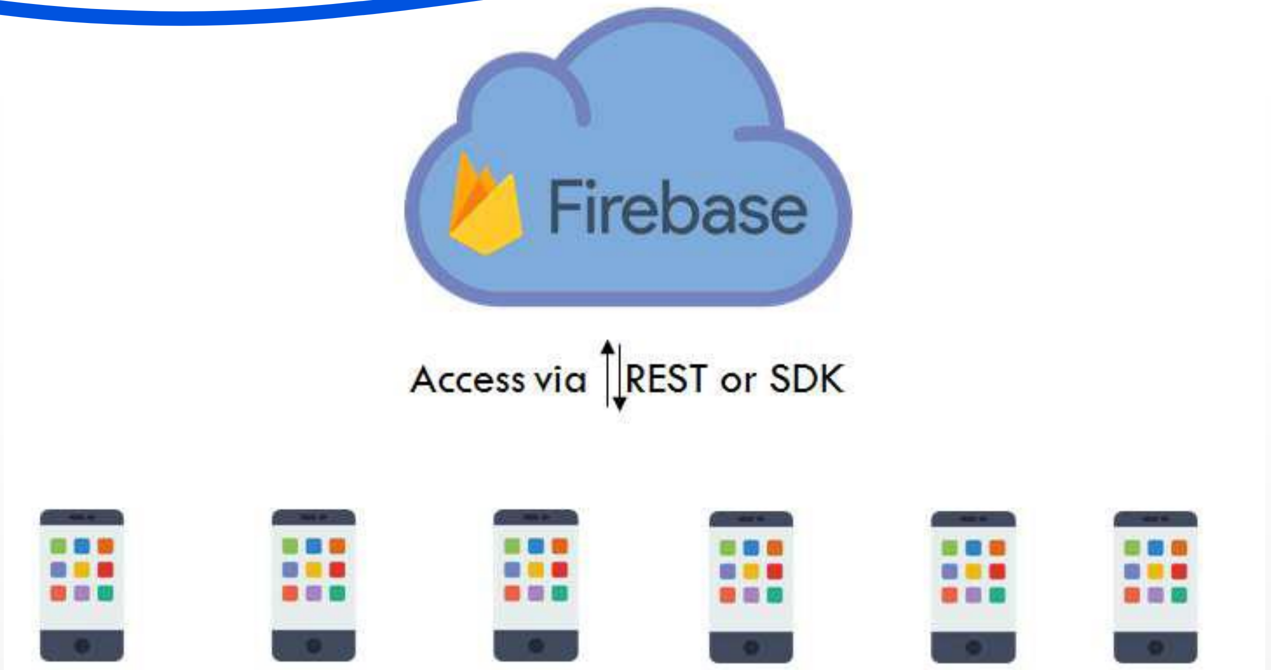
Spark	Blaze
No-cost $0/month	Pay as you go
Usage quotas for Database, Firestore, Storage, Phone Auth, Hosting, and Test Lab	Includes usage at no cost, calculated daily. After, pay only for what your project uses.
Ability to extend your project with Google Cloud	Ability to extend your project with Google Cloud
✓ Included in all plans Analytics, Notifications, Crash Reporting, support and more	✓ Included in all plans Analytics, Notifications, Crash Reporting, support and more
See full plan details	See full plan details
Current Plan	Select plan

General architecture



- Firebase client is added to an application by including the relevant Firebase library
- Firebase has libraries for JavaScript, Java, Android, iOS and a REST API. More features - ?
- Once it is added any data structure can be saved to Firebase
- This will automatically save data to Firebase backend and synchronize the data across various instances of the application

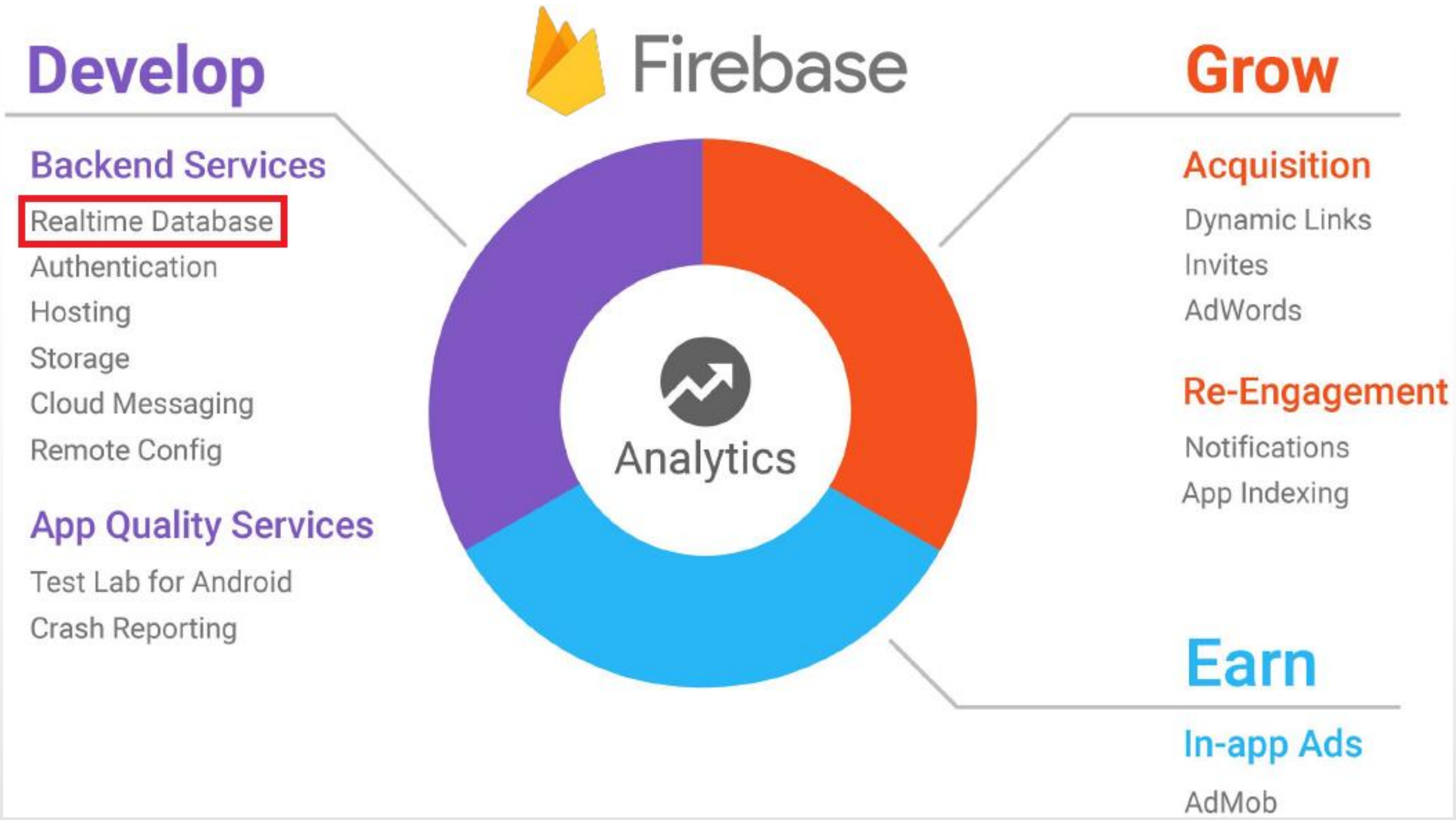
General architecture (cont.)



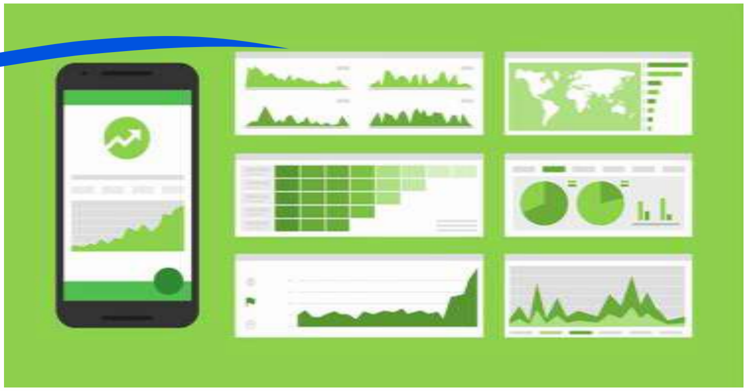
REST, or REpresentational State Transfer, is an architectural style for providing standards between computer systems on the web, making it easier for systems to communicate with each other.

https://www.codecademy.com/articles/what-is-rest

Services


Develop
Firebase
Grow
Backend Services
Realtime Database
Authentication
Hosting
Storage
Cloud Messaging
Remote Config
App Quality Services
Test Lab for Android
Crash Reporting
Analytics
Acquisition
Dynamic Links
Invites
AdWords
Re-Engagement
Notifications
App Indexing
Earn
In-app Ads
AdMob

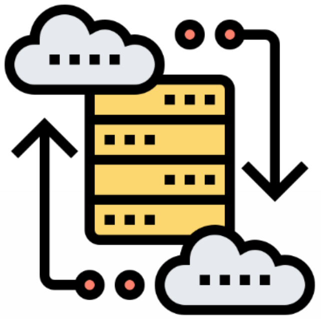
Services (cont.)



- **Analytics**
  - Unlimited reporting of 500 event types, each with up to 25 attributes
  - One dashboard to view user behavior and cross-network campaign performance
  - Demographic segmentation, including age, gender, and location, available out-of-the-box
  - Export raw data to BigQuery for custom querying

Services (cont.)

- **Cloud messaging**
  - Send unlimited upstream/downstream messages
  - Send messages to individual devices or a user segment
  - Handle all aspects of queuing and delivery
  - It can send billions of messages with 95% of messages sent in 250ms



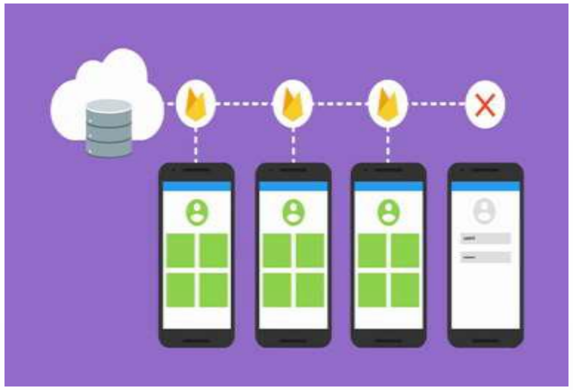
Services (cont.)

- **Authentication**
  - Support multiple social accounts
  - Optional, out-of-the-box authentication UI optimized to give your users the best experience
  - It can also integrate to your existing accounts
  - Advanced functionality like email verification, anonymous accounts, and account linking
  - Firebase will also manage user session

Services (cont.)

- **Realtime database**
  - A cloud-hosted NoSQL database
  - Intuitive and easy-to-use API
  - Remains responsive regardless of network latency or Internet connectivity
  - Handles the complexity of real time synchronization and provides flexible conflict resolution
  - Accessible directly from client SDKs, or from the server with the REST API

Services (cont.)



- **Storage**
  - Robust uploads and downloads in the background, regardless of network quality
  - Secure client-side authorization, integrated with Authentication
  - Petabyte scale data storage backed by Google Cloud Storage
  - API access throughout Firebase or Google Cloud Storage APIs

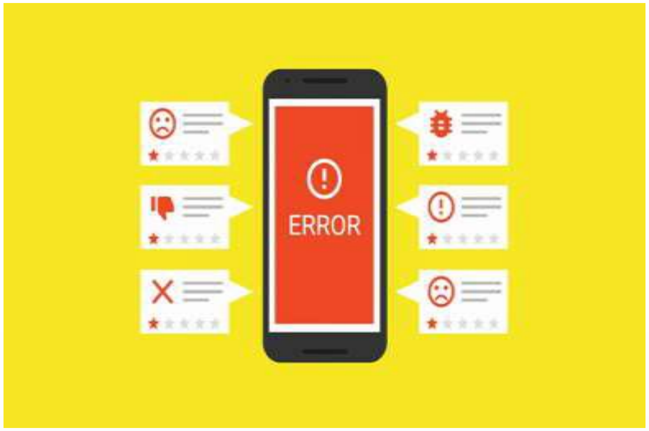
Services (cont.)

