

FIREBASE



Ko Zaw Htet Naing

@MM-Digital-Solutions

@PADC-Myanmar

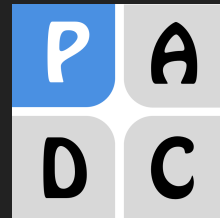


What is Firebase?



Firebase

- **Firebase is a mobile platform from Google offering a number of different features**
- **Specifically, these features revolve around cloud services, allowing users to save and retrieve data to be accessed from mobile device**



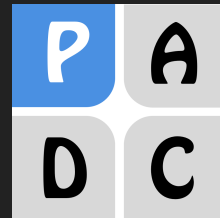
Firebase Features Overview

- **Data Storage**
- **Authentication**
- **Messaging**
- **Crash Reporting**
- **Machine Learning Features with MLKit**
- **Analytics**



Advantages of using Firebase

- **Has data storage features such as Realtime Database and Cloud Firestore eliminating the need of a backend web service with Rest API**
- **Firebase authentication handles most authentication methods and has an easy to use SDK**
- **MLKit feature has pre trained models that recognise text and faces and an easy to use sdk to utilise them.**
- **Dynamic Links are the smart URLs that dynamically change behavior to provide the best experience across various platforms. You can use it in web, email, social media, referral or any promotions to increase user acquisition, maintenance,**

