Firebase Realtime Database

https://firebase google com

Jan Schoepp, Akshay Mehta and Dr Christelle Scharff

Updated 12/10/2019

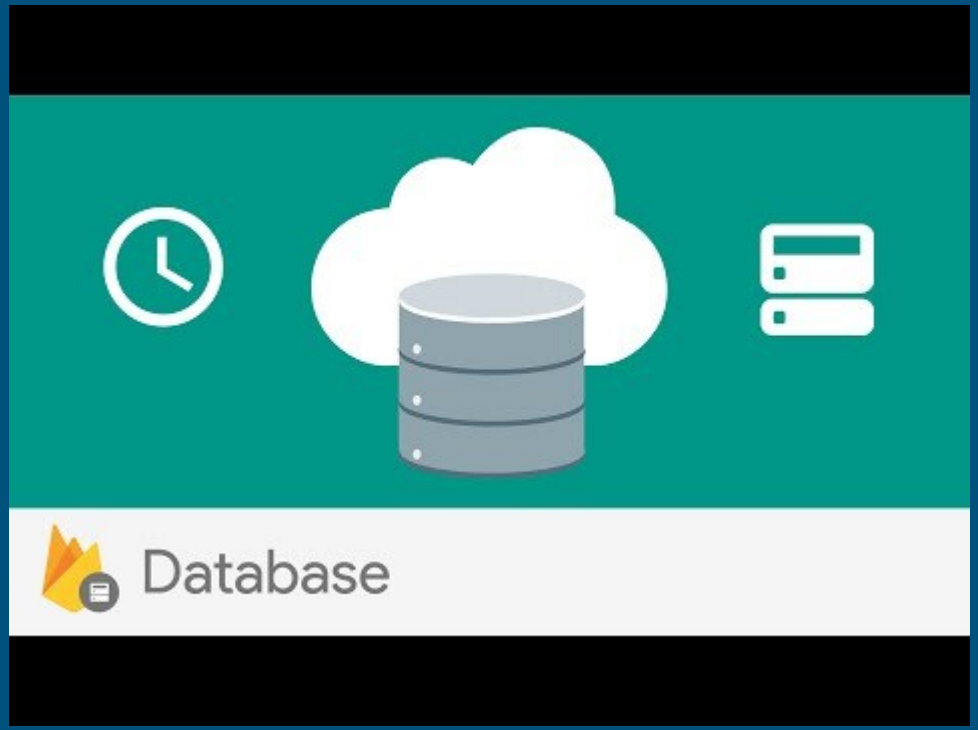
Brief introduction

- A realtime database
- Allows developers to store and sync data across multiple clients
- Cross-platform and real time - Automatically receives updates with the newest data
- Out-of-the-box authentication with Facebook, GitHub, Twitter, and Google
- Information is stored offline after downloading and updates once online
- Allows developers to quickly build a proof-of-concept

History

- Created by Andrew Lee and James Tamplin
- Evolved from their previous startup called Envolve in 2011
- Founded Firebase as a separate company in 2012
- Raised $7 0 million in funding during 2012-13
- Acquired by Google in 2014