- **Hosting**
  - Automatically provisioned SSL certificate
  - Support for client-side routing
  - Blazing-fast content worldwide
  - Atomic deploys and one-click rollbacks on one command
  - Every site is served over secured connection

Services (cont.)

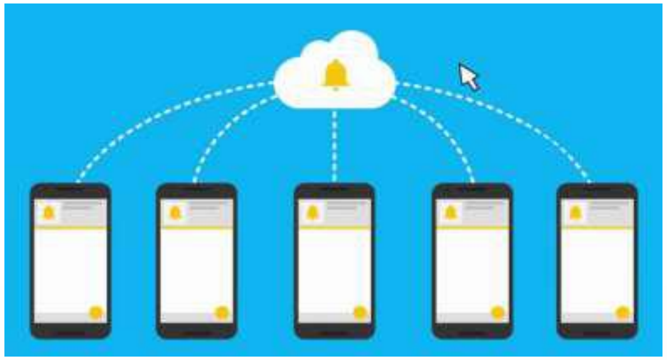
- **Test lab**
  - Generate detailed reports and screenshots to help identify bugs
  - Run custom test scripts on hundreds of device configurations
  - Supplement your existing workflow through integration with Android Studio, command-line tools, and Web-based consoles

Services (cont.)



- **Crash reporting**
  - Prioritize crashes by frequency and impact
  - Comprehensive data surrounding each crash, including device characteristics, device circumstances, a stack trace, and more
  - Reliably collect crashes that occur while the device is online or offline

Services (cont.)



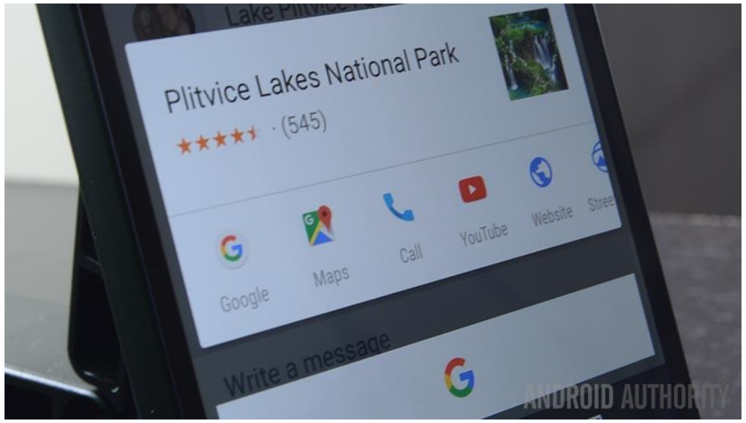
- **Notification**
  - Send free and unlimited notifications across Android and iOS
  - Send messages and analyze effectiveness in one dashboard without writing any code
  - Integrate with Firebase Analytics to deliver messages to a user segment

Services (cont.)

- **Remote config**
  - Modify your app without a new production deployment
  - Customize content for different Firebase Analytics audiences and measure results
  - Roll out features gradually and monitor the impact

Services (cont.)

- **App indexing**

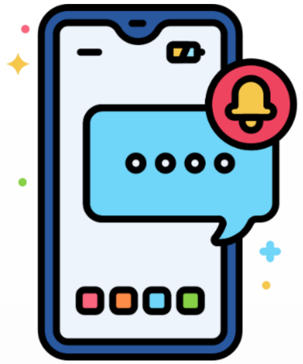


  - Show your in-app content via Google Search
  - Make your content accessible through auto complete and *Now on Tap* for Android device
  - Improve your app ranking in Google Search results

Services (cont.)

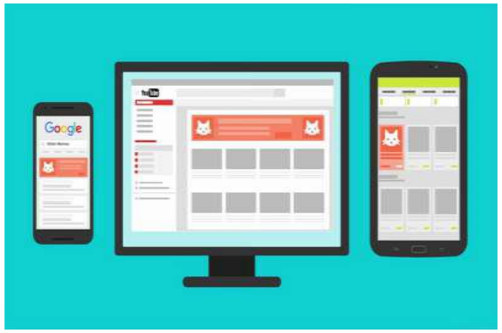
- **Dynamics links**
  - Improve acquisition and engagement by bringing users directly to content that they were originally searching for, whether they have your app installed or not
  - Delight new users with personalized promotions and messages after install

Services (cont.)



- **Invites**
  - Invite the most relevant contacts with smart suggestions
  - Free email and SMS delivery
  - Powered by Firebase Dynamic Links

Services (cont.)



- **AdWords**
  - Conversion tracking for first opens and in-app events without implementing any additional SDKs
  - Cross-network attribution measurement and LTV in one dashboard
  - Show ads to users based on user segments from Firebase Analytics

Services (cont.)

- **AdMOB**
  - Show ads from millions of Google advertisers competing in real time
  - Choose a format to suit your app, including banner, video and native ads
  - Work with more than 40 top ad networks using AdMob Mediation
  - Cross-promote between your apps for free with AdMob house ads

Adding Firebase Realtime Database to Android app

- About JSON structured data
  - The data stored in the Firebase Realtime Database is JSON structured, i.e., the entire database will be a JSON tree with multiple nodes
  - Unlike SQL database, Firebase Realtime Database doesn't have tables or records in the JSON tree. Whenever we are adding some data to the JSON tree, then it becomes a node in the existing JSON structure with some key associated with it. All Firebase Realtime Database data is stored as JSON objects.
  - Following is an example of JSON structured data:

```
{
     "company": {
          "name": "MindOrks",
          "address": "Gurugram"
     }
}
```

Adding Firebase Realtime Database to Android app

- Prerequisites
  - Install or update Android Studio to its latest version (ok, 4.0 is good enough)
  - Make sure that your project meets these requirements:
    - Targets API level 16 (Jelly Bean) or later
    - Uses Gradle 4.1 or later
    - Uses Jetpack (AndroidX), which includes meeting these version requirements:
      - com.android.tools.build:gradle v3.2.1 or later
      - compileSdkVersion 28 or later
  - Set up a physical device or use an emulator to run your app
  - Emulators must use an **emulator image with Google Play**
  - Sign into Firebase using your Google account

Oct 3, 2022:
Android Studio Dolphin 2021.3.1

Adding Firebase Realtime Database to Android app

- Database configuration rules
  - The data present in the database is very important and we shouldn't give access to everyone to use the data present in our database. So, Firebase Realtime Database has some database configuration rules that can be used to provide different access to different users:

    - Default

    - Public

    - User

https://blog.mindorks.com/firebase-realtime-database-android-tutorial

Adding Firebase Realtime Database to Android app

- Database configuration rules
  - *Default*: By default, the read and write access to our database is disabled and no one can read or write data from the database in this mode. Here, we can access the data of the database from the Firebase Console only.

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

Adding Firebase Realtime Database to Android app

- Database configuration rules
  - *Public*: By using public rules, anyone can change the data presented in the database. This rule is generally used when we are testing our application and after testing the app, we can set the rule to User only.

```
// 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
  }
}
```

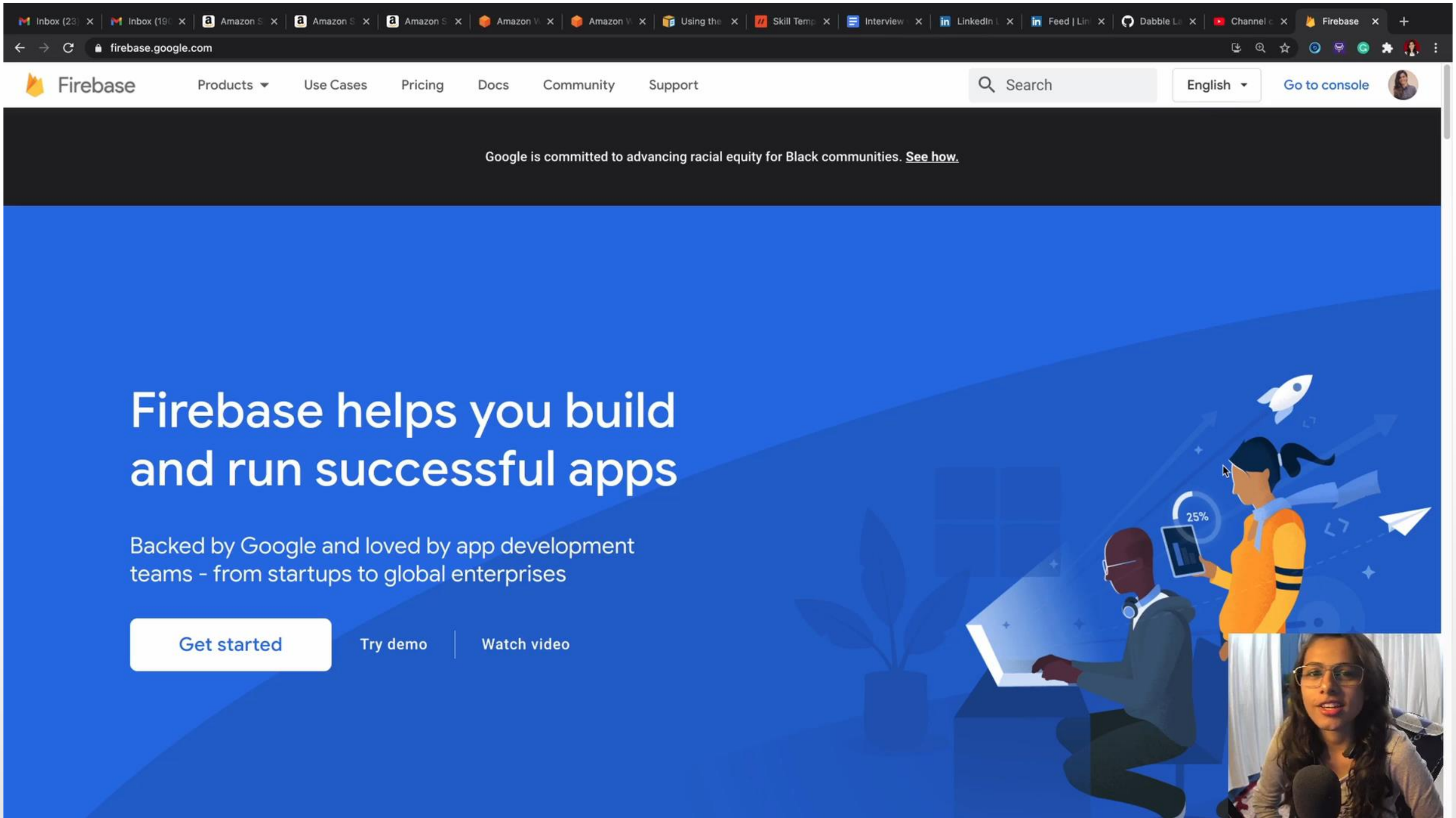
Adding Firebase Realtime Database to Android app

- Database configuration rules
  - *User*: In this rule, the user of our application can read and write in our database. We can authenticate our user using the Firebase Login and Authentication and after that, our user will have the access to our database.

```
// These rules grant access to a node matching the authenticated
// user's ID from the Firebase auth token
{
  "rules": {
    "users": {
      "$uid": {
        ".read": "$uid === auth.uid",
        ".write": "$uid === auth.uid"
      }
    }
  }
}
```

