1. Create a new Android Studio project – Empty Activity, API 16 or greater.
2. Update build.gradle (Module:app)
   1. Add packaging options inside of android definition

android {

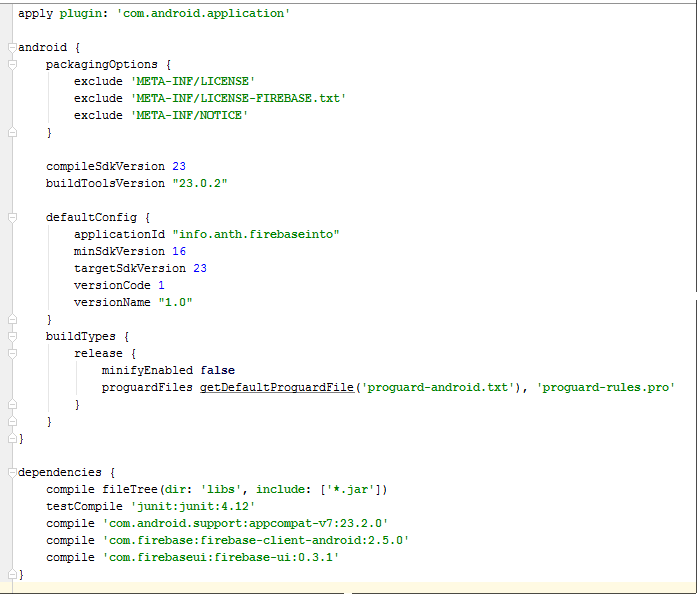
packagingOptions {  
 exclude **'META-INF/LICENSE'** exclude **'META-INF/LICENSE-FIREBASE.txt'** exclude **'META-INF/NOTICE'**}

. . .  
}

* 1. Add new firebase dependencies to dependency definition   
     dependencies {

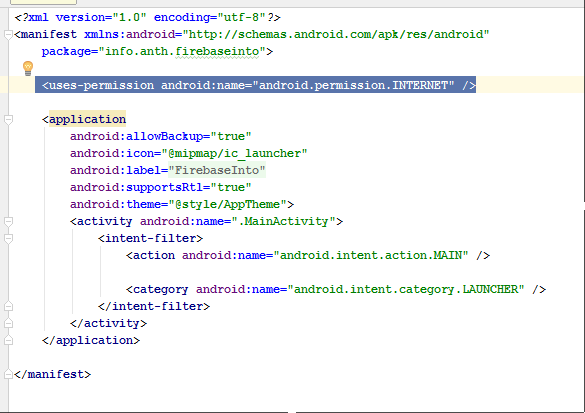
. . .  
compile **'com.firebase:firebase-client-android:2.5.0'** compile **'com.firebaseui:firebase-ui:0.3.1'**}

1. Press Sync now – verify no errors



1. Update AndroidManifest.xml file – add user permissions to the internet

**<uses-permission android:name="android.permission.INTERNET"** />

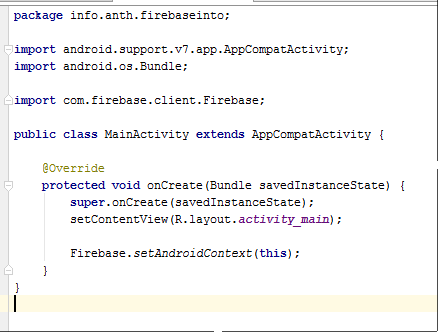


1. Update MainActivity.java – Firebase Android Context code, if it doesn’t include the import firebase code add that as well

Firebase.*setAndroidContext*(**this**);

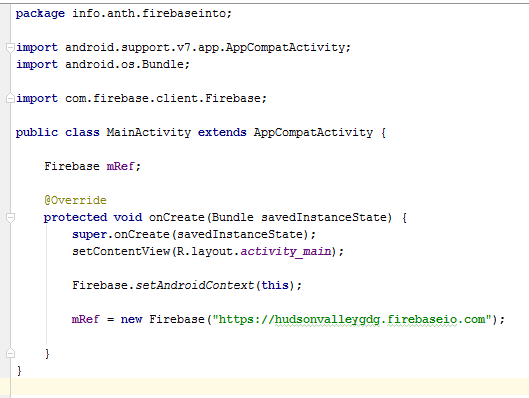
(maybe)