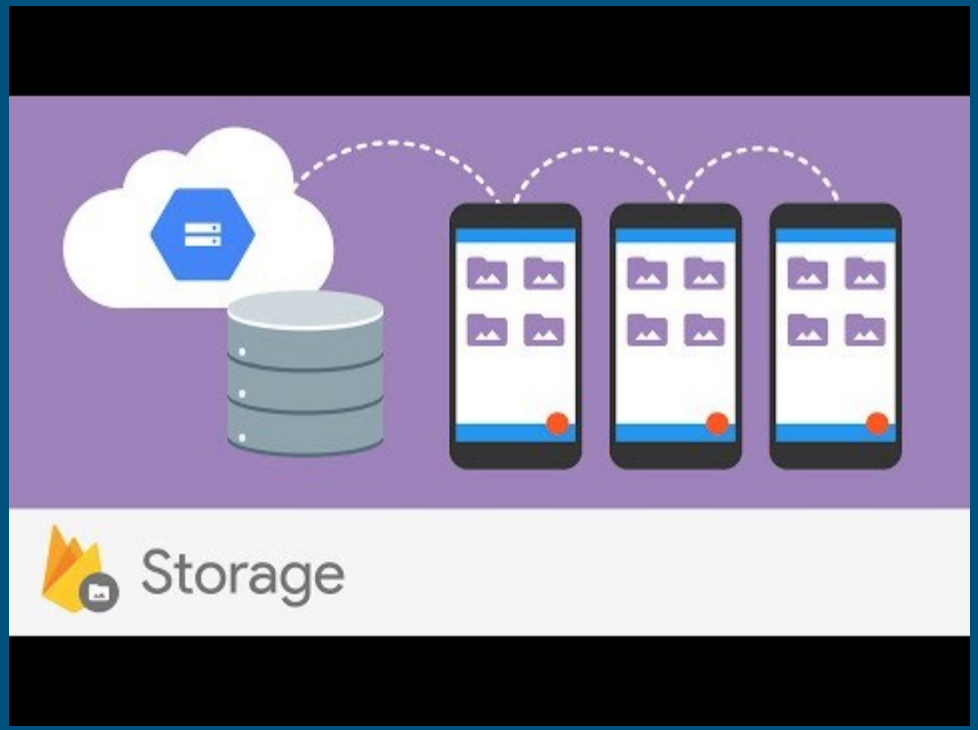
Firebase Realtime Database



https://www youtube com/watch?time_continue=2&v=U5aeM5dvUpA

- Firebase Realtime Database
  - Store and sync data with a NoSQL cloud database
  - Data is synced across all clients in realtime
  - Remains available when your app goes offline, and updates once online

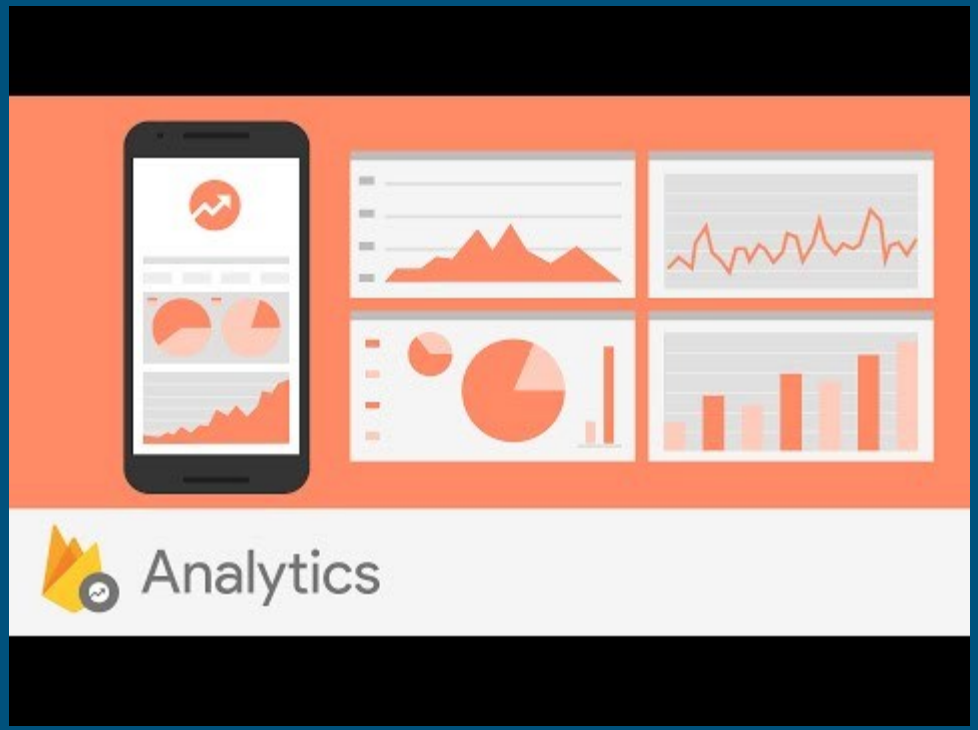
Firebase Storage



https://www youtube com/watch?time_continue=3&v=_tyjqozrEPY

- Firebase Storage
  - Secure file upload and download
  - Robust - Uploading and downloading restarts where they stopped
  - Scalable - Powered by Google Cloud Storage