Firebase Realtime Database Creation in 5 mins

https://www.youtube.com/watch?v=qKxisFLQRpQ

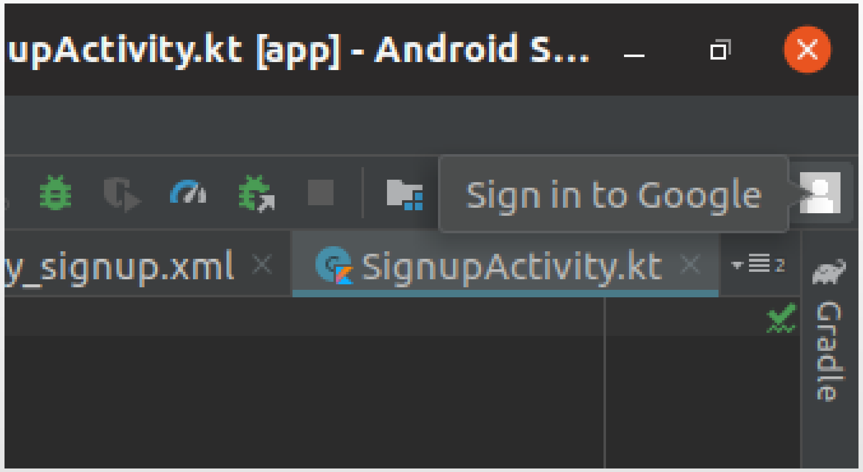

firebase.google.com
Firebase
Products
Use Cases
Pricing
Docs
Community
Support
Search
English
Go to console
Google is committed to advancing racial equity for Black communities. See how.
Firebase helps you build and run successful apps
Backed by Google and loved by app development teams - from startups to global enterprises
Get started
Try demo
Watch video
25%

Adding Firebase Realtime Database to Android app

- Create the Firebase Realtime database and connect it to our Android project

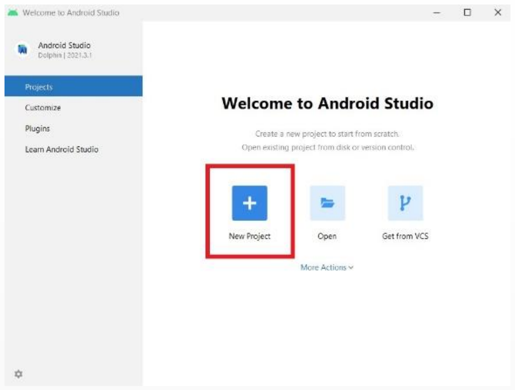
  Step 1. Open Android Studio and create a new project or open an existing project.

  Step 2. In Android Studio, log in with your Google account email. You can find the login button at the top right corner of the Android Studio. (Pls remember the email ID that you have used here :)

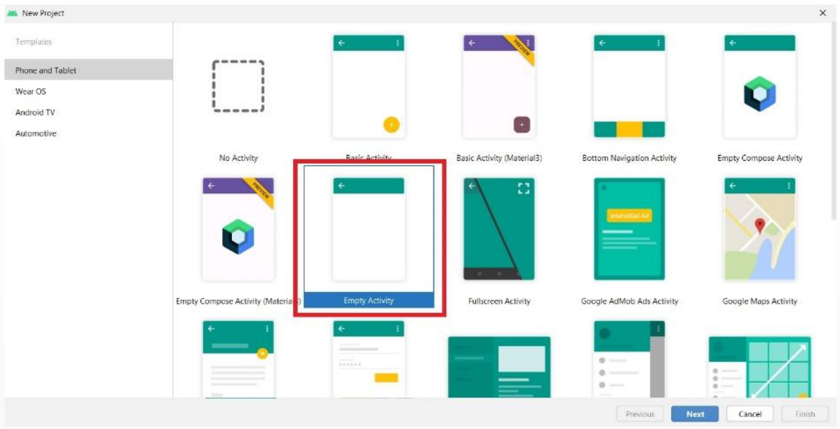


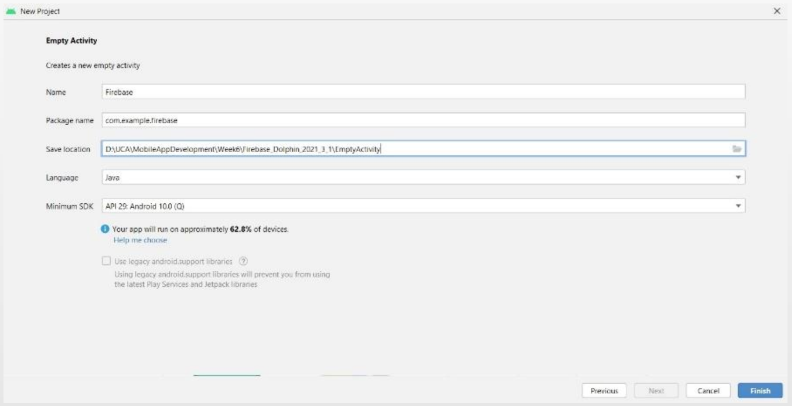
Adding Firebase Realtime Database to Android app

- If we create a new empty Android project:



Welcome to Android Studio
Android Studio
Dolphin | 2021.3.1
Projects
Customize
Plugins
Learn Android Studio
Welcome to Android Studio
Create a new project to start from scratch.
Open existing project from disk or version control.
New Project
Open
Get from VCS
More Actions




New Project

Empty Activity

Creates a new empty activity

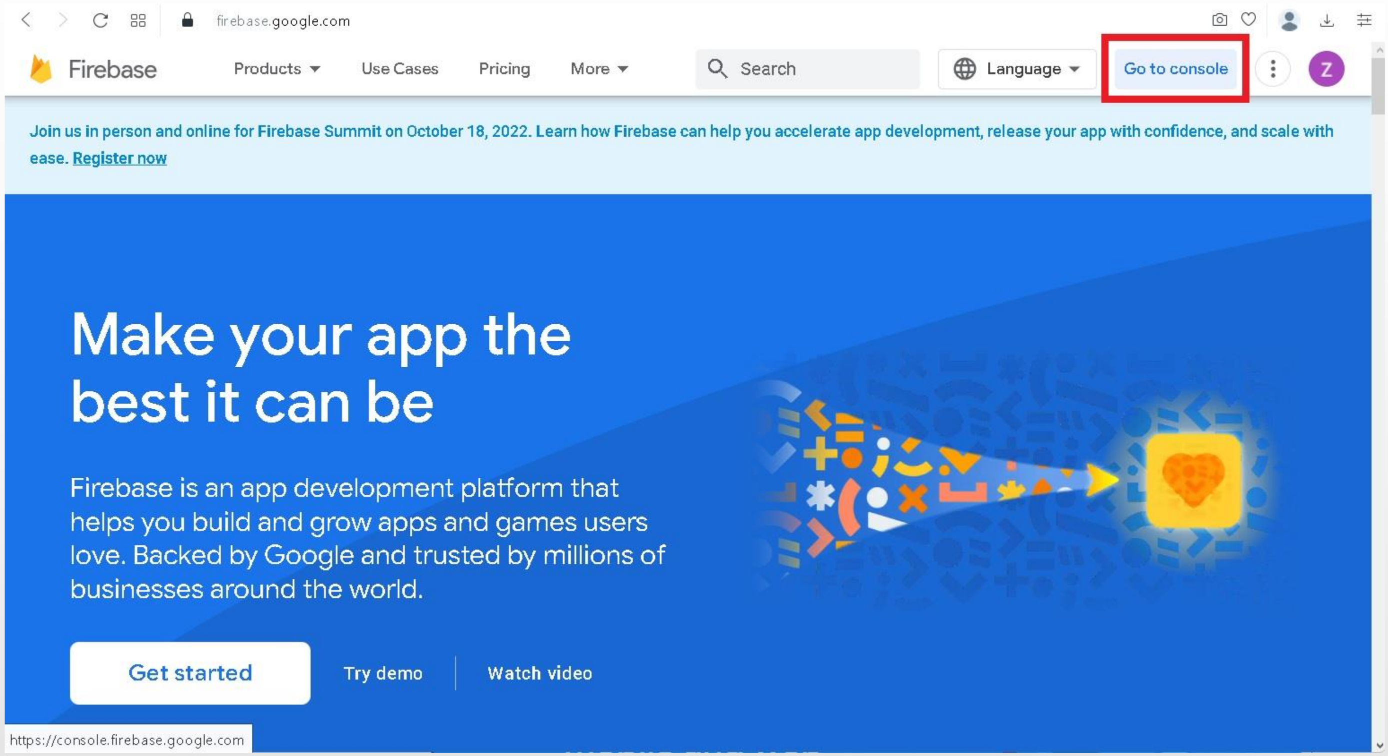
Name	Firebase
Package name	com.example.firebase
Save location	D:\UCA\MobileAppDevelopment\Week6\Firebase_Dolphin_2021_3_1\EmptyActivity
Language	Java
Minimum SDK	API 29: Android 10.0 (Q)

Your app will run on approximately **62.8%** of devices.
Help me choose

☐ Use legacy android.support libraries
Using legacy android.support libraries will prevent you from using the latest Play Services and Jetpack libraries

Adding Firebase Realtime Database to Android app

Step 3. Open the Firebase website and login into it (use the same email id as used in Android Studio for login)
Step 4. After login, click on the "Go To Console" button that is present of the upper right side of the website



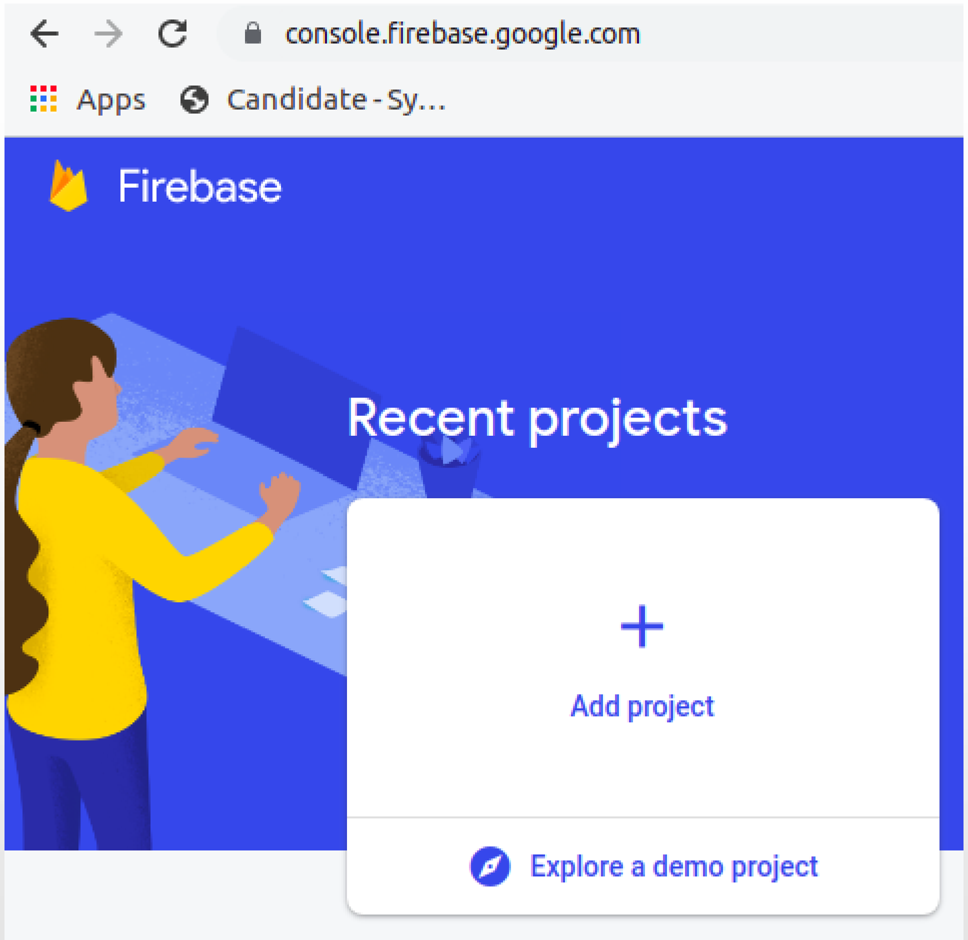
Join us in person and online for Firebase Summit on October 18, 2022. Learn how Firebase can help you accelerate app development, release your app with confidence, and scale with ease. Register now

Make your app the best it can be

Firebase is an app development platform that helps you build and grow apps and games users love. Backed by Google and trusted by millions of businesses around the world.