**import** com.firebase.client.Firebase;



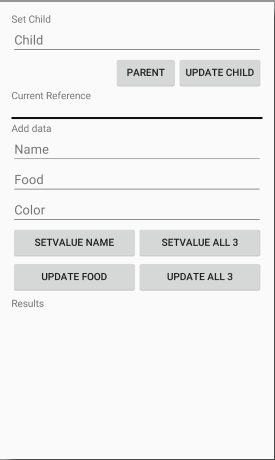
1. Run the app to verify no errors
2. Create Firebase Account at firebase.com
3. Create global reference (mRef) to your Firebase database
   1. Add variable: Firebase mRef;
   2. Add assignment to your database inside onCreate method: REPLACE hudsonvalleygdg with the name of your database.

mRef = **new** Firebase(**"https://hudsonvalleygdg.firebaseio.com"**);



1. Replace activity\_main.xml code:
2. *<?***xml version="1.0" encoding="utf-8"***?>*<**ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout\_width="match\_parent"  
    android:layout\_height="match\_parent"**>  
     
    <**RelativeLayout  
    android:id="@+id/main\_relative\_layout"  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:paddingBottom="@dimen/activity\_vertical\_margin"  
    android:paddingLeft="@dimen/activity\_horizontal\_margin"  
    android:paddingRight="@dimen/activity\_horizontal\_margin"  
    android:paddingTop="@dimen/activity\_vertical\_margin"**>  
     
    <**TextView  
    android:id="@+id/heading\_text"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="Set Child"** />  
     
    <**EditText  
    android:id="@+id/edit\_child"  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:layout\_below="@id/heading\_text"  
    android:hint="Child"** />  
     
    <**Button  
    android:id="@+id/button\_child"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:layout\_alignEnd="@+id/edit\_child"  
    android:layout\_alignRight="@+id/edit\_child"  
    android:layout\_below="@+id/edit\_child"  
    android:text="Update Child"  
    android:onClick="clickChild"**/>  
     
    <**Button  
    android:id="@+id/button\_parent"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:layout\_below="@+id/edit\_child"  
    android:layout\_toLeftOf="@+id/button\_child"  
    android:layout\_toStartOf="@+id/button\_child"  
    android:text="Parent"  
    android:onClick="clickParent"**/>  
     
    <**TextView  
    android:id="@+id/heading\_ref"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:layout\_below="@+id/button\_child"  
    android:text="Current Reference"** />  
     
    <**TextView  
    android:id="@+id/text\_reference"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:layout\_below="@+id/heading\_ref"** />  
     
    <**TextView  
    android:id="@+id/line1"  
    android:layout\_width="match\_parent"  
    android:layout\_height="3dp"  
    android:layout\_below="@+id/text\_reference"  
    android:layout\_marginBottom="5dp"  
    android:layout\_marginTop="5dp"  
    android:background="#111111"** />  
     
    <**TextView  
    android:id="@+id/heading\_add"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:layout\_below="@+id/line1"  
    android:text="Add data"** />  
     
    <**EditText  
    android:id="@+id/edit\_name"  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:layout\_below="@id/heading\_add"  
    android:hint="Name"** />  
     
    <**EditText  
    android:id="@+id/edit\_food"  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:layout\_below="@id/edit\_name"  
    android:hint="Food"** />  
     
    <**EditText  
    android:id="@+id/edit\_color"  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:layout\_below="@id/edit\_food"  
    android:hint="Color"** />  
     
    <**LinearLayout  
    android:id="@+id/layout\_set"  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:layout\_below="@+id/edit\_color"  
    android:orientation="horizontal"**>  
     
    <**Button  
    android:id="@+id/button\_set"  
    android:layout\_width="0dp"  
    android:layout\_height="wrap\_content"  
    android:layout\_weight="1"  
    android:text="setValue name"  
    android:onClick="clickSetName"**/>  
     
    <**Button  
    android:id="@+id/button\_set3"  
    android:layout\_width="0dp"  
    android:layout\_height="wrap\_content"  
    android:layout\_weight="1"  
    android:text="setvalue all 3"  
    android:onClick="clickSet3"**/>  
    </**LinearLayout**>  
     
    <**LinearLayout  
    android:id="@+id/layout\_update"  
    android:layout\_width="match\_parent"  
    android:layout\_height="wrap\_content"  
    android:layout\_below="@+id/layout\_set"  
    android:orientation="horizontal"**>  
     
    <**Button  
    android:id="@+id/button\_update"  
    android:layout\_width="0dp"  
    android:layout\_height="wrap\_content"  
    android:layout\_weight="1"  
    android:text="update food"  
    android:onClick="clickUpdateFood"**/>  
     