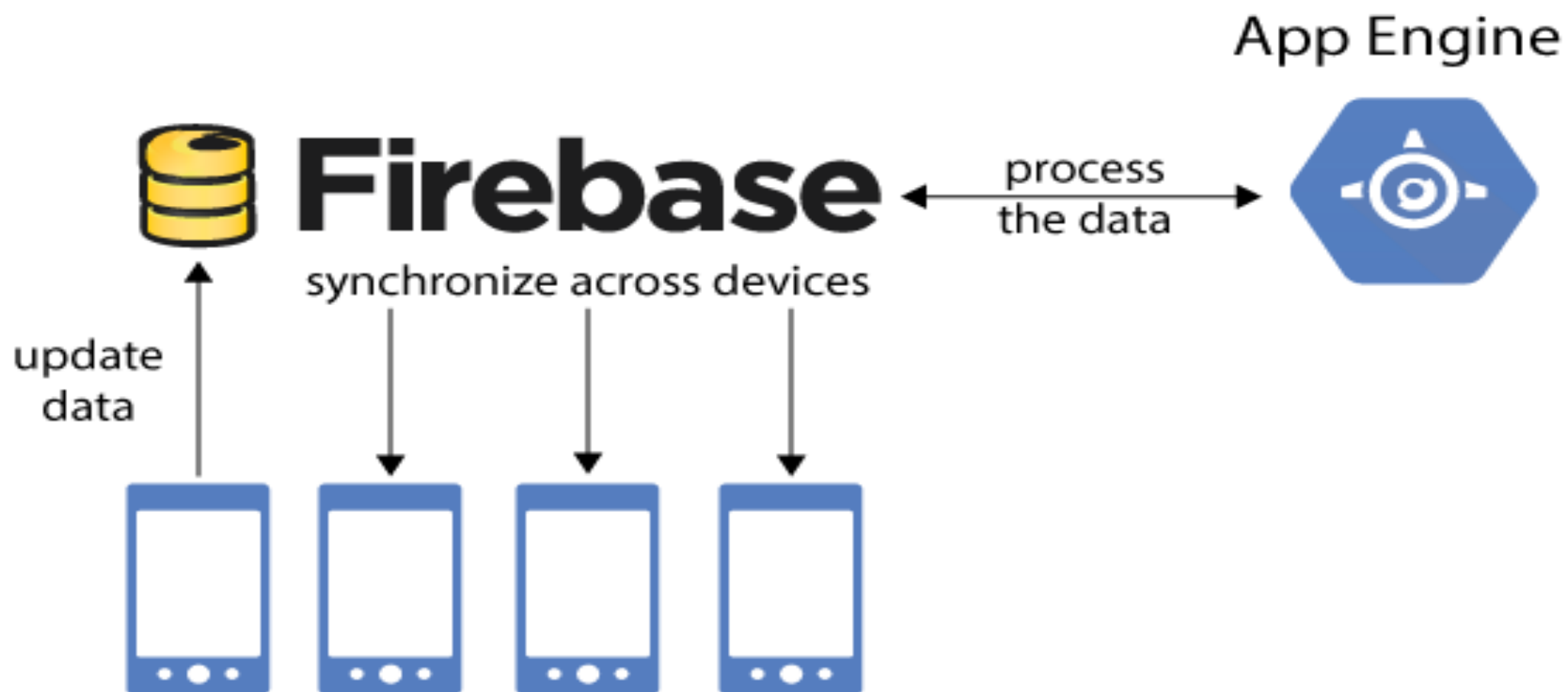


Firestore Installation and Configuration

(Will be explained in detail during screencast)

Firestore Cloud Database

- **Firestore has two cloud-based database solutions that provide real time syncing**
 - **Realtime Database**
 - **Cloud Firestore**





Firestore Realtime Database

- Lets the app store and sync data in real time
- When data in the Realtime Database is updated, it stores the data in the cloud and immediately notifies all the devices in seconds
- Stores data in JSON Format.

Storing Data in JSON Structure

- **Stores data in Key value pairs**

```
{
  "company": {
    "name": "PADC",
    "address": "Golden Gate Tower, Pazuntaung"
  }
}
```

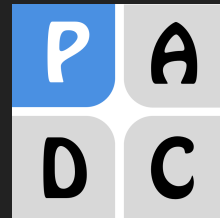
- **Realtime Database is Schemaless which means there is no restriction on data type of value being stored.**
- **To Specify the data type of values, Security Rules have to be used**

What are security Rules?

- **Firestore Realtime Database Security Rules determine who has read and write access to your database, how your data is structured. These rules are enforced automatically at all times.**

```
{
  "rules": {
    "company": {
      ".validate": "name.isString() && address.isString()"
    }
  }
}
```

- **In the above security rule, the data type of name and address is restricted to String.**



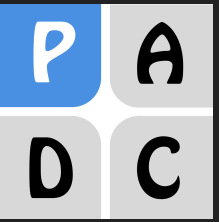
Preparing JSON data and Exporting it to Realtime Database

(Will be explained in detail during screencast)



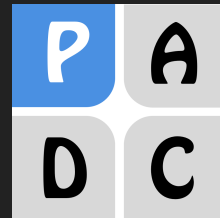
Reading Data From App

(Will be explained in detail during screencast)



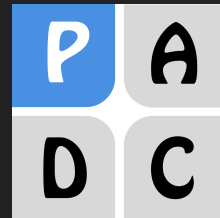
Saving Data From App

(Will be explained in detail during screencast)



Updating and Deleting Data From App

(Will be explained in detail during screencast)