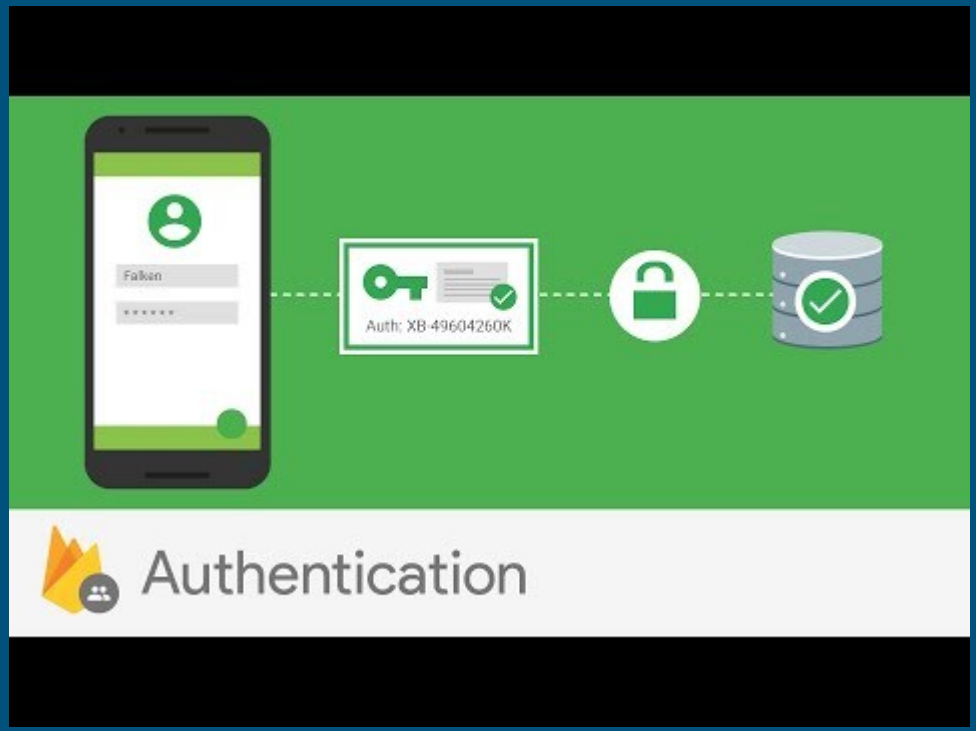
Firebase Analytics



https://www youtube com/watch?v=iT6EaIwtonY

- Firebase Analytics
  - Free and unlimited analytics solution
  - Crash reporting

Firebase Authentication



https://www youtube com/watch?v=8sGY55yxicA

- Firebase Auth
  - Authenticate users using client-side code
  - Supports social login providers such as Facebook, GitHub, Twitter, and Google
  - Includes user management system whereby developers can enable user authentication with email and password login stored with Firebase

Pros

- Easy setup - No back-end (server) coding needed
- Supports cross-platform out of the box
- Real-time updates
- Free (Up to 100 users simultaneously)
- Support for several frameworks, including Javascript (AngularJS, Ionic, ReactJS), iOS, Android, Rest (Ruby, Python)
- Fine-tuned security rules
- Easier testing

Cons

- Files uploaded to Firebase are stored in Google Cloud Storage and **not** in the Realtime Database (RTD), as the RTD only supports Strings
- Query capabilities are somewhat limited - Must string together pre-existing queries
- Potential data processing delays when it comes to large pieces of data
- Network connection dependency - Potential issue with mobile devices with lots of in-app traffic

Pricing

- https://firebase google com/pricing

Competition

AWS Mobile Hub

What is it: *Integrated console that helps you create, build, test, and monitor your mobile apps that leverage AWS (Amazon Web Service)*

- Integrates many AWS services like authentication, data storage, backend logic, push notifications, and more in one convenient dashboard
- Very scalable
- Best in-class pricing with generous limits

- Need to use another programming language (Java, Python, or Node js)
- Somewhat steep learning curve

Json Format

- All Firebase Realtime Database data is stored as a JSON object
- You can think of the database as a cloud-hosted JSON tree
- Unlike a SQL database, there are no tables or records
- When you add data to the JSON tree, it becomes a node in the existing JSON structure with an associated key
- You can provide your own keys, such as user IDs or semantic names, or they can be provided for you using push()

```
{
 "users":    {
             "alovelace": {
                          "name": "Ada Lovelace",
                          "contacts":
                                       { "ghopper":true },
              },
             "ghopper": {    },
             "eclarke": {    }
             }
}
```