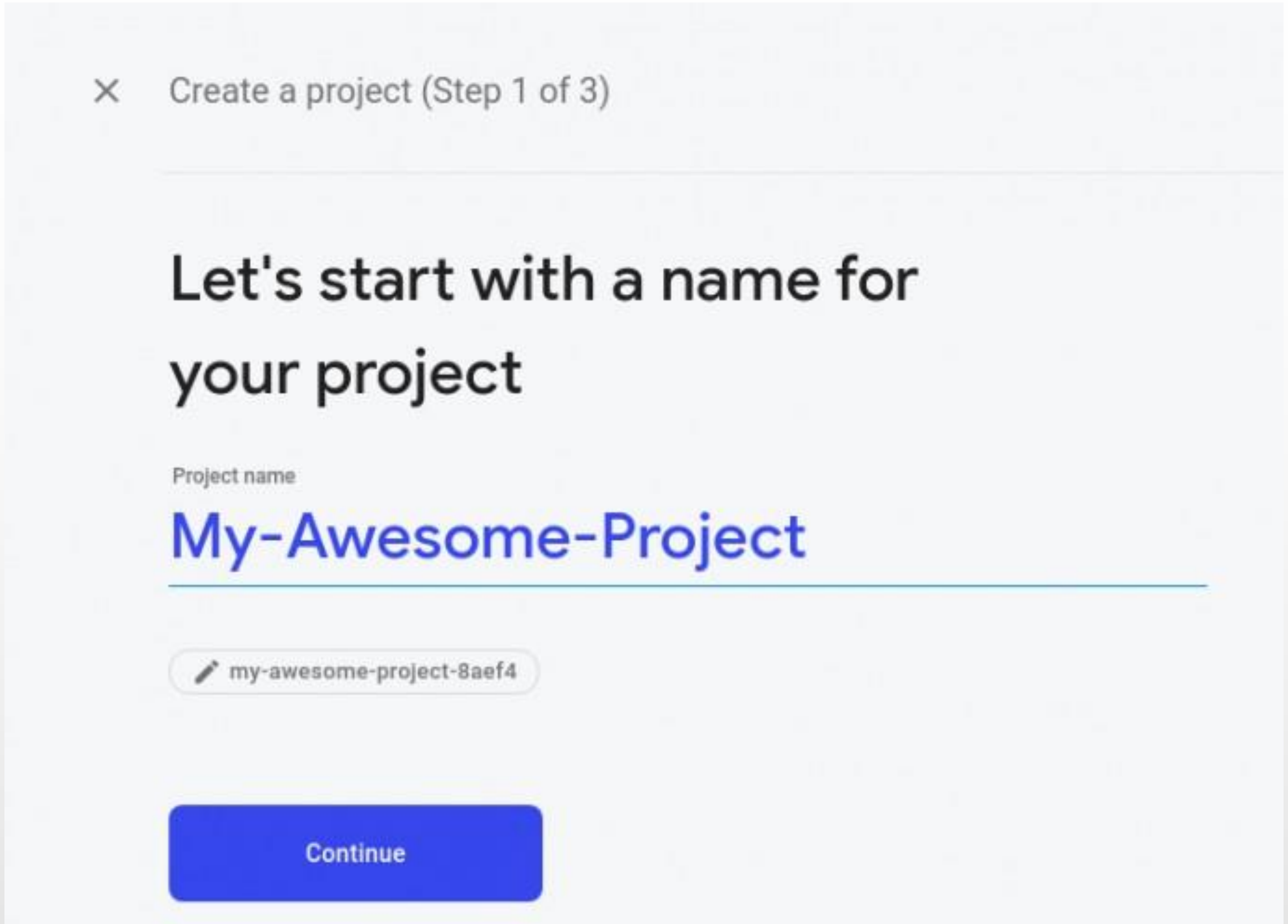
Adding Firebase Realtime Database to Android app

Step 5. Click on **"Add Project"**



Adding Firebase Realtime Database to Android app

Step 6. Enter the required details of the project and click on submit


Create a project (Step 1 of 3)

Let's start with a name for your project

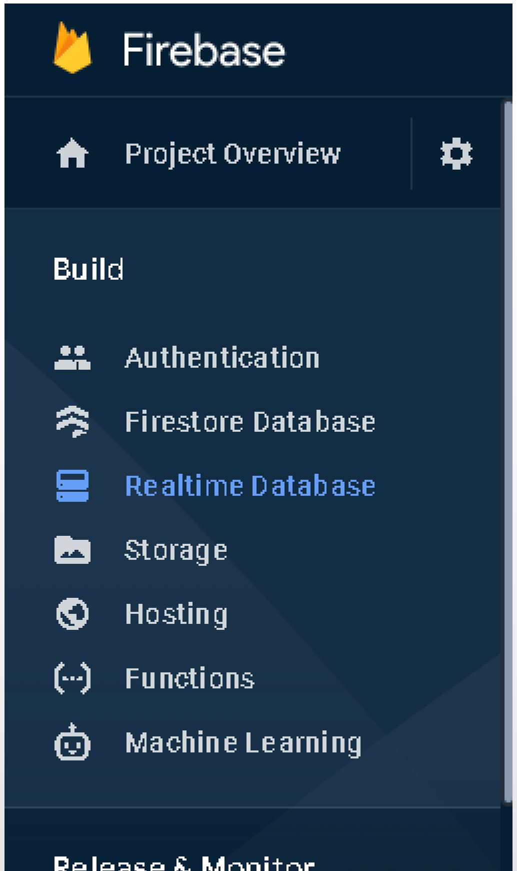
Project name

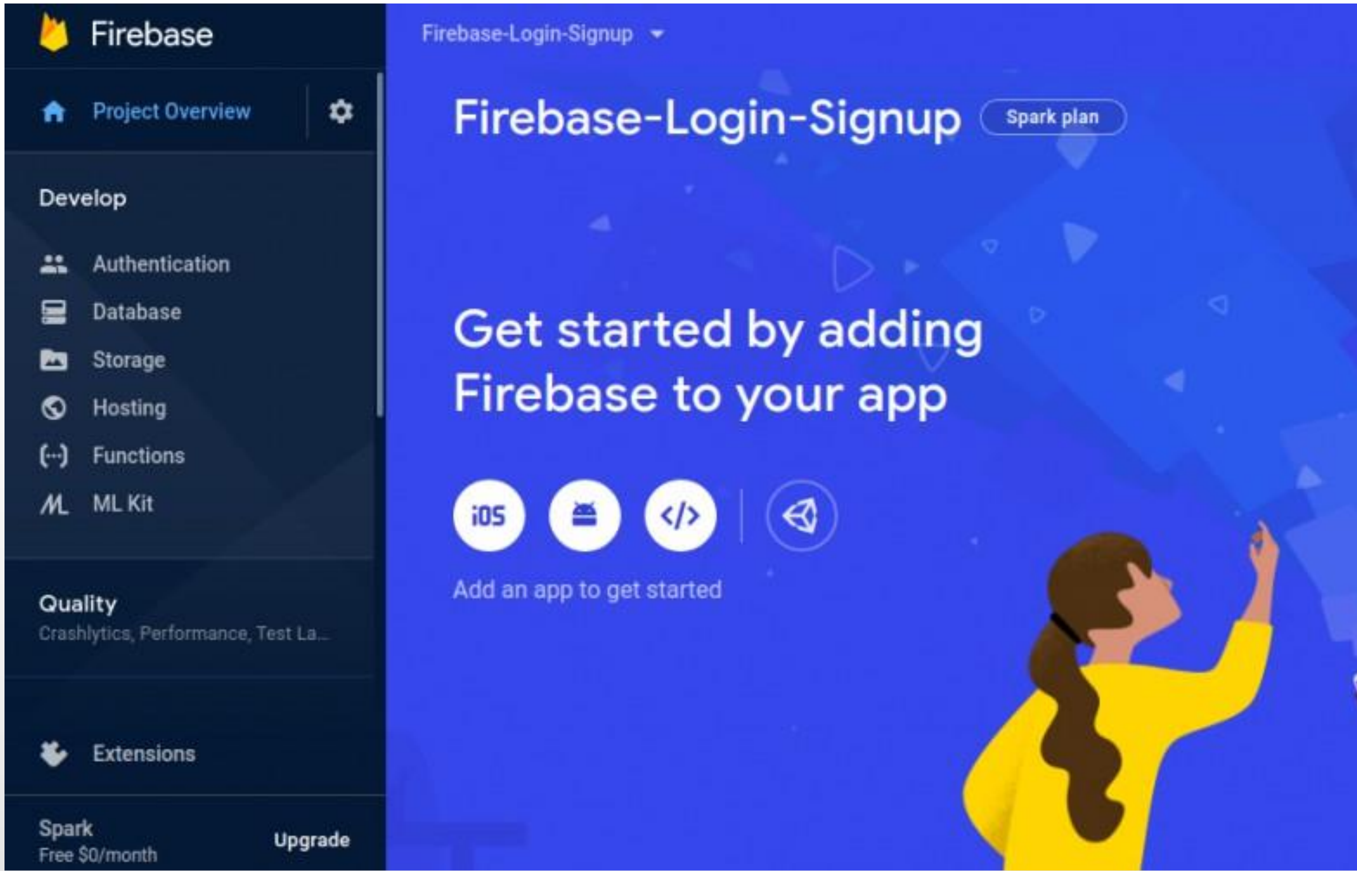
My-Awesome-Project

my-awesome-project-8aef4

Adding Firebase Realtime Database to Android app

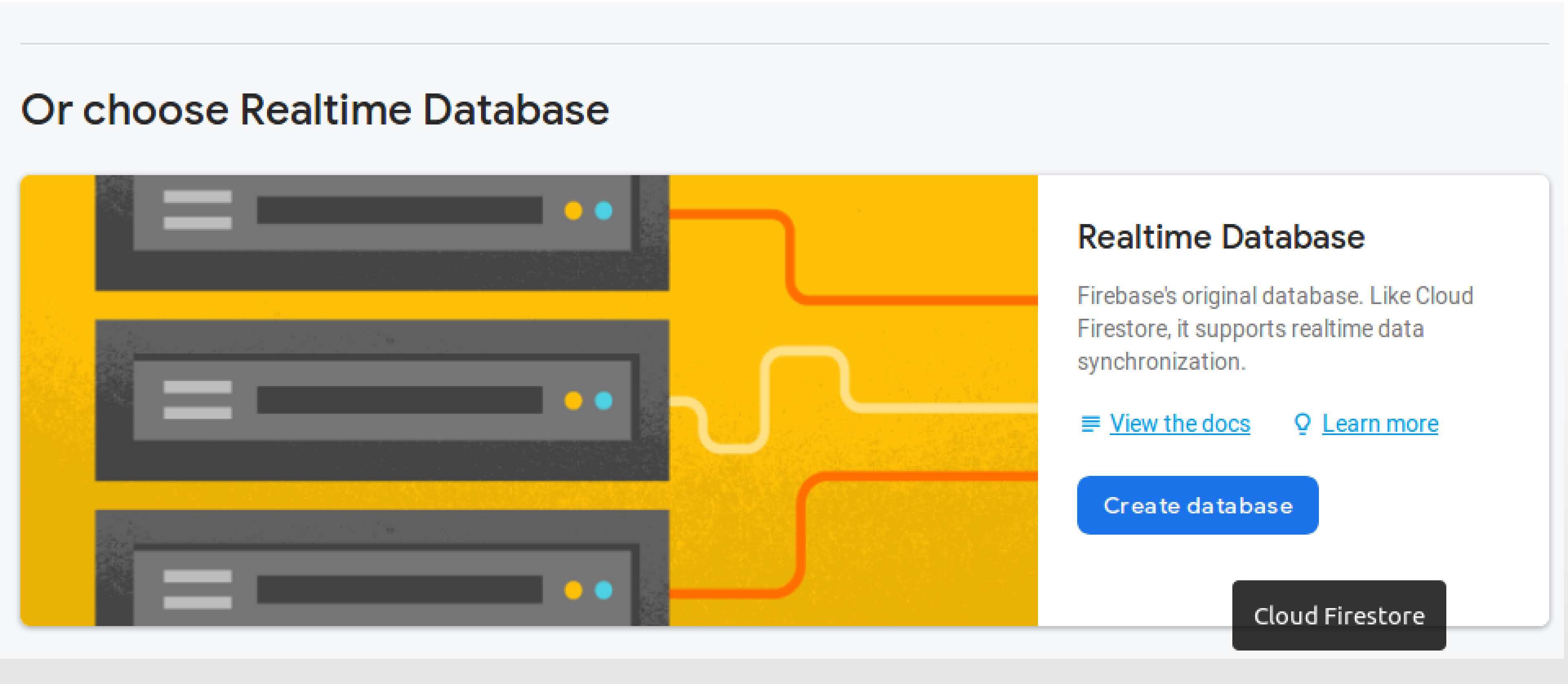
Step 7. After creating a project, we will see the below image (or something similar :) of our project dashboard