    <**Button  
    android:id="@+id/button\_update3"  
    android:layout\_width="0dp"  
    android:layout\_height="wrap\_content"  
    android:layout\_weight="1"  
    android:text="update all 3"  
    android:onClick="clickUpdate3"**/>  
    </**LinearLayout**>  
     
    <**TextView  
    android:id="@+id/heading\_results"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:layout\_below="@+id/layout\_update"  
    android:layout\_marginBottom="5dp"  
    android:layout\_marginTop="5dp"  
    android:text="Results"** />  
    <**TextView  
    android:id="@+id/text\_results"  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:layout\_below="@+id/heading\_results"** />  
    </**RelativeLayout**>  
   </**ScrollView**>

Should look like:



1. Add empty button methods to the MainActivity class.

**public class** MainActivity **extends** AppCompatActivity {

**. . .**  
  
 **public void** clickChild(View view){  
 }  
  
 **public void** clickParent(View view){  
 }  
  
 **public void** clickSetName(View view){  
 }  
  
 **public void** clickSet3(View view){  
 }  
  
 **public void** clickUpdateFood(View view){  
 }  
  
 **public void** clickUpdate3(View view){  
 }  
}

1. Update the click child method. This method allows you to subsets of data, children, in your database. It uses the Firebase function reference.child(“<child-name>”) to update the reference. Until a value is added to the path, you will not see anything change in the database. It also uses the Firebase function reference.getRef() to show the current reference.

**public void** clickChild(View view){  
 *// This method moves along the children of the database  
 // Each call will add a new child to the path  
 //  
 // In file systems terms: creating a sub-directory  
  
 // Get the Edit Text* EditText childEntry = (EditText) findViewById(R.id.***edit\_child***);  
 *// Get reference display text field* TextView displayRef = (TextView) findViewById(R.id.***text\_reference***);  
  
 *// update the Firebase reference (if child has a value)* **mRef** = **mRef**.child(childEntry.getText().toString());  
 *// set reference display text* displayRef.setText(**mRef**.getRef().toString());  
  
 *// clear Child edit box* childEntry.setText(**""**);  
}

1. Update the click parent method. This method moves your reference to the parent of your current reference. Stopping at the root. It uses the reference.getParent() Firebase method, and the reference.getRef() method for display. It checks to see if it is already at the root using reference.getRoot() and reference.getRef(). Nothing will be displayed in the Firebase database.

**public void** clickParent(View view){  
 *// This method moves to the parent of the current reference  
  
 // Get reference display text field* TextView displayRef = (TextView) findViewById(R.id.***text\_reference***);  
  
 *// Need to check if already at root.* **if** (!**mRef**.getRef().toString().equals(**mRef**.getRoot().toString())) {  
 *// NOT at root  
 // update the Firebase reference (if child has a value)* **mRef** = **mRef**.getParent();  
 }  
 *// set reference display text* displayRef.setText(**mRef**.getRef().toString());  
}

1. Run Program
2. Update the clickSetName and clickSet3 methods with the code below. This code navigates to the child and sets its value. Beware: the setValue method deletes all child nodes at its calling reference and inserts the new values. Navigating to the direct child avoids this issue.

**public void** clickSetName(View view){  
 *// This method saves the data for the name using the set method  
  
 // Get the Name entry* EditText nameEntry = (EditText) findViewById(R.id.***edit\_name***);  
  
 *// Navigate to the name child and set the value* **mRef**.child(**"name"**).setValue(nameEntry.getText().toString());  
}  
  
**public void** clickSet3(View view){  
 *// This method saves the data for name, food, and color using the set method  
  
 // Get the Name entry* EditText nameEntry = (EditText) findViewById(R.id.***edit\_name***);  
 *// Get the Food entry* EditText foodEntry = (EditText) findViewById(R.id.***edit\_food***);  
 *// Get the Name entry* EditText colorEntry = (EditText) findViewById(R.id.***edit\_color***);  
  
 *// Navigate to the name child and set the value* **mRef**.child(**"name"**).setValue(nameEntry.getText().toString());  
 *// Navigate to the food child and set the value* **mRef**.child(**"food"**).setValue(foodEntry.getText().toString());  
 *// Navigate to the color child and set the value* **mRef**.child(**"color"**).setValue(colorEntry.getText().toString());  
}

1. Run the program. Changes will save to the database. Use the Firebase dashboard to see the changes. <https://YOUR-DATABASE-NAME.firebaseio.com/>
2. Replace the clickSetName and clickSet3 methods with the code below. This code uses Hash Maps to set key, value pairs. Beware: the setValue method deletes all child nodes at its calling reference and inserts the new values. Navigating to the direct child avoided this issue, using hash maps does not.