Google IO Videos

Full video

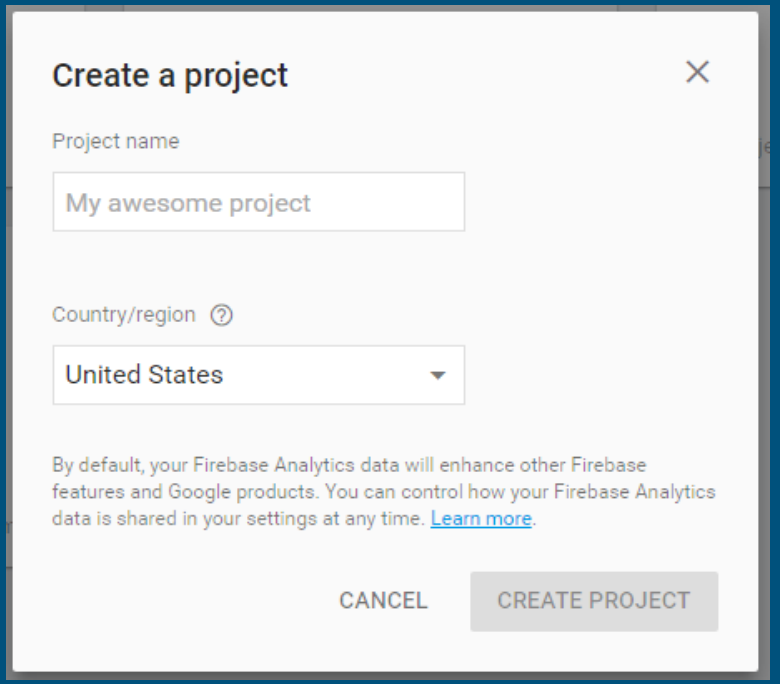
https://www youtube com/playlist?list=PLl-K7zZEsYLlAyGS6_paVoGJ9YKC7J3NN

Firebase and Realtime Database on Android

https://console firebase google com/

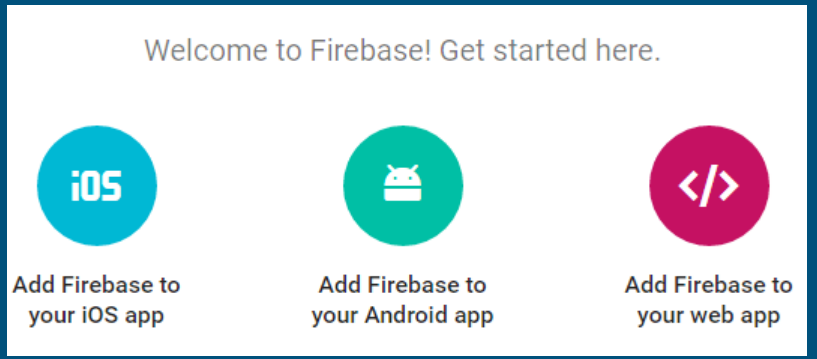
Create a project

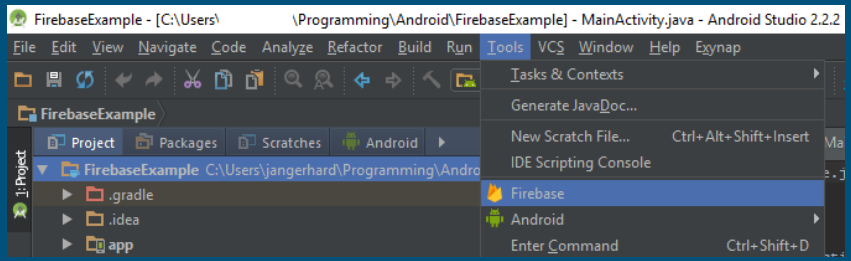
- Log in, and click “Create New Project”
- A project is a container for your apps across all platforms: Android, iOS, and the Web
- You can name this whatever you want! Enter a name and click “Create Project”



