import com.google.firebase.database.\*

// Get a reference to your Firebase database

val database = FirebaseDatabase.getInstance()

// Get a reference to the "students" collection

val studentsRef = database.getReference("students")

// Query the "students" collection and retrieve the last record

val studentsQuery = studentsRef.orderByChild("timestamp").limitToLast(1)

// Add a ValueEventListener to retrieve the data

studentsQuery.addListenerForSingleValueEvent(object : ValueEventListener {

override fun onDataChange(dataSnapshot: DataSnapshot) {

if (dataSnapshot.exists()) {

// Retrieve the last student record

val lastStudentRecord = dataSnapshot.children.last().getValue(Student::class.java)

println("Last student record: $lastStudentRecord")

} else {

println("No student data available")

}

}

override fun onCancelled(databaseError: DatabaseError) {

println("Database error: ${databaseError.message}")

}

})

// Get a reference to the "tutors" collection

val tutorsRef = database.getReference("tutors")

// Query the "tutors" collection and retrieve the last record

val tutorsQuery = tutorsRef.orderByChild("timestamp").limitToLast(1)

// Add a ValueEventListener to retrieve the data

tutorsQuery.addListenerForSingleValueEvent(object : ValueEventListener {

override fun onDataChange(dataSnapshot: DataSnapshot) {

if (dataSnapshot.exists()) {

// Retrieve the last tutor record

val lastTutorRecord = dataSnapshot.children.last().getValue(Tutor::class.java)

println("Last tutor record: $lastTutorRecord")

} else {

println("No tutor data available")

}

}

override fun onCancelled(databaseError: DatabaseError) {

println("Database error: ${databaseError.message}")

}

})

In this example:

* We have two collections: "students" and "tutors".
* We retrieve the last record from each collection separately by creating different references (**studentsRef** and **tutorsRef**) and querying each reference individually.
* Replace **Student** and **Tutor** with your actual data model classes for students and tutors, respectively.
* Ensure that the field used for sorting (**timestamp** in this case) exists in both collections if you're sorting by the same criteria. Otherwise, adjust the sorting criteria as needed for each collection.

Top of Form