Adding Firebase Realtime Database to Android app

Step 8. Click on *"Database"* and then in the Realtime Database section, click on *"Create Database"*



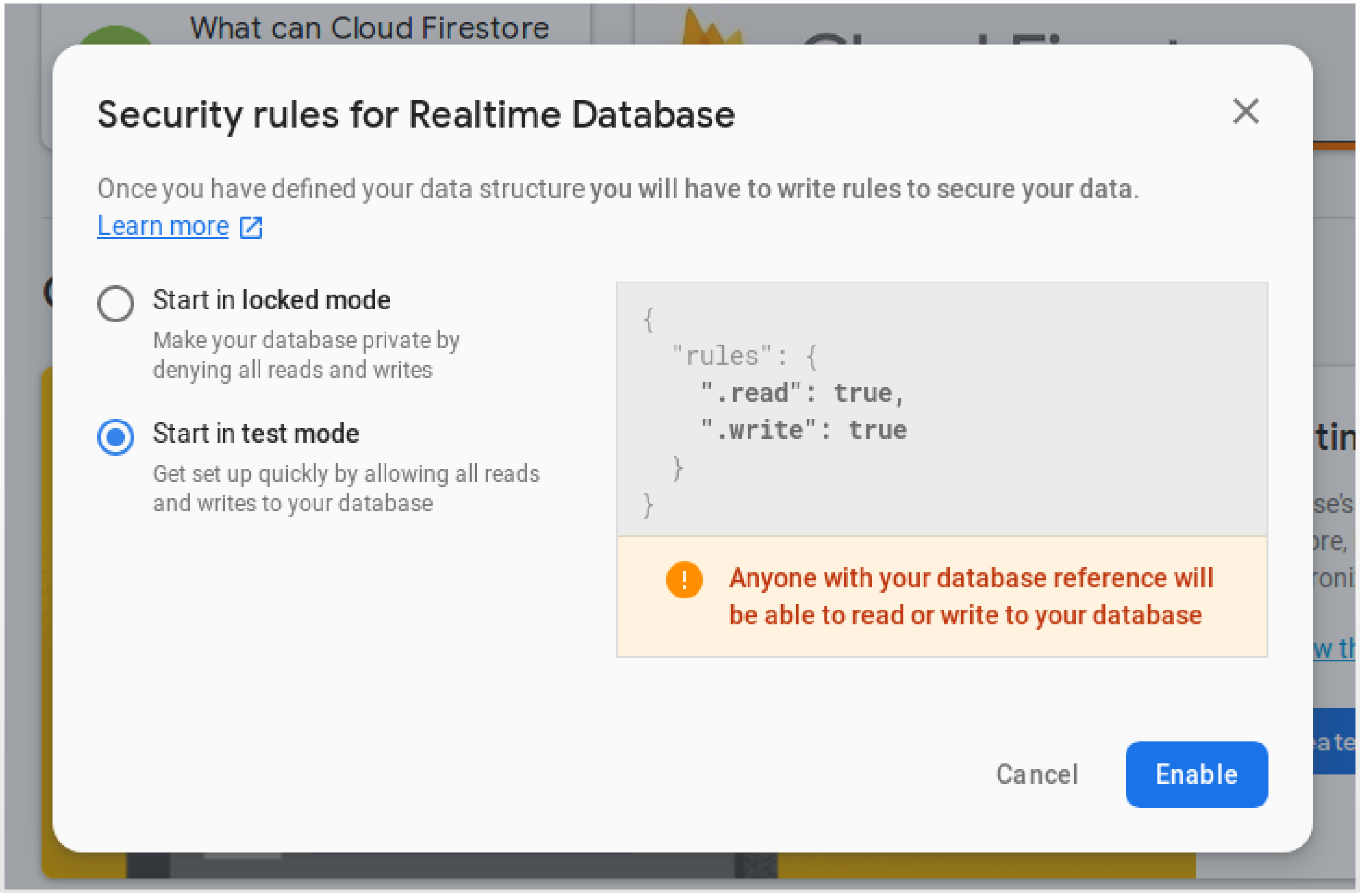
Or choose Realtime Database

Realtime Database

Firebase's original database. Like Cloud Firestore, it supports realtime data synchronization.

Adding Firebase Realtime Database to Android app

Step 9. We use the database for the educational purpose. Hence, select the "*Start in test mode*" option and click on enable.



Security rules for Realtime Database

Once you have defined your data structure you will have to write rules to secure your data.
Learn more

Start in locked mode
Make your database private by denying all reads and writes

[✓ Start in test mode]
Get set up quickly by allowing all reads and writes to your database

```
{
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

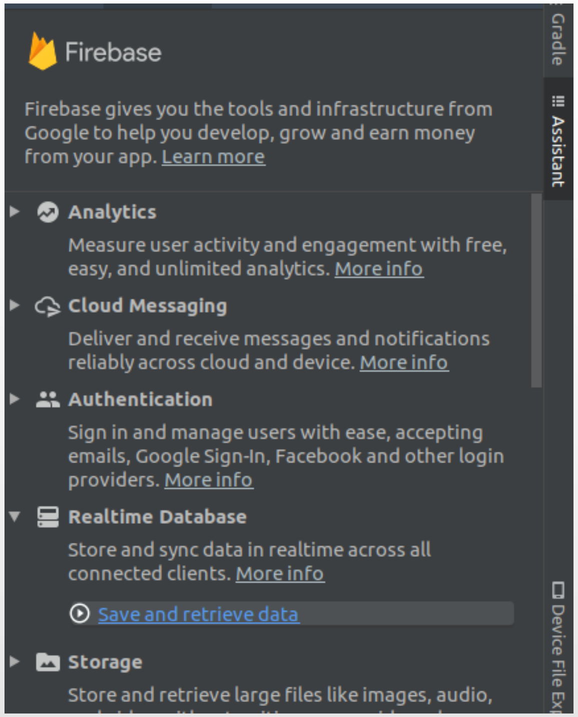
Anyone with your database reference will be able to read or write to your database

Adding Firebase Realtime Database to Android app

Step 10. Now, come back to our Android Studio project. We have to connect our Firebase project with the Android Studio project. So, click on **Tools** > **Firebase** > **Realtime Database** > **Save and retrieve data**

OR

"Get started with Realtime Database"



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

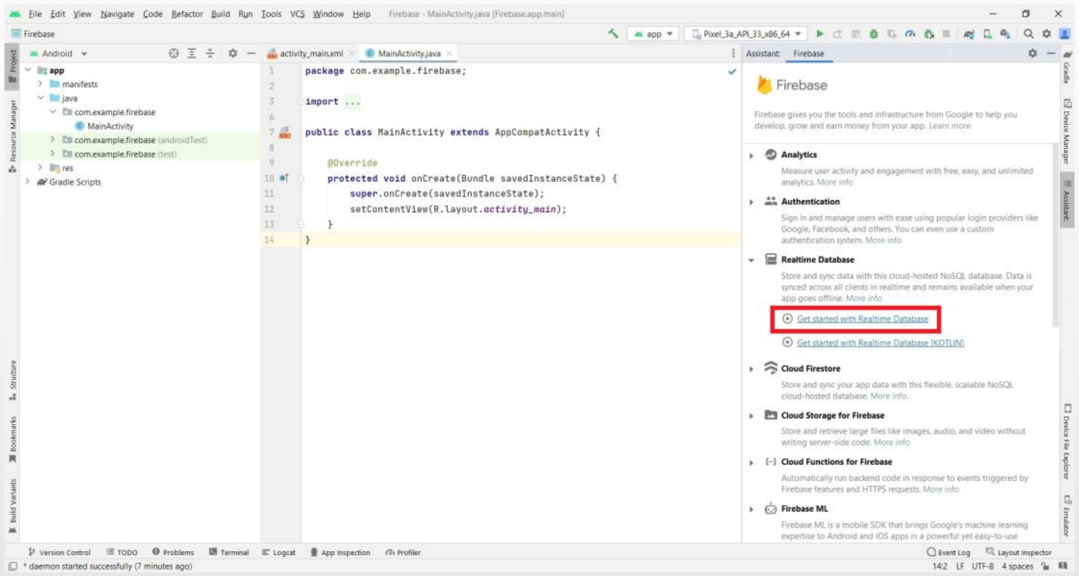
Cloud Messaging
Deliver and receive messages and notifications reliably across cloud and device. More info

Authentication
Sign in and manage users with ease, accepting emails, Google Sign-In, Facebook and other login providers. More info

Realtime Database
Store and sync data in realtime across all connected clients. More info

Save and retrieve data

Storage
Store and retrieve large files like images, audio,



```java
package com.example.firebase;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

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

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

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

Get started with Realtime Database

Get started with Realtime Database [KOTLIN]

Cloud Firestore
Store and sync your app data with this flexible, scalable NoSQL cloud-hosted database. More info.

Cloud Storage for Firebase
Store and retrieve large files like images, audio, and video without writing server-side code. More info

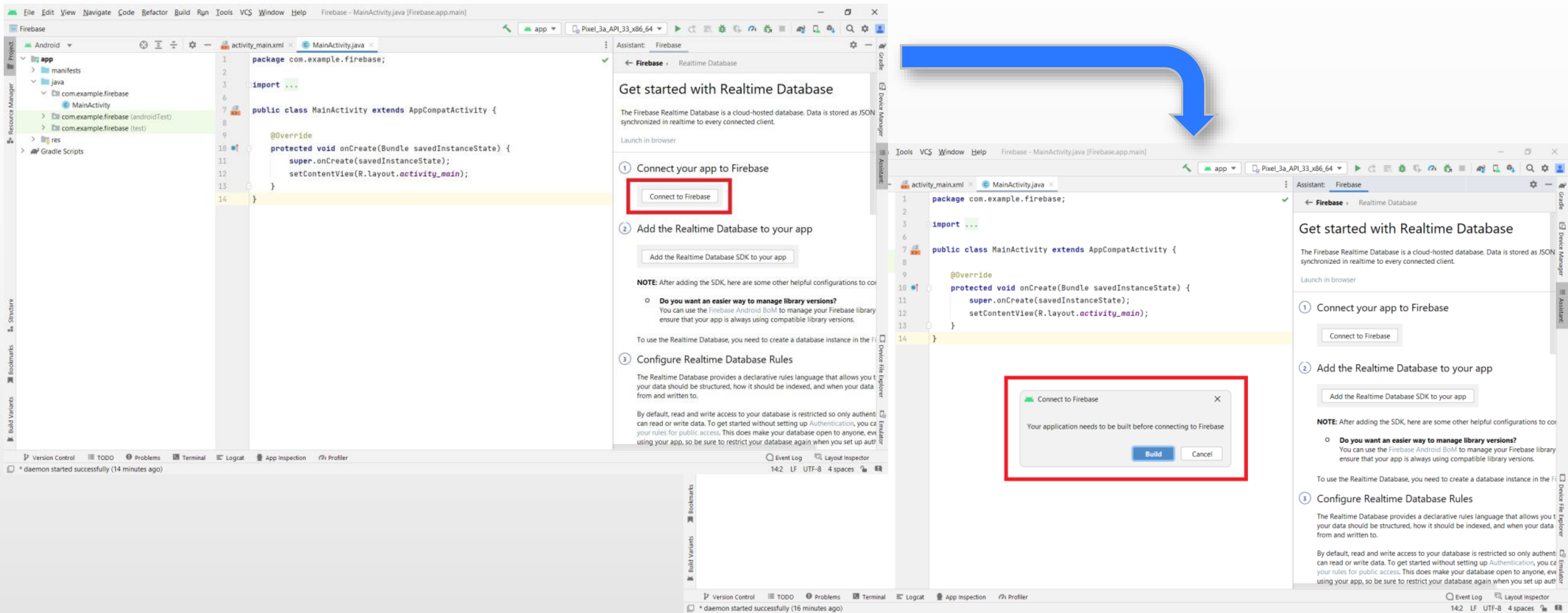
Cloud Functions for Firebase
Automatically run backend code in response to events triggered by Firebase features and HTTPS requests. More info

