**Class Activity: Firebase – Step by Step Solution**

**\*\* Due to version issue, the Java solution provided is in API 34, need to run with API 34 emulator devices as well.**

1. Create a new project with the following configuration,
   1. Empty Views Activity
   2. Language: Java
   3. Minimum API level: API 21: Android 5.0 (Lollipop)

**A screenshot of a computer

Description automatically generated**

A screenshot of a computer

Description automatically generated

1. Reuse the xml layout file and the Note class created in Activity 10\_room\_database for this activity.

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"  android:padding="16dp"  android:gravity="center"  tools:context=".MainActivity">  <EditText  android:id="@+id/editTitle"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:background="#d3d3d3"  android:padding="16dp"  android:layout\_marginBottom="8dp"  android:hint="Enter your notes title here" />  <EditText  android:id="@+id/editContent"  android:layout\_width="match\_parent"  android:layout\_height="136dp"  android:background="#d3d3d3"  android:layout\_marginBottom="16dp"  android:padding="16dp"  android:hint="Enter your notes content here"  android:lines="5"  android:inputType="textMultiLine" />  <Button  android:id="@+id/saveButton"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="32dp"  android:padding="8dp"  android:text="Save" />  <TextView  android:id="@+id/notesTitle"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="8dp"  android:paddingBottom="8dp"  android:text="Notes Title"  android:textAlignment="center"  android:textSize="24sp"  android:textStyle="bold" />  <TextView  android:id="@+id/notesContent"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="32dp"  android:paddingBottom="128dp"  android:text="Set up your notes content!"  android:textAlignment="center"  android:textSize="16sp" />  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  android:gravity="center">  <Button  android:id="@+id/deleteButton"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Delete"  android:padding="8dp" />  </LinearLayout>  </LinearLayout> |

|  |
| --- |
| package com.example.firebase;  public class Note {  private int id;  private String title;  private String content;  public Note() {  }  public Note(int id, String title, String content) {  this.id = id;  this.title = title;  this.content = content;  }  public void setId(int id) {  this.id = id;  }  public int getId() {  return this.id;  }  public void setTitle(String title) {  this.title = title;  }  public String getTitle() {  return this.title;  }  public void setContent(String content) {  this.content = content;  }  public String getContent() {  return this.content;  }  } |

1. Follow the steps taught in the slides to create a new project in Firebase console and connect your Android app to Firebase project.
2. After creating the Firebase project, create a Realtime Database and set up the database with its location and security rules.

A screenshot of a computer

Description automatically generatedIn this activity, you can set it in Singapore.

A screenshot of a computer

Description automatically generatedChoose the “Start in test mode” for security rules for this activity.

1. In the *Main Activity.java*, declare variables to link specific UI elements from the Android layout file. Retrieve an instance of your database using getInstance() and reference the location you want to write to.

Since we are using Singapore as our database location, you need to pass the database URL to getInstance(). For a us-central1 default database, you can call getInstance() without arguments.

A screenshot of a computer

Description automatically generated

Get your database URL here.

|  |
| --- |
| editTitle = findViewById(R.id.editTitle);  editContent = findViewById(R.id.editContent);  displayTitle = findViewById(R.id.notesTitle);  displayContent = findViewById(R.id.notesContent);  saveButton = findViewById(R.id.saveButton);  deleteButton = findViewById(R.id.deleteButton);  // Create reference to the database  FirebaseDatabase database = FirebaseDatabase.getInstance(//Replace with your own firebase url here);  DatabaseReference myRef = database.getReference("notes"); |

1. To make your app data update in real-time, you should add a ValueEventListener to the reference you just created.

|  |
| --- |
| // Read from the database  myRef.addValueEventListener(new ValueEventListener() {  @Override  public void onDataChange(DataSnapshot dataSnapshot) {  for (DataSnapshot snapshot : dataSnapshot.getChildren()) {  // Get the Note object from the snapshot  Note note = snapshot.getValue(Note.class);  // Display the retrieved data only for Note with id = 1  if (note != null && note.getId() == 1) {  displayTitle.setText(note.getTitle());  displayContent.setText(note.getContent());  }  }  }  @Override  public void onCancelled(DatabaseError error) {  // Failed to read value  Log.w(TAG, "Failed to read value.", error.toException());  }  });  } |

1. In the *MainActivity.java*, create setOnClickListener for the save and delete buttons. For the save button, it should store the title and content inputted by the user as a Note object into the database. For the delete button, it should delete the Note object from the database.

For the simplicity of this activity, set id = 1 when you insert or delete the Note object into Firebase. You can choose to use the id attribute as the primary key of Note object in Firebase. When a new record with the same primary key is inserted into Firebase, it will replace the old record.

|  |
| --- |
| saveButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  String title = editTitle.getText().toString();  String content = editContent.getText().toString();  Note note = new Note(1, title, content);  myRef.child(String.valueOf(note.getId())).setValue(note);  displayTitle.setText(title);  displayContent.setText(content);  }  });  deleteButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  int id = 1;  // Delete the note from the database  myRef.child(String.valueOf(id)).removeValue();  displayTitle.setText("Notes Title");  displayContent.setText("Set up your notes content!");  }  }); |

1. **Expected Output**

A screen shot of a tablet

Description automatically generated