Introduction to Firebase

Topics

- Introduction
- Firebase Realtime Database
- Setting up the development environment for the Firebase
- Creating Realtime Database , Cloud Fire store
- Reading and writing the data
- Examples

INTRODUCTION

- Back end as Service Tool developed by google Provide a number of cross plat form services
  - **Analytics** : insight on app usage and user engagement
  - **Realtime Database** lets you sync data across all clients
  - **Cloud Firestore**: uses a scalable NoSQL cloud database to store and sync data.
  - **Storage:-** lets you store data such as photos or videos.
  - **Authentication:**

REALTIME DATABASE

- Is a cloud-based database
- The data is stored a json objects and is only intended for text, to allow for fast responses.
  - Any time data changes, any connected device receives that update within milliseconds
  - Works even offline, with cached data then sync when online.

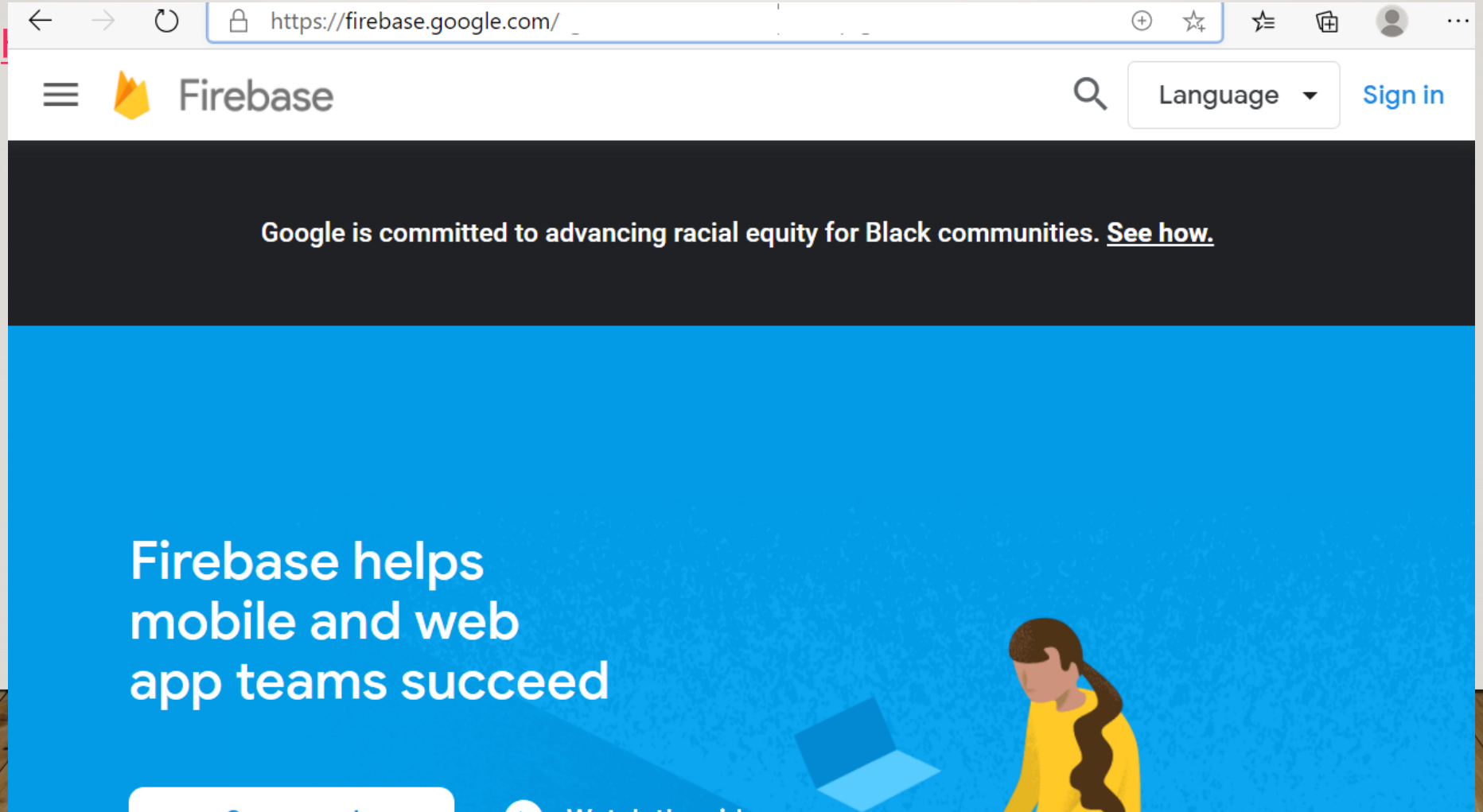
Cloud Firestore

- is a flexible, scalable database for mobile, web
- stores data in the form of documents which are organised in collections
- it keeps your data in sync across client apps through realtime listeners
- It provides offline support for mobile and web clients

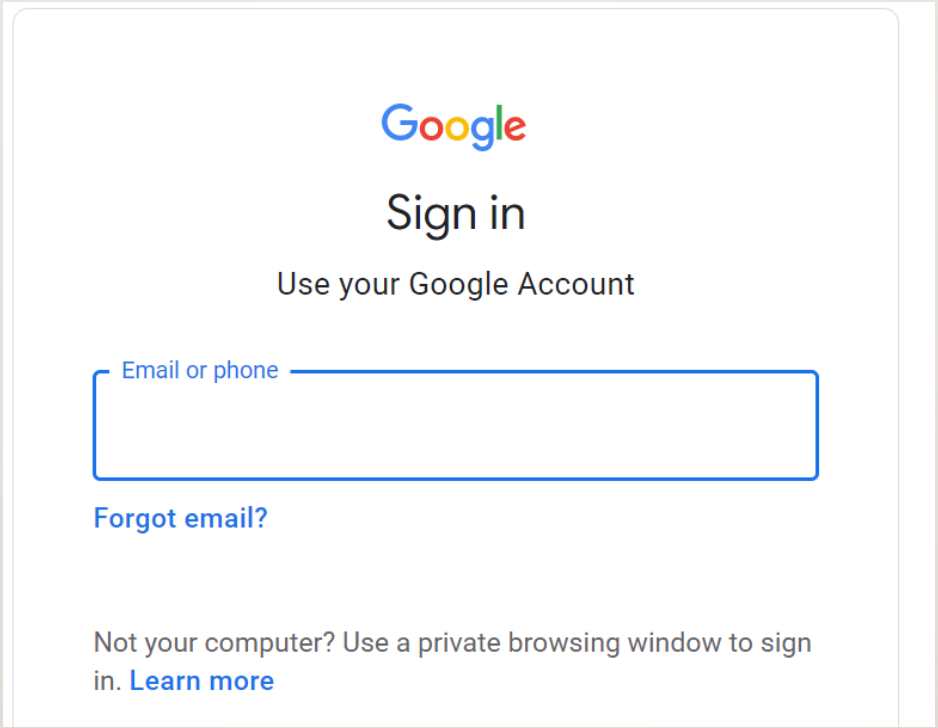
EXAMPLE

- Setting up the development environment for the Firebase
- 1- create android project
- 2- create firebase account
- 3- add firebase to android project