Firebase ML
Firebase ML is a mobile SDK that brings Google's machine learning expertise to Android and iOS apps in a powerful yet easy-to-use

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Firebase - MainActivity.java [Firebase.app.main]
app
Pixel_3a_API_33_x86_64
Android
app
manifests
java
com.example.firebase
MainActivity
com.example.firebase (androidTest)
com.example.firebase (test)
res
Gradle Scripts
activity_main.xml
MainActivity.java
Assistant: Firebase
Event Log
Layout Inspector
14:2 LF UTF-8 4 spaces
daemon started successfully (7 minutes ago)

Adding Firebase Realtime Database to Android app

Step 11. After that click on "*Connect to Firebase*". A list of projects will be shown to you. Select the project that you have created on the Firebase website and click on "*Connect to Firebase*".



```java
package com.example.firebase;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Get started with Realtime Database

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON synchronized in realtime to every connected client.

Launch in browser

1. Connect your app to Firebase

   Connect to Firebase

2. Add the Realtime Database to your app

   Add the Realtime Database SDK to your app

   NOTE: After adding the SDK, here are some other helpful configurations to co

   - **Do you want an easier way to manage library versions?**
     You can use the Firebase Android BoM to manage your Firebase library ensure that your app is always using compatible library versions.

   To use the Realtime Database, you need to create a database instance in the Fi

3. Configure Realtime Database Rules

   The Realtime Database provides a declarative rules language that allows you t your data should be structured, how it should be indexed, and when your data from and written to.

   By default, read and write access to your database is restricted so only authenti can read or write data. To get started without setting up Authentication, you ca your rules for public access. This does make your database open to anyone, eve using your app, so be sure to restrict your database again when you set up auth

Connect to Firebase

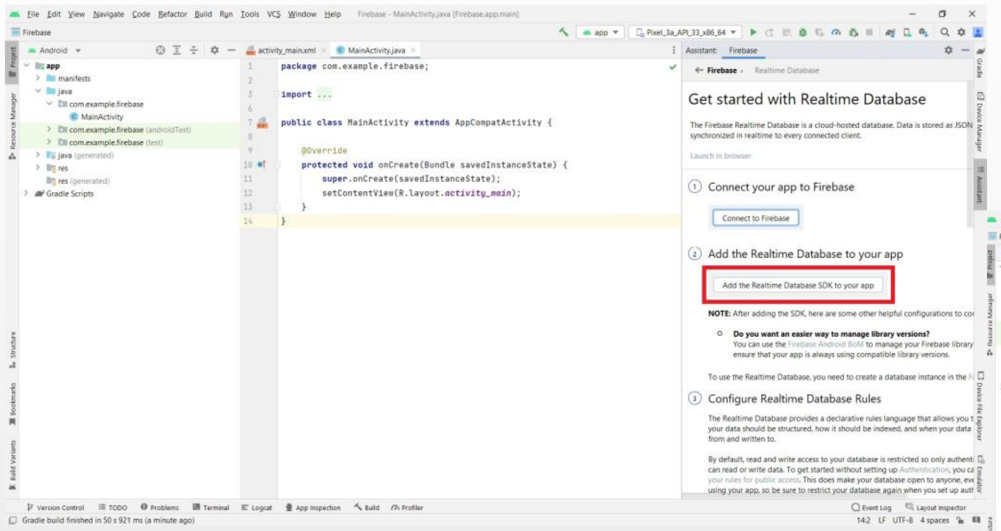
Your application needs to be built before connecting to Firebase

Build   Cancel

* daemon started successfully (14 minutes ago)

Adding Firebase Realtime Database to Android app

Step 12. Lastly, we have to add the dependency of Firebase Realtime Database in our project by clicking on "*Add the Realtime Database SDK to your app*" button and then "*Accept changes*".



```java
package com.example.firebase;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Get started with Realtime Database

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON synchronized in realtime to every connected client.

Launch in browser

1. Connect your app to Firebase

   Connect to Firebase

2. Add the Realtime Database to your app

   Add the Realtime Database SDK to your app

   NOTE: After adding the SDK, here are some other helpful configurations to co

   - **Do you want an easier way to manage library versions?**
     You can use the Firebase Android BoM to manage your Firebase library
     ensure that your app is always using compatible library versions.

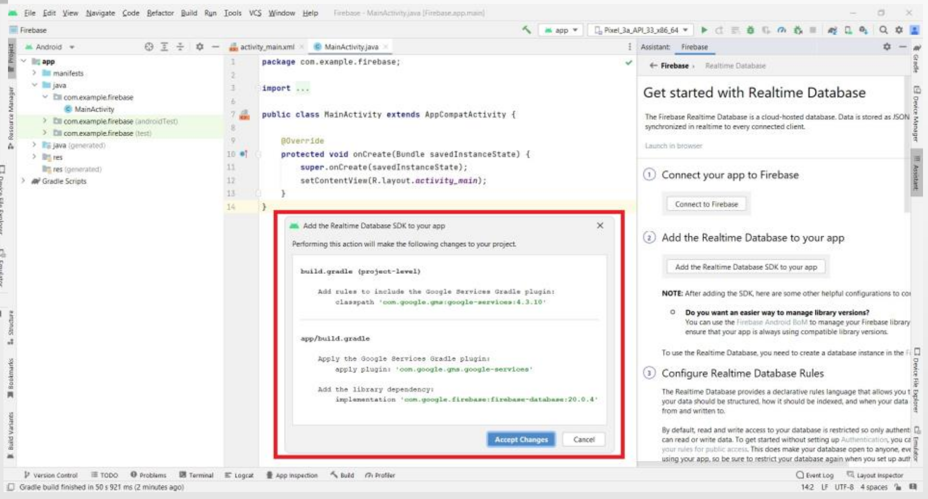
   To use the Realtime Database, you need to create a database instance in the F

3. Configure Realtime Database Rules

   The Realtime Database provides a declarative rules language that allows you t your data should be structured, how it should be indexed, and when your data from and written to.

   By default, read and write access to your database is restricted so only authent can read or write data. To get started without setting up Authentication, you ca your rules for public access. This does make your database open to anyone, eve using your app, so be sure to restrict your database again when you set up auth

Gradle build finished in 50 s 921 ms (a minute ago)



Add the Realtime Database SDK to your app

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

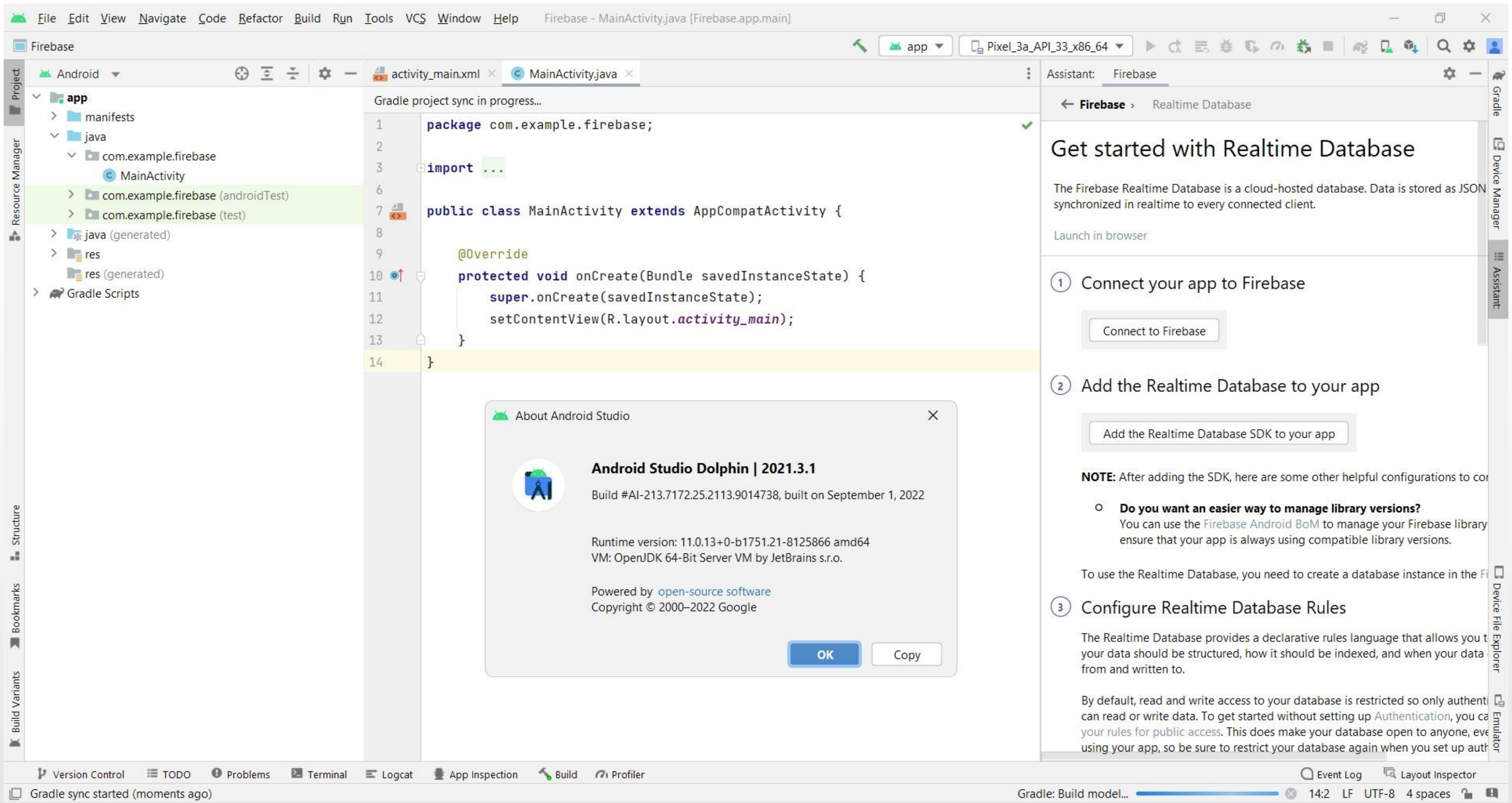
```
build.gradle (project-level)

    Add rules to include the Google Services Gradle plugin:
        classpath 'com.google.gms:google-services:4.3.10'

app/build.gradle

    Apply the Google Services Gradle plugin:
        apply plugin: 'com.google.gms.google-services'

    Add the library dependency:
        implementation 'com.google.firebase:firebase-database:20.0.4'
```

Accept Changes | Cancel

Gradle build finished in 50 s 921 ms (2 minutes ago)