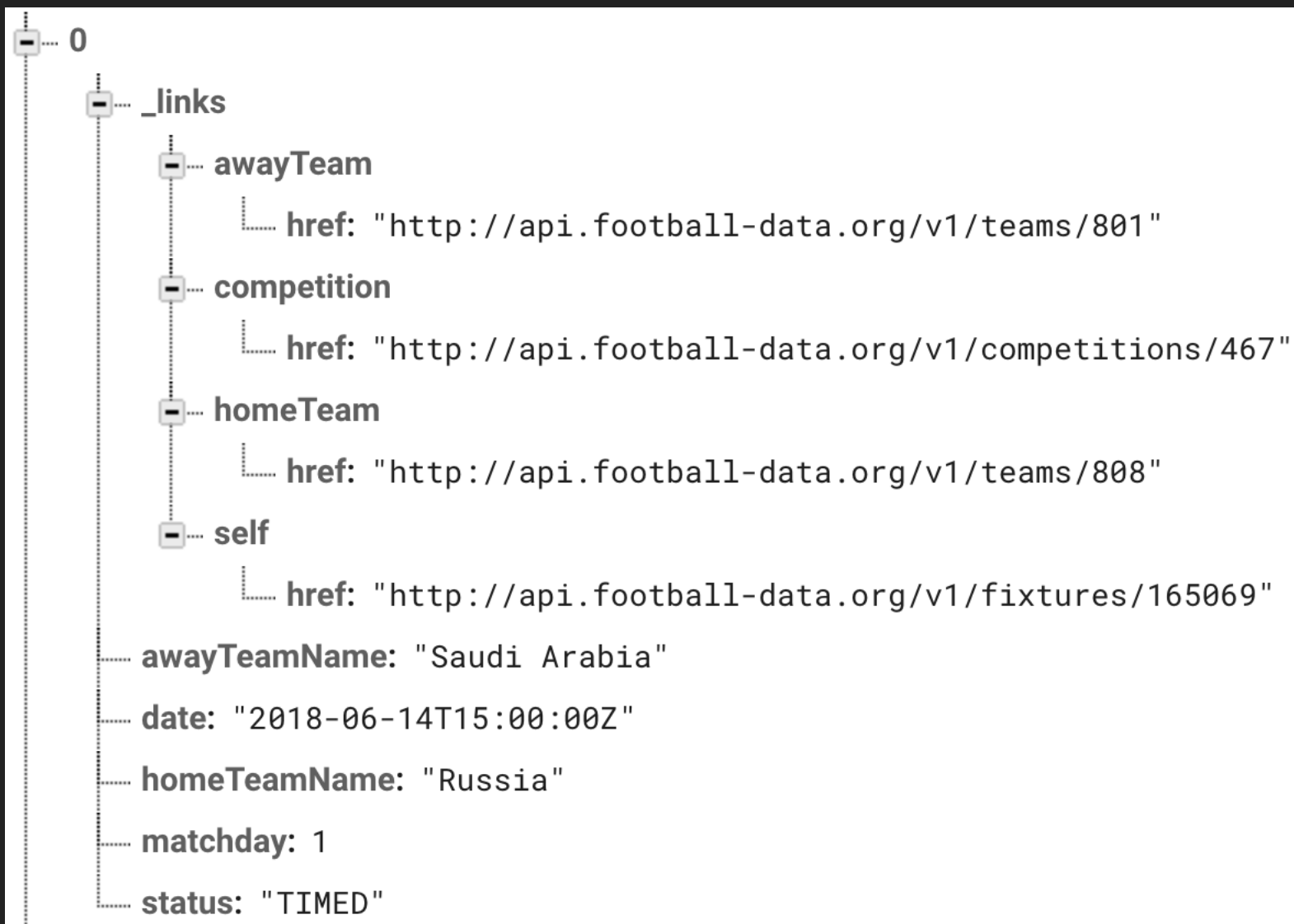


Cloud Firestore

- **Cloud based NOSQL Database**
- **Also store and sync data across devices in Real time**
- **Stores data in Documents and Collections**

Differences between Realtime Database and Cloud Firestore

- **Realtime Database stores data as one large JSON tree which makes it easier to store simple data but harder to organize complex, hierarchical data at scale.**



Differences between Realtime Database and Cloud Firestore

Firestore

- **Cloud Firestore stores data in documents arranged in collections. Simple data is stored in documents, which is easy and similar to the way data is stored in JSON. Complex, hierarchical data is conveniently organized at scale using subcollections within documents.**

Database

Cloud Firestore BETA

DATA

RULES ⓘ

INDEXES

USAGE

🏠

>

users_data

>

mKjSOK4gpDxf...

<div>🔗 user-auth-b1fa7</div> <div>+ ADD COLLECTION</div>	<div>📁 users_data</div> <div>+ ADD DOCUMENT</div>	<div>📄 mKjSOK4gpDxfqyfmoek3</div> <div>+ ADD COLLECTION</div>
<div>users_data</div> <div>></div>	<div>mKjSOK4gpDxfqyfmoek3</div> <div>></div>	<div>+ ADD FIELD</div> <div>email_id: "jinalshah999@gmail.com"</div> <div>mobile_no: "9825889888"</div> <div>name: "jinal"</div> <div>prefred_time: "morning"</div>

Documents and Collections

Documents

Document

```
bird_type: "swallow"
airspeed: 42.733
coconut_capacity: 0.62
isNative: false
icon: <binary data>
vector:
  {x: 36.4255,
   y: 25.1442,
   z: 18.8816}
distances_traveled:
  [42, 39, 12, 42]
```

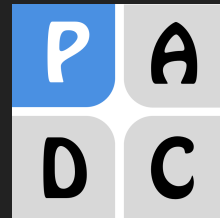
- Documents are similar to json objects, they store data in key value pairs which are referred to as “Fields”.
- The value of the Fields can be different data types consisting of mostly primitives and json objects known as “Maps”. They can also contain “Sub Collections”.
- A document should be only 1MB in size should contain less Fields as possible. Any bigger than 1MB and we need to break it up.