Choose a platform

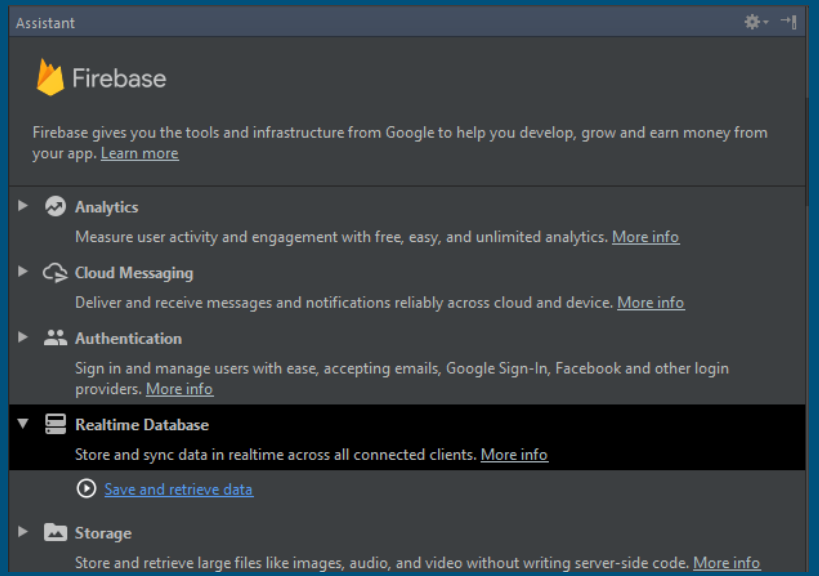
- Choose your desired platform for development For this tutorial we will focus on Android
- Open Android Studio and create a blank project called FirebaseExample
- Make sure Android Studio is version 2 2+
- Follow the instructions
- When your android project is created, you can go to the assistant Tools > Firebase in the toolbar for the next steps





Configuring Firebase

- In the assistant window, click Realtime Database, and then “Save and retrieve data”
- Follow the instructions connecting Firebase to your project by completing step 1, step 2 adding code to your project

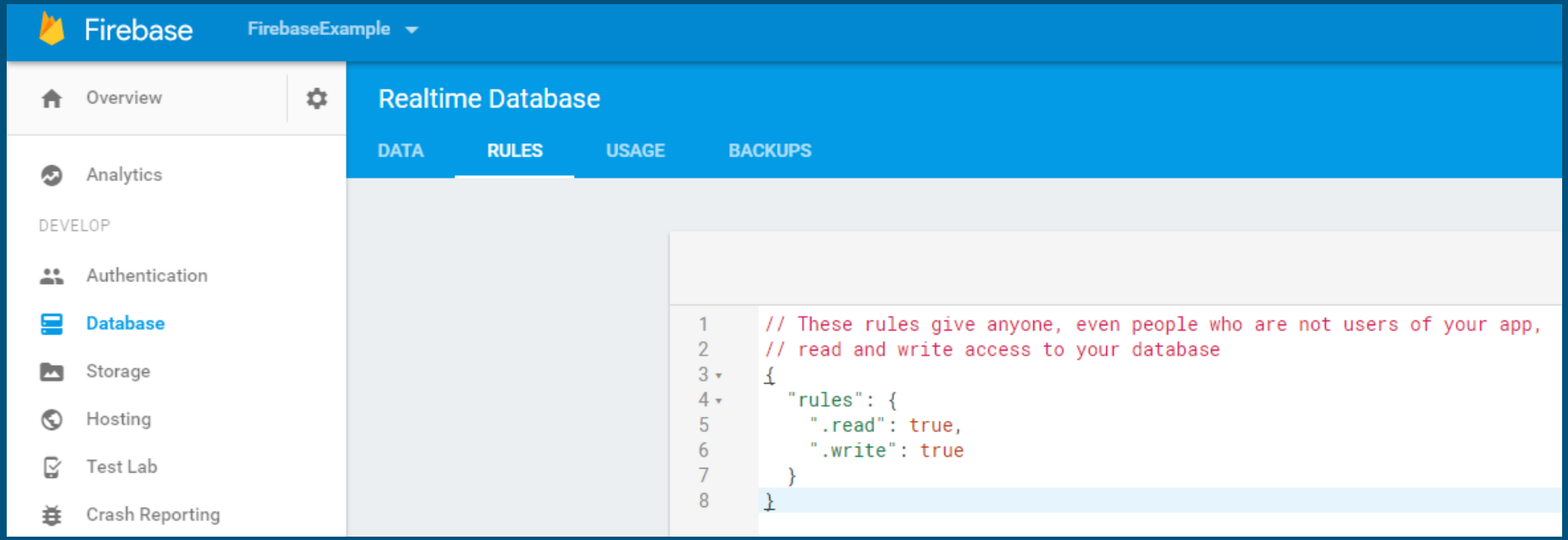


Configuring Authentication

- Let’s configure an open app for this tutorial, completing step 3 When making a real application you should set up user authentication, but we don’t need it now
- Go to your Firebase Console, and click “Database” in the left-most panel
- Paste the code, given here, under the tab “Rules”

```
// 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

 }

}
```