Adding Firebase Realtime Database to Android app

- To develop the app, we use Android Studio Dolphin 2021.3.1



Gradle project sync in progress...

```
package com.example.firebase;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

About Android Studio

Android Studio Dolphin | 2021.3.1

Build #AI-213.7172.25.2113.9014738, built on September 1, 2022

Runtime version: 11.0.13+0-b1751.21-8125866 amd64
VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o.

Powered by open-source software
Copyright © 2000–2022 Google

OK Copy

Get started with Realtime Database

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON synchronized in realtime to every connected client.

Launch in browser

1. Connect your app to Firebase

   Connect to Firebase

2. Add the Realtime Database to your app

   Add the Realtime Database SDK to your app

   NOTE: After adding the SDK, here are some other helpful configurations to co

   - **Do you want an easier way to manage library versions?**
     You can use the Firebase Android BoM to manage your Firebase library ensure that your app is always using compatible library versions.

   To use the Realtime Database, you need to create a database instance in the Fi

3. Configure Realtime Database Rules

   The Realtime Database provides a declarative rules language that allows you t your data should be structured, how it should be indexed, and when your data from and written to.

   By default, read and write access to your database is restricted so only authenti can read or write data. To get started without setting up Authentication, you ca your rules for public access. This does make your database open to anyone, eve using your app, so be sure to restrict your database again when you set up auth

Gradle sync started (moments ago)   Gradle: Build model...   14:2 LF UTF-8 4 spaces

Adding Firebase Realtime Database to Android app

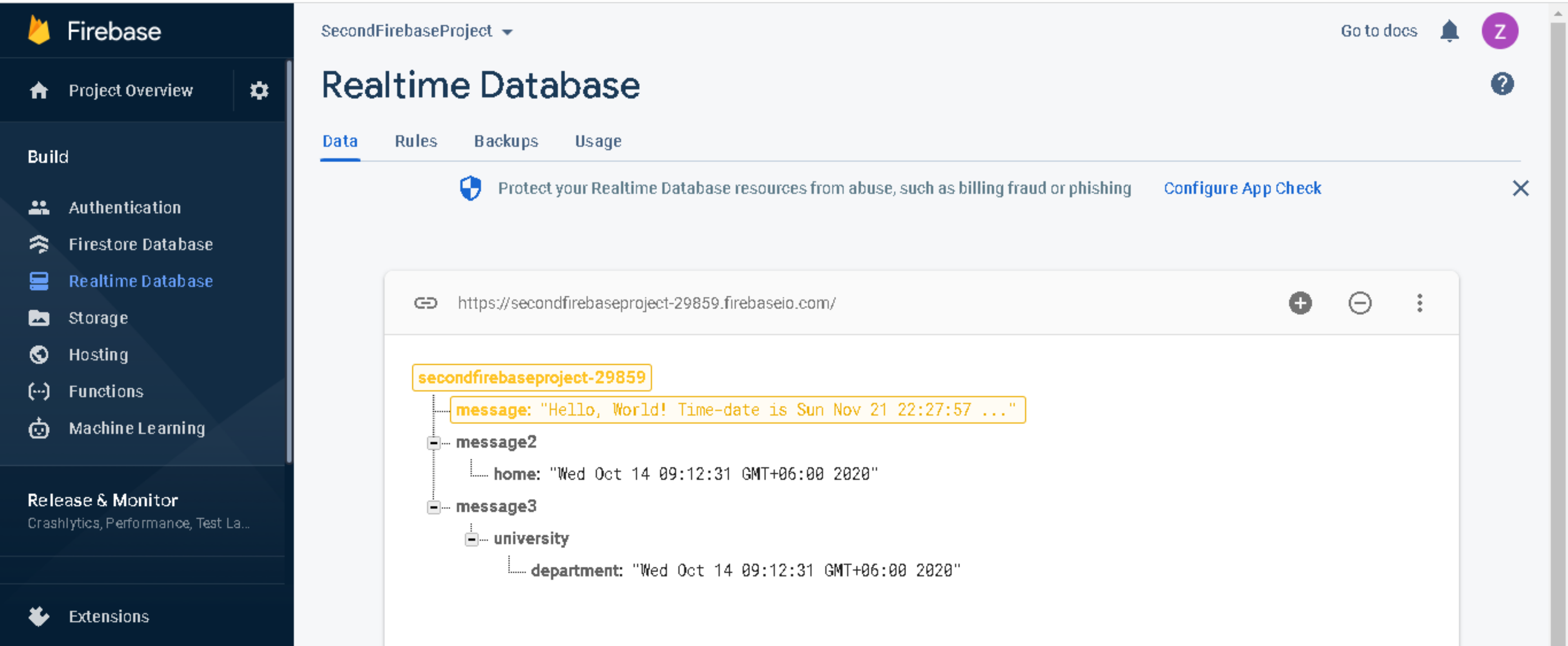
- How to write a database
  - Here, *getInstance*() is used to get the complete instance of the Database. Then, with the help of that instance, we store the data at the particular location as shown in the below code:

```
public void WriteInfotoFirebase (View view){
        // Write a message to the database
        // Get the database instance and store into object
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        // getReference() gets the reference if the reference is already created...
        // if reference is not created then it will create a new reference here
        DatabaseReference myRef = database.getReference("message");
        // assign value to the particular reference
        myRef.setValue("Hello, World! Time-date is "+
Calendar.getInstance().getTime());
    }
```

https://firebasetutorials.com/create-firebase-realtime-database/

Adding Firebase Realtime Database to Android app

- The Database reference looks like in the Firebase console in the below image (the yellow color means that the node was updated):

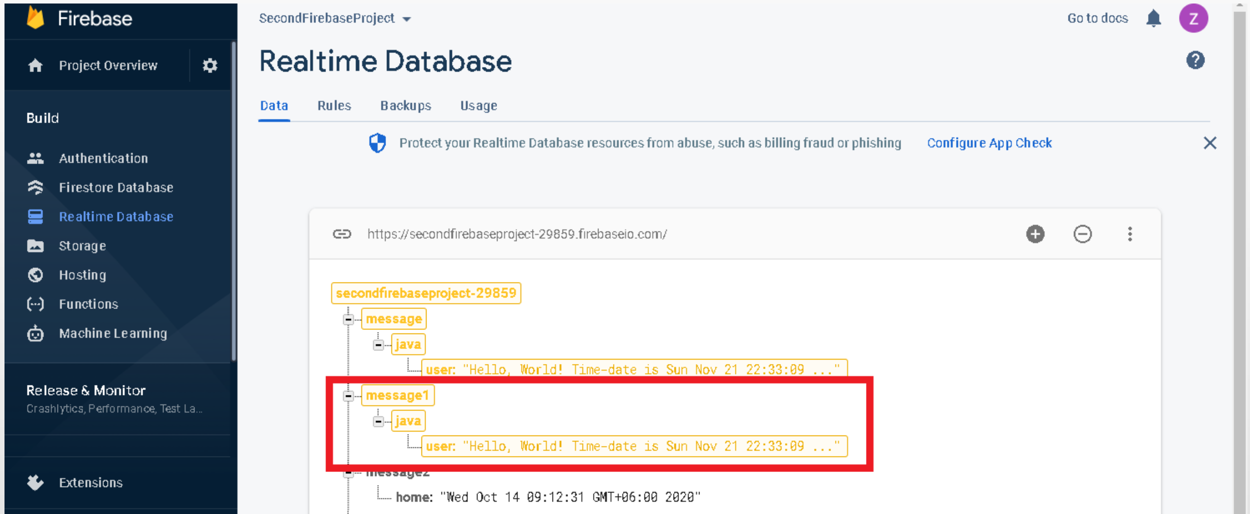


```
https://secondfirebaseproject-29859.firebaseio.com/

secondfirebaseproject-29859
├── message: "Hello, World! Time-date is Sun Nov 21 22:27:57 ..."
├── message2
│   └── home: "Wed Oct 14 09:12:31 GMT+06:00 2020"
└── message3
    └── university
        └── department: "Wed Oct 14 09:12:31 GMT+06:00 2020"
```

Adding Firebase Realtime Database to Android app

- If we want to create a multi-node database, we can set the path in the *getReference*() as follows (we can also use *.child()* to get the same results):

```
public void WriteInfotoFirebaseMultiNode (View view){
        // Write a message to the database
        // Write a message to the database
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        // Set path to get the multiple dropdown nodes
        DatabaseReference myRef = database.getReference("message/java/user");
        myRef.setValue("Hello, World! Time-date is "+ Calendar.getInstance().getTime());
        //We can also use .child() to get the same results
        myRef = database.getReference("message1").child("java").child("user");
        myRef.setValue("Hello, World! Time-date is "+ Calendar.getInstance().getTime());
    }
```

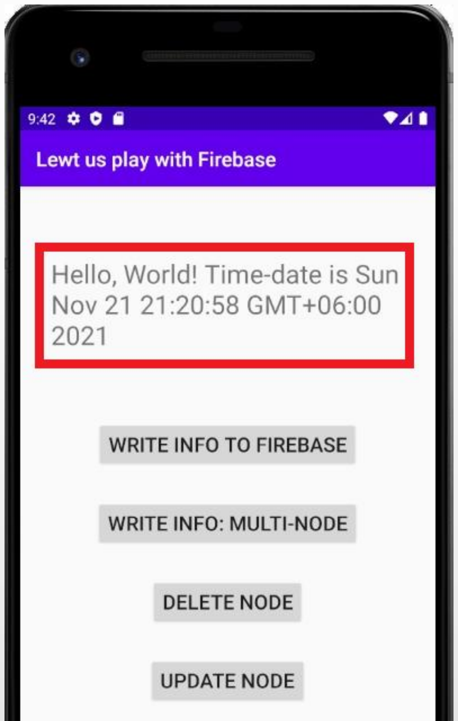


```
https://secondfirebaseproject-29859.firebaseio.com/
secondfirebaseproject-29859
  message
    java
      user: "Hello, World! Time-date is Sun Nov 21 22:33:09 ..."
  message1
    java
      user: "Hello, World! Time-date is Sun Nov 21 22:33:09 ..."
  message2
    home: "Wed Oct 14 09:12:31 GMT+06:00 2020"
```

Adding Firebase Realtime Database to Android app

- How to read database
  - To update our app data in the real-time, we have to attach a *ValueEventListener* to the object of the reference we have created above.
  - *onDataChange*() method is called once when this method is attached to the listener, whenever a data is changed, including their children's *onDataChange*() method triggered again.

```
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    // getReference() gets the reference if the reference is already created...
    // if reference is not created then it will create a new reference here
    DatabaseReference myRef = database.getReference("message");
    private static final String TAG=MainActivity.class.getSimpleName();
    TextView textView;

@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textView = (TextView) findViewById(R.id.textViewID);

        // Read from the database
        myRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
// This method is called once with the initial value and again
// whenever data at this location is updated
                try{
                    String value = dataSnapshot.getValue(String.class);
                    Log.i(TAG, "Value is: " + value);
                    textView.setText(value);
                } catch (Exception exception){}
            }
            @Override
            public void onCancelled(DatabaseError error) {
                // Failed to read value
                Log.i(TAG, "Failed to read value.", error.toException());
            }
        });
    }
```



Hello, World! Time-date is Sun Nov 21 21:20:58 GMT+06:00 2021

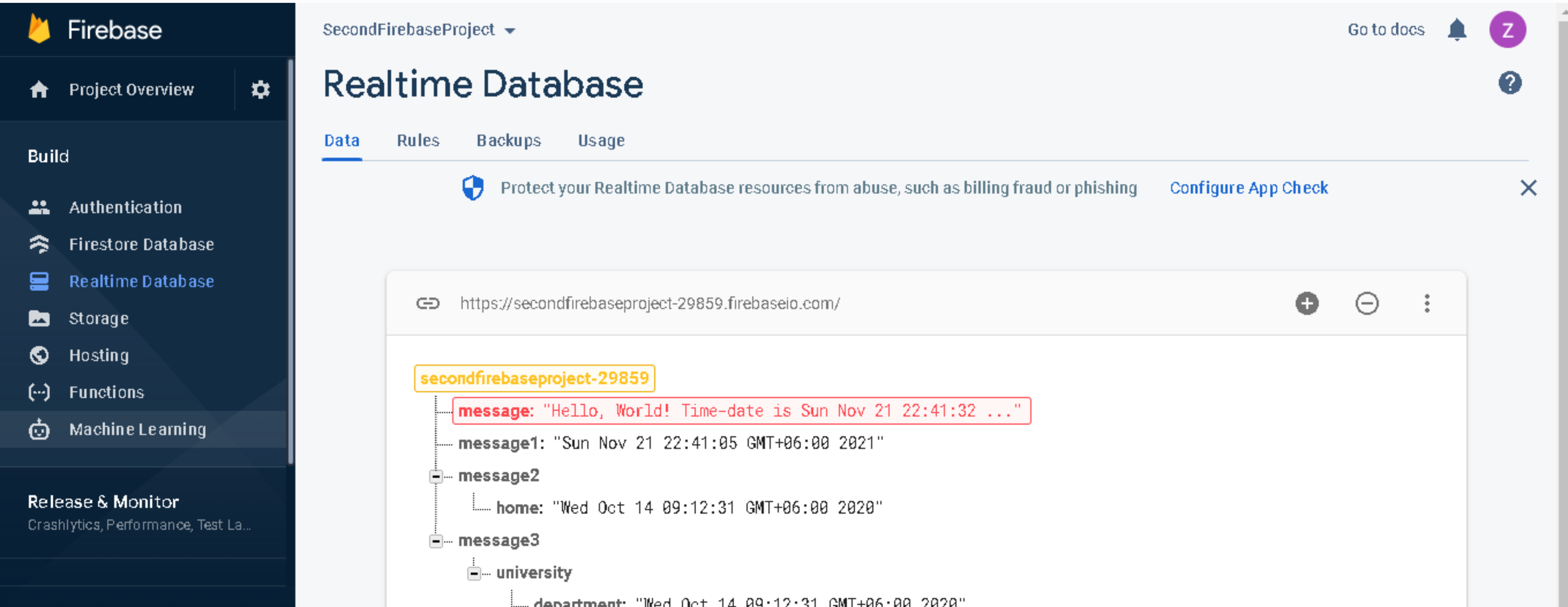
Adding Firebase Realtime Database to Android app

- How to delete node from database
  - We can delete the node by putting the reference path and setting the value to *null* or by attaching *removeValue*() to it