Documents and Collections

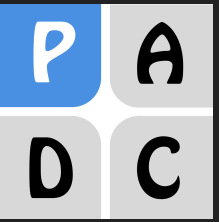
Collections

- **Collection is a group of Documents.**
- **A collection of documents inside another document is known as a “Sub Collection”.**
- **Collections and Sub Collections can contain only Documents, not primitives.**
- **The root of the Cloud Firestore structure must be a Collection. It must not be a Document.**



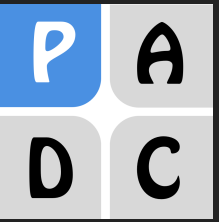
Creating Documents and Collections in Firebase Dashboard

(Will be explained in detail during screencast)



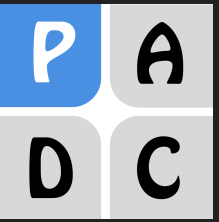
Reading Data From App

(Will be explained in detail during screencast)



Saving Data From App

(Will be explained in detail during screencast)

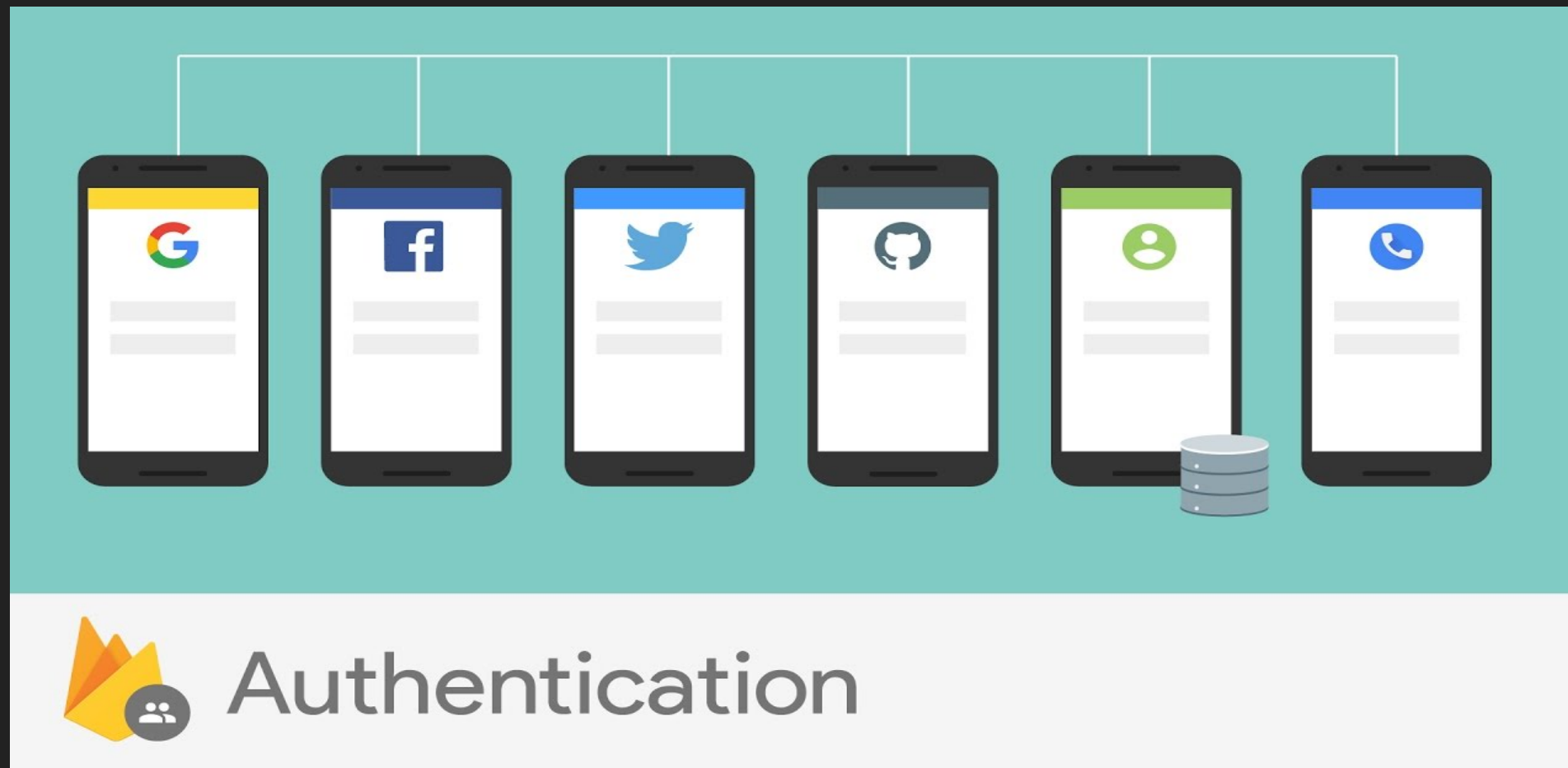


Updating and Deleting Data From App

(Will be explained in detail during screencast)

Firestore Authentication

- **Firestore provides an authentication sdk to allow only the registered users to use the app.**
- **The main feature of the sdk is allowing the users to authenticate with Email and a password.**
- **Firestore Authentication also supports third party platforms such as Facebook, Google and Twitter**





Firestore Email Authentication

- **Authentication in Firestore is done By Firestore Auth framework**
- **To create a user createUserWithEmailAndPassword function is called from the Firestore Auth Framework and email and password must be provided**
- **This process can fail for a different reasons and an a way to show the user that the authentication has failed should be implemented**
- **The email format can be customised in the Firestore Authentication dashboard in the Firestore Console.**

User account creation

- An user account with an email and password can easily be created with the function “`createUserWithEmailAndPassword`” and passing in email and password as arguments.
- And in the callback, the created user can be accessed with “`currentUser`” attribute
- If the task is not successful, the error message can be displayed. In this example, Toast message is used to display the error message.

```
auth.createUserWithEmailAndPassword(email, password)