CREATING FIREBASE ACCOUNT

- 

Login Using Gmail



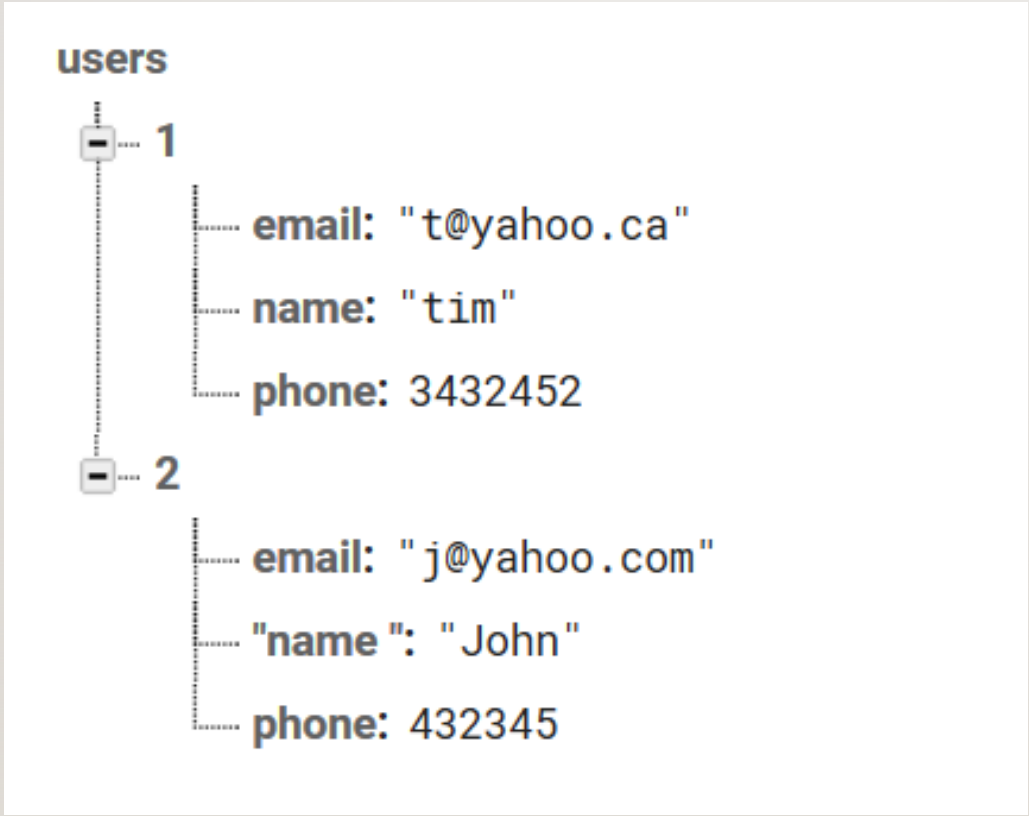
Installation & Setup On Android

- Installation & Setup on Android

USERS

id	name	email	phone
1	Tim	t@yahoo.ca	60896
2	John	j@yahoo.ca	98698
3	Jane	jane@gmail.com	98-7-989
4	Mark	m@gmail.com	32452

Realtime Database



```java
public class User {
    int id;
    String name;
    String email;
    User(){}
    User( int id , String n, String e){
        this.id = id;
        this.name = n;
        this.email = e;
    }
```

Reference To Database

```
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("Users");

  // adding new user
User user = User( “Jane",“jane@yahoo.ca",”9058986");
myRef.child(”100").setValue(user);
```

```
// 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.
         DataSnapshot users = dataSnapshot.child("users");

         for ( DataSnapshot  c : users.getChildren()){

               User u = c.getValue(User.class);
               Log.d(TAG, msg: "onDataChange: "+u);
         }
        // Log.d(TAG, "Value is: " + value);
     }
```

Cloud Fire store

```
void  addData( String name , int count){

    Map<String, User> user = new HashMap<>(); // Create a  Map u
    User u = new User(count, name,  e: name+"@yahoo.com");  // Create a new user
    user.put(name, u);

// Add a new document with a generated ID
    FirebaseFirestore db = FirebaseFirestore.getInstance();
    db.collection( collectionPath: "Customers") CollectionReference
            .add(user) Task<DocumentReference>
            .addOnSuccessListener(new OnSuccessListener<DocumentReference>() {
                @Override
                public void onSuccess(DocumentReference documentReference) {
                    Log.d(TAG,  msg: "DocumentSnapshot added with ID: " + documentReference.getId());
                }
            })
            .addOnFailureListener(new OnFailureListener() {...});
  }
```

Exercise:- Create books Firebase database

Id	Name	Author	Price
1	Java	Cay S.	40
2	C++	Paul Deitel	30
3	Android	Ian Darwin	34
4	IOS	Ahmad Sahar	35

- Get reference to database
- add new book