```
public void DeleteNode (View view){
        DatabaseReference dbNode =

FirebaseDatabase.getInstance().getReference().getRoot().child("message1").child("java").
child("user");
        dbNode.setValue(null);
        //or we can delete the node with the following code too
        DatabaseReference dbNodetwo =
                FirebaseDatabase.getInstance().getReference().getRoot()
                        .child("message");
        dbNodetwo.removeValue();
    }
```

Adding Firebase Realtime Database to Android app

◦ When we delete the node, it will instantly update in the real-time database. In the image below, we can see that when we delete the node it will turn red and then remove from the real-time database console



Protect your Realtime Database resources from abuse, such as billing fraud or phishing

https://secondfirebaseproject-29859.firebaseio.com/

```
secondfirebaseproject-29859
  message: "Hello, World! Time-date is Sun Nov 21 22:41:32 ..."
  message1: "Sun Nov 21 22:41:05 GMT+06:00 2021"
  message2
    home: "Wed Oct 14 09:12:31 GMT+06:00 2020"
  message3
    university
      department: "Wed Oct 14 09:12:31 GMT+06:00 2020"
```

Adding Firebase Realtime Database to Android app

- How to update node in the database
  - *.setValue("String");* is used to set the value in the database
  - Also, we can attach the listener just to know whether the operation is successful or not

```
 public void UpdateNode (View view){
        DatabaseReference reference = database.getReference().child("message");
        reference.setValue("Hello, World! Time-date is "+ Calendar.getInstance().getTime())
// we can also attach listener just to know whether it is successful or not
                .addOnSuccessListener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void aVoid) {
                        // Write was successful!
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        // Write failed
                    }
                });
    }
```

Adding Firebase Realtime Database to Android app

- if we need to use Google Services, we have to add the following lines of code inside the build.gradle (Project) file:

```
plugins {
    id 'com.android.application' version '7.2.0' apply false
    id 'com.android.library' version '7.2.0' apply false
    id 'com.google.gms.google-services' version '4.3.10' apply false
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

and be sure that we have the following plugin IDs inside the build.gradle (Module) file:

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

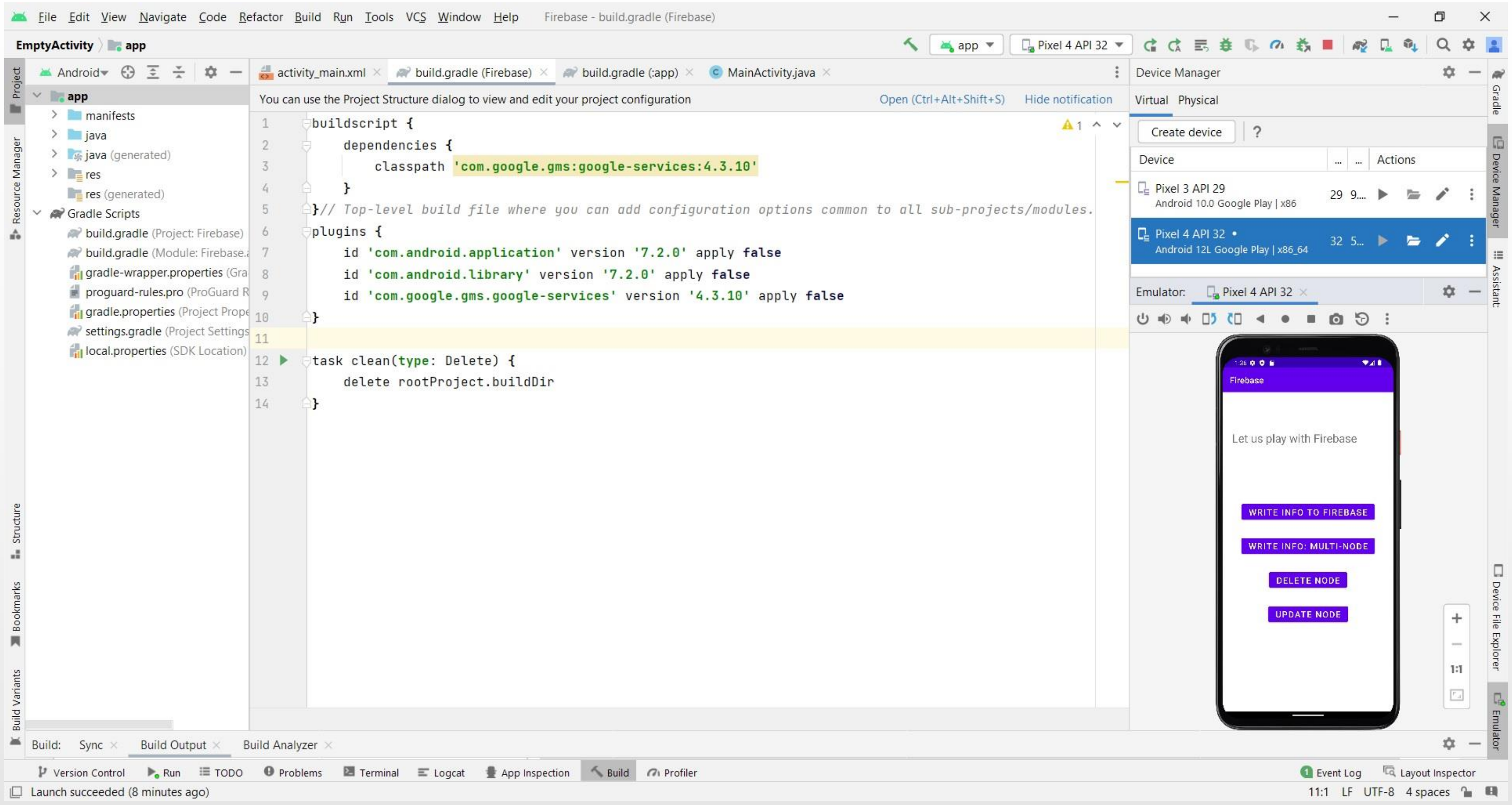
Adding Firebase Realtime Database to Android app

- activity_main.xml is as follows:

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".MainActivity">

  <TextView
    android:id="@+id/textViewID"
    android:layout_width="350dp"
    android:layout_height="133dp"
    android:text="Let us play with Firebase"
    android:textSize="26sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.146" />

  <Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:onClick="WriteInfotoFirebase"
    android:text="Write info to Firebase"
    android:textSize="20sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textViewID" />

  <Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:onClick="WriteInfotoFirebaseMultiNode"
    android:text="Write info: Multi-node"
    android:textSize="20sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button" />

  <Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:onClick="DeleteNode"
    android:text="Delete node"
    android:textSize="20sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button2" />

  <Button
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:onClick="UpdateNode"
    android:text="Update node"
    android:textSize="20sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button3" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Adding Firebase Realtime Database to Android app

- An example with smartphone Pixel 4 (API 32, Android 12, Google Play):



You can use the Project Structure dialog to view and edit your project configuration   Open (Ctrl+Alt+Shift+S)   Hide notification

```
buildscript {
    dependencies {
        classpath 'com.google.gms:google-services:4.3.10'
    }
}// Top-level build file where you can add configuration options common to all sub-projects/modules.
plugins {
    id 'com.android.application' version '7.2.0' apply false
    id 'com.android.library' version '7.2.0' apply false
    id 'com.google.gms.google-services' version '4.3.10' apply false
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

Device			Actions
Pixel 3 API 29 Android 10.0 Google Play | x86	29	9...	
Pixel 4 API 32 Android 12L Google Play | x86_64	32	5...	

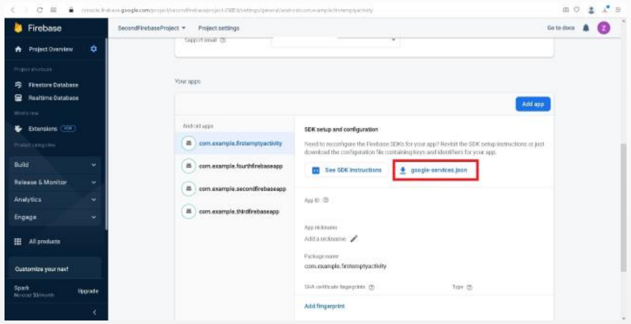
Launch succeeded (8 minutes ago)

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Firebase - build.gradle (Firebase)
EmptyActivity app
Android
app
manifests
java
java (generated)
res
res (generated)
Gradle Scripts
build.gradle (Project: Firebase)
build.gradle (Module: Firebase.
gradle-wrapper.properties
proguard-rules.pro
gradle.properties
settings.gradle
local.properties (SDK Location)
activity_main.xml
build.gradle (Firebase)
build.gradle (:app)
MainActivity.java
Pixel 4 API 32
Device Manager
Virtual Physical
Create device
Emulator: Pixel 4 API 32
Firebase
Let us play with Firebase
WRITE INFO TO FIREBASE
WRITE INFO: MULTI-NODE
DELETE NODE
UPDATE NODE
Build: Sync Build Output Build Analyzer
Version Control Run TODO Problems Terminal Logcat App Inspection Build Profiler
Event Log Layout Inspector
11:1 LF UTF-8 4 spaces

Adding Firebase Realtime Database to Android app

- *Perhaps* we need to add the file google-services.json manually: https://stackoverflow.com/questions/33866061/error-file-google-services-json-is-missing-from-module-root-folder-the-google/50173745

  1. Goto https://console.firebase.google.com/
  2. Select your project
  3. On the left menu, click on settings > project settings



SDK setup and configuration

Need to reconfigure the Firebase SDKs for your app? Revisit the SDK setup instructions or just download the configuration file containing keys and identifiers for your app.

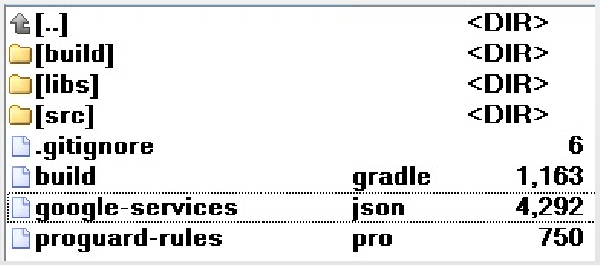
App ID

App nickname
Add a nickname

Package name
com.example.firstemptyactivity

SHA certificate fingerprints   Type

  4. Add an app or download the google-services.json file under the Your App section



```
[..]                                  <DIR>
[build]                               <DIR>
[libs]                                <DIR>
[src]                                 <DIR>
.gitignore                                6
build              gradle             1,163
google-services    json               4,292
proguard-rules     pro                  750
```

Do you have any questions or comments?

Thank you
for your attention !

In this presentation:
• Some icons were downloaded from flaticon.com and iconscout.com