```
001. package lightbox.sep.fire3;
002.
003. import android.os.Bundle;
004. import android.support.v7.app.AppCompatActivity;
005. import android.util.Log;
006. import android.view.View;
007. import android.widget.Button;
008. import android.widget.TextView;
009.
010. import com.google.firebase.database.DataSnapshot;
011. import com.google.firebase.database.DatabaseError;
012. import com.google.firebase.database.DatabaseReference;
013. import com.google.firebase.database.FirebaseDatabase;
014. import com.google.firebase.database.GenericTypeIndicator;
015. import com.google.firebase.database.ValueEventListener;
016.
017. import java.io.PrintWriter;
018. import java.io.StringWriter;
019. import java.util.ArrayList;
020. import java.util.HashMap;
021. import java.util.Iterator;
022.
023. public class MainActivity extends AppCompatActivity {
024.
025.   private FirebaseDatabase database;
026.   private DatabaseReference mDatabase;
027.
028.   @Override
029.   protected void onCreate(Bundle savedInstanceState) {
030.     super.onCreate(savedInstanceState);
031.     setContentView(R.layout.activity_main);
032.
033.     database = FirebaseDatabase.getInstance();
034.     mDatabase = database.getReference();
035.
036.     Button readButton = (Button) MainActivity.this.findViewById(R.id.readButton);
037.     readButton.setOnClickListener(new View.OnClickListener() {
038.       @Override
039.       public void onClick(View v) {
040.
041.         mDatabase.addListenerForSingleValueEvent(new ValueEventListener() {
```

```
042.         @Override
043.         public void onDataChange(DataSnapshot dataSnapshot) {
044.          if ( dataSnapshot.exists() ) {
045.           Log.i("lightbox", dataSnapshot.getValue().toString());
046.          }
047.         }
048.
049.         @Override
050.         public void onCancelled(DatabaseError databaseError) {
051.
052.         }
053.        });
054.
055.
056.        mDatabase.child("users").addListenerForSingleValueEvent(new ValueEventListener() {
057.         @Override
058.         public void onDataChange(DataSnapshot dataSnapshot) {
059.          if ( dataSnapshot.exists() ) {
060.           Log.i("lightbox", "-- users の状態");
061.           Log.i("lightbox", dataSnapshot.getValue().toString());
062.           Log.i("lightbox", "-- ArrayList<User>");
063.
064.           GenericTypeIndicator<ArrayList<User>> t = new GenericTypeIndicator<ArrayList<User>>() {};
065.           ArrayList<User> user = dataSnapshot.getValue(t);
066.           Iterator<User> it = user.iterator();
067.           while( it.hasNext() ) {
068.            User work = it.next();
069.            Log.i("lightbox", work.getCode() );
070.            Log.i("lightbox", work.getName() );
071.           }
072.
073.           Log.i("lightbox", "-- 配列");
074.
075.           User[] users = user.toArray(new User[0]);
076.           for( int i = 0; i < users.length; i++){
077.            Log.i("lightbox", users[i].getCode() );
078.            Log.i("lightbox", users[i].getName() );
079.           }
080.
081.           Log.i("lightbox", "-- ArrayList<HashMap<String,Object>>");
082.
083.           GenericTypeIndicator<ArrayList<HashMap<String,Object>>> t2
```

```
            = new GenericTypeIndicator<ArrayList<HashMap<String,Object>>>() {};
          ArrayList<HashMap<String,Object>> useList = dataSnapshot.getValue(t2);
          Iterator<HashMap<String,Object>> it2 = useList.iterator();
          while( it2.hasNext()) {
           HashMap<String,Object> userMap = it2.next();
           Log.i( "lightbox", userMap.get("code").toString() );
           Log.i( "lightbox", userMap.get("name").toString() );
          }
         }
         else {
          Log.i("lightbox","データを読み込めませんでした");
         }
        }

        @Override
        public void onCancelled(DatabaseError databaseError) {
         Log.i("lightbox","onCancelled");
         StringWriter sw = new StringWriter();
         PrintWriter pw = new PrintWriter(sw);
         databaseError.toException().printStackTrace(pw);
         pw.flush();
         String stackTrace = sw.toString();
         Log.i("lightbox",stackTrace);

        }
       });

  }
 });

 Button readUserButton = (Button) MainActivity.this.findViewById(R.id.readUserButton);
 readUserButton.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {

   mDatabase.child(String.format("users/%d",0)).addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
     if ( dataSnapshot.exists() ) {
      Log.i("lightbox", dataSnapshot.getValue().toString());

      User user = dataSnapshot.getValue(User.class);
```

```
126.
127.           TextView tv1 = (TextView) MainActivity.this.findViewById(R.id.textCode);
128.           tv1.setText(user.getCode());
129.           TextView tv2 = (TextView) MainActivity.this.findViewById(R.id.textName);
130.           tv2.setText(user.getName());
131.         }
132.       }
133.
134.       @Override
135.       public void onCancelled(DatabaseError databaseError) {
136.
137.       }
138.      });
139.
140.    }
141.   });
142.
143.
144.  }
145.
146.  public static class User {
147.
148.   private String code;
149.   private String name;
150.
151.   public User() {}
152.
153.   public User(String code, String name) {
154.    this.code = code;
155.    this.name = name;
156.   }
157.
158.   public String getCode() {
159.    return code;
160.   }
161.   public String getName() {
162.    return name;
163.   }
164.   public void setCode(String code) {
165.    this.code = code;
166.   }
167.   public void setName(String name) {
```

```
168.       this.name = name;
169.     }
170.
171.   }
172.
173. }
```