Authentication



Adding a textbox

- In Android Studio, add this Textview to activity_main xml
- This is the textview that will display real time information from Firebase
- Add a reference to the TextView in MainActivity java

```
<TextView
    android:id="@+id/firebaseText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!" />
```

```
public class MainActivity extends AppCompatActivity {

    TextView firebase_tv;

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

        firebase_tv = (TextView) findViewById(R.id.firebaseText);
```

Write to Firebase


```
// Write a message to the database
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("message");

myRef.setValue("Hello, World!");
```

- Follow step 4 of the Assistant in Android Studio
- Add the following code
- This will set up the necessary connection to firebase, under a reference called “message” Since this does not already exist, Firebase will create it
- We will set the value under “message” to be “Hello, World!”

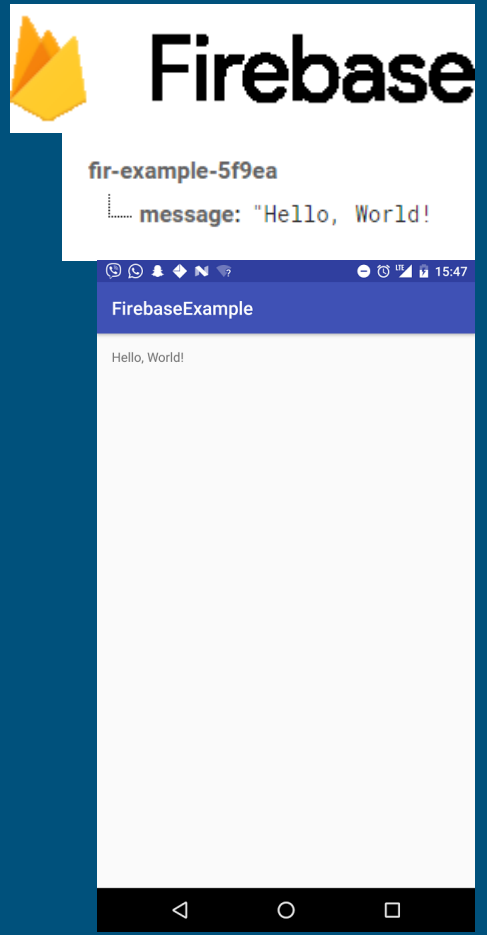
Read from Firebase

- Follow step 5 of the Assistant in Android Studio
- Add the following code
- This will update the TextView whenever there is a change in the database
- Run your app on your phone and in an emulator! It’s ready!

```
// 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.
        String value = dataSnapshot.getValue(String.class);
        firebase_tv.setText(value);
        //Log.d(TAG, "Value is: " + value);
    }
```

In the Firebase console

- As you open your console, and look at your Realtime Database, you will see a new entry “message: “Hello, World!”
- This was created as your app ran If you change this value, you can see it also updates instantly on your phone! (Given an active internet connection)



Quick recap on Read/Write

- Set up a reference point to any place in your database
- Use setValue() to set the value to any string
- Use addValueEventListener() to listen to changes made to the reference, and update your textview accordingly

```
// Write a message to the database
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("message");

myRef.setValue("Hello, World!");

// 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.
        String value = dataSnapshot.getValue(String.class);
        firebase_tv.setText(value);
        //Log.d(TAG, "Value is: " + value);
    }
```

Extra sources

- For more information on saving and writing data, visit the following link: *https://firebase google com/docs/database/android/read-and-write*

- Add Firebase features to your app:
  - Gain insights on user behavior with *Analytics*
  - Set up user authentication with *Authentication*
  - Store user info with *Realtime Database* or blob data with *Storage*
  - Send notifications to users with *Cloud Messaging*
  - Find out when and why your app is crashing with *Crash Reporting*