**public void** clickSetName(View view){  
 *// This method saves the data for the name using the set method  
  
 // Get the Name entry* EditText nameEntry = (EditText) findViewById(R.id.***edit\_name***);  
  
 *// Define hash map and set the values* Map<String, Object> updateMap = **new** HashMap<>();  
 updateMap.put(**"name"**, nameEntry.getText().toString());  
 **mRef**.setValue(updateMap);  
}  
  
**public void** clickSet3(View view){  
 *// This method saves the data for name, food, and color using the set method  
  
 // Get the Name entry* EditText nameEntry = (EditText) findViewById(R.id.***edit\_name***);  
 *// Get the Food entry* EditText foodEntry = (EditText) findViewById(R.id.***edit\_food***);  
 *// Get the Name entry* EditText colorEntry = (EditText) findViewById(R.id.***edit\_color***);  
  
 *// Define hash map and set the values* Map<String, Object> updateMap = **new** HashMap<>();  
 updateMap.put(**"name"**, nameEntry.getText().toString());  
 updateMap.put(**"color"**, colorEntry.getText().toString());  
 updateMap.put(**"food"**, foodEntry.getText().toString());  
 **mRef**.setValue(updateMap);  
}

1. Run the program. Changes will save to the database. Use the Firebase dashboard to see the changes. <https://YOUR-DATABASE-NAME.firebaseio.com/> Notice that clicking the SETVALUE NAME button after the SETVALUE ALL 3 will delete the color and food children. This is because the setValue() Firebase function deletes all children at the reference it is executed and then inserts the new values.
2. Update the clickUpdateFood and clickUpdate3 methods with the code below. This code uses hash Maps to update the child key with its value.

**public void** clickUpdateFood(View view){  
 *// This method saves the data for food using the updateChildren method  
  
 // Get the Food entry* EditText foodEntry = (EditText) findViewById(R.id.***edit\_food***);  
  
 *// Define hash map and set the values* Map<String, Object> updateMap = **new** HashMap<>();  
 updateMap.put(**"food"**, foodEntry.getText().toString());  
 **mRef**.updateChildren(updateMap);  
}  
  
**public void** clickUpdate3(View view){  
 *// This method saves the data for name, food, and color   
 // using the updateChildren method  
  
 // Get the Name entry* EditText nameEntry = (EditText) findViewById(R.id.***edit\_name***);  
 *// Get the Food entry* EditText foodEntry = (EditText) findViewById(R.id.***edit\_food***);  
 *// Get the Name entry* EditText colorEntry = (EditText) findViewById(R.id.***edit\_color***);  
  
 *// Define hash map and set the values* Map<String, Object> updateMap = **new** HashMap<>();  
 updateMap.put(**"name"**, nameEntry.getText().toString());  
 updateMap.put(**"color"**, colorEntry.getText().toString());  
 updateMap.put(**"food"**, foodEntry.getText().toString());  
 **mRef**.updateChildren(updateMap);  
}

1. Run the program. Changes will save to the database. Use the Firebase dashboard to see the changes. <https://YOUR-DATABASE-NAME.firebaseio.com/>
2. Setup a listener at the base reference level of the database. Show in the results section of screen. (Add highlighted lines)
   1. Add variables, store results TextView

Firebase **mRefRoot**;  
ValueEventListener **eventListener**;  
TextView **resultsText**;  
  
@Override  
**protected void** onCreate(Bundle savedInstanceState) {

. . .

**mRefRoot** = **mRef**;  
 **resultsText** = (TextView) findViewById(R.id.***text\_results***);  
}

* 1. Override onResume and onPause methods

@Override  
**public void** onResume() {  
 **super**.onResume();  
 **eventListener** = **new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(DataSnapshot dataSnapshot) {  
 Log.*i*(**"FirebaseInto"**, String.*valueOf*(dataSnapshot.getValue()));  
 **resultsText**.setText(String.*valueOf*(dataSnapshot.getValue()));  
 }  
 @Override  
 **public void** onCancelled(FirebaseError firebaseError) {  
 Log.*e*(**"FirebaseInto"**, firebaseError.toString());  
 }  
 };  
 **mRefRoot**.addValueEventListener(**eventListener**);  
}  
  
@Override  
**public void** onPause() {  
 **super**.onPause();  
 **mRefRoot**.removeEventListener(**eventListener**);  
}

1. Run the program. All database values should show in the results area. Changes directly made using the Firebase dashboard will also show: <https://YOUR-DATABASE-NAME.firebaseio.com/>