    .addOnCompleteListener(this) { task ->
        if (task.isSuccessful) {
            // Successfully created the user
            val user = auth.currentUser
        } else {
            // Create User Proccess Failed, Error is displayed as Toast
            Toast.makeText(baseContext, "Authentication failed.",
                Toast.LENGTH_SHORT).show()
        }
    }
```

Logging In From the App

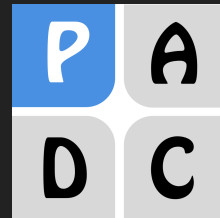
- An existing user can be easily logged in with the function “signInWithEmailAndPassword”.
- And in the callback, the created user can be accessed with “currentUser” attribute
- If the task is not successful, the error message can be displayed. In this example, Toast message is used to display the error message.

```
auth.signInWithEmailAndPassword(email, password)
    .addOnCompleteListener(this) { task ->
        if (task.isSuccessful) {
            // Successfully created the user
            val user = auth.currentUser
        } else {
            // Create User Proccess Failed, Error is displayed as Toast
            Toast.makeText(baseContext, "Authentication failed.",
                Toast.LENGTH_SHORT).show()
        }
    }
}
```



Firestore Cloud Messaging

- **Easy to use sdk from Firestore to send messages to the app Users.**
- **These data from these messages can be populated in a notification and can be shown to the user from the app.**
- **The message can be sent to a single user, a group of users or with Topics.**
- **Test messages can be sent via Firestore dashboard but in real world applications, the messages are sent via rest api since sending from Firestore Dashboard does not provide the option to send dynamic data messages (For eg. The id of a product in an E-Commerce App)**
- **Sending messages to a user or a group of users is done by unique device tokens.**
- **Sending messages with a “Topic” is different. The message is sent to the devices listening to a specific “Topic” Regardless of the device token.**



Praparation to receive Firebase Cloud Messages

(Will be explained in detail during screencast)



YOU ARE WELCOME



Ko Zaw Htet Naing

[@MM-Digital-Solutions](#)

[@PADC-Myanmar](#)

