JÖNKÖPING UNIVERSITY

School of Engineering

ANDROID CLOUD FIRESTORE

Peter Larsson-Green
Jönköping University
Spring 2019

ANDROID APP ←→ FIREBASE PROJECT

Instructions: https://firebase.google.com/docs/android/setup

1. Add your Android package name to your Firebase project.
2. Download `google-services.json` and add it to your Android app.
3. Change your gradle build files to load Firebase dependencies.

ANDROID APP ←→ CLOUD FIRESTORE

Instructions: https://firebase.google.com/docs/firestore/quickstart

1. Change your gradle build files to load Firestore dependencies.

```
implementation 'com.google.firebase:firebase-firestore:18.0.1'
```

2.
```
FirebaseFirestore db = FirebaseFirestore.getInstance();
```

INSERT A NEW DOCUMENT

```
Map<String, Object> human = new HashMap<>();
human.put("name", "Alice");
human.put("age", 10);
db.collection("humans")
  .add(human)
  .addOnSuccessListener(new OnSuccessListener<DocumentReference>() {
    @Override
    public void onSuccess(DocumentReference documentReference) {
      String id = documentReference.getId();
    }
  })
  .addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) { }
  });
```

INSERT A NEW/REPLACE A DOCUMENT

```
Map<String, Object> human = new HashMap<>();
human.put("name", "Alice");
human.put("age", 10);
db.collection("humans").document("the-id")
  .set(human)
  .addOnSuccessListener(new OnSuccessListener<Void>() {
    @Override
    public void onSuccess(Void aVoid) {
    }
  })
  .addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) { }
  });
```

UPDATE AN EXISTING DOCUMENT

```
Map<String, Object> changes = new HashMap<>();
changes.put("name", "Bob");
db.collection("humans").document("the-id")
  .update(changes)
  .addOnSuccessListener(new OnSuccessListener<Void>() {
    @Override
    public void onSuccess(Void aVoid) {
    }
  })
  .addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) { }
  });
```

DELETE AN EXISTING DOCUMENT

```
db.collection("humans").document("the-id")
  .delete()
  .addOnSuccessListener(new OnSuccessListener<Void>() {
    @Override
    public void onSuccess(Void aVoid) {
    }
  })
  .addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) { }
  });
```

RETRIEVE A SINGLE DOCUMENT

```
db.collection("humans").document("the-id")
  .get()
  .addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
    @Override
    public void onSuccess(DocumentSnapshot documentSnapshot) {
      bool exists = documentSnapshot.exists();
      String id = documentSnapshot.getId();
      HashMap<String, Object> data = documentSnapshot.getData();
    }
  })
  .addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) { }
  });
```

RETRIEVE A SINGLE DOCUMENT

```
ListenerRegistration listener = db.collection("humans").document("the-id")
  .addSnapshotListener(new EventListener<DocumentSnapshot>() {
    @Override
    public void onEvent(@Nullable DocumentSnapshot snapshot,
                        @Nullable FirebaseFirestoreException e) {
    }
  });

listener.remove();
```

RETRIEVE MULTIPLE DOCUMENTS

```
db.collection("humans").whereXXX("age", 10)
  .get()
  .addOnSuccessListener(new OnSuccessListener<QuerySnapshot>() {
    @Override
    public void onSuccess(QuerySnapshot querySnapshot) {
      List<DocumentSnapshot> snapshots = querySnapshot.getDocuments();
    }
  })
  .addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) { }
  });
```

RETRIEVE MULTIPLE DOCUMENTS

```
ListenerRegistration listener = db.collection("humans").whereXXX("age", 10)
  .addSnapshotListener(new EventListener<DocumentSnapshot>() {
    @Override
    public void onEvent(@Nullable DocumentSnapshot snapshot,
                        @Nullable FirebaseFirestoreException e) {
    }
  });

listener.remove();
```

USING CLASSES INSTEAD

```
db.collection("humans").add(new Human("Alice", 10));
```

```
Human human = documentSnapshot.toObject(Human.class));
```

```
public class Human{
  public String name;
  public int age;
  public Human(){ }
  public Human(String name,
               int age){
    this.name = name;
    this.age = age;
  }
  public void setName(String name){
    this.name = name;
  }
  public void setAge(int age){
    this.age = age;
  }
}
```

EXAMPLE