
```

1 package com.example.dipendra.paygetping;
2
3 import android.app.ProgressDialog;
4 import android.content.Intent;
5 import android.content.SharedPreferences;
6 import android.os.Bundle;
7 import android.preference.PreferenceManager;
8 import android.support.annotation.NonNull;
9 import android.support.v7.app.AppCompatActivity;
10 import android.util.Log;
11 import android.view.Menu;
12 import android.view.MenuItem;
13 import android.widget.Toast;
14
15 import com.example.dipendra.paygetping.login.LoginActivity;
16 import com.example.dipendra.paygetping.login.RegisterActivity;
17 import com.example.dipendra.paygetping.models.User;
18 import com.example.dipendra.paygetping.models.WalletListWrapper;
19 import com.example.dipendra.paygetping.utils.Constants;
20 import com.example.dipendra.paygetping.wallet.AddUserToList;
21 import com.example.dipendra.paygetping.wallet.WalletsActivity;
22 import com.google.android.gms.tasks.OnCompleteListener;
23 import com.google.android.gms.tasks.Task;
24 import com.google.firebase.auth.AuthResult;
25 import com.google.firebase.auth.FirebaseAuth;
26 import com.google.firebase.auth.FirebaseUser;
27 import com.google.firebase.database.DataSnapshot;
28 import com.google.firebase.database.DatabaseError;
29 import com.google.firebase.database.DatabaseReference;
30 import com.google.firebase.database.ValueEventListener;
31
32 /**
33  * Created by dipendra on 06/04/17.
34  */
35
36 public class BaseActivity extends AppCompatActivity {
37     protected FirebaseAuth auth;
38     protected ProgressDialog progressDialog;
39     protected SharedPreferences sp;
40     protected FirebaseAuth.AuthStateListener authStateListener;
41     protected User user;
42
43     public User getUser() {
44         return user;
45     }
46
47     public WalletListWrapper getCurrentList() {
48         return currentList;
49     }
50
51     protected WalletListWrapper currentList;
52     protected DatabaseReference reference;
53     @Override

```

```

54     protected void onCreate(Bundle savedInstanceState) {
55         super.onCreate(savedInstanceState);
56         auth = FirebaseAuth.getInstance();
57         authStateListener = new FirebaseAuth.AuthStateListener() {
58             @Override
59             public void onAuthStateChanged(@NonNull FirebaseAuth
60                 firebaseAuth) {
61                 FirebaseUser firebaseUser = auth.getCurrentUser();
62                 if(firebaseUser == null){
63                     getToLoginScreen();
64                 }
65                 else{
66                     readSharedPreferences();
67                     try {
68                         if (currentList.getOwnerID() != null &&
69                             currentList.getOwnerID().equals(user.
70                                 getEncodedEmail()) && BaseActivity.this
71                                 instanceof MainActivity) {
72                                     Constants.menu = R.menu.menu_main;
73                                 }
74                                 else {
75                                     Constants.menu = R.menu.menu_main_1;
76                                 }
77                             }catch (Exception e){
78                                 Log.e("TAG:: MAIN ACTIVITY", "Menu not yet created
79                                     ");
80                             }
81                             invalidateOptionsMenu();
82                         }
83                     };
84                     initialize();
85                 }
86             }
87         private void initialize(){
88             sp = PreferenceManager.getDefaultSharedPreferences(this);
89             user = new User();
90             currentList = new WalletListWrapper();
91             progressDialog = new ProgressDialog(this);
92             progressDialog.setIndeterminate(true);
93             progressDialog.setCancelable(false);
94             if(Constants.isOnline(this) == false){
95                 Toast.makeText(this, "Working Off-the line is not good! Please
96                     be online!", Toast.LENGTH_SHORT).show();
97             }
98         }
99         private void getToLoginScreen(){
100             Intent i = new Intent(BaseActivity.this, LoginActivity.class);
101             i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.
102                 FLAG_ACTIVITY_CLEAR_TASK);
103             startActivity(i);
104         }
105         protected void readSharedPreferences(){
106             sp = PreferenceManager.getDefaultSharedPreferences(this);
107             user.setEncodedEmail(sp.getString("encodedEmail", null));

```

```

101         user.setName(sp.getString("username", null));
102         String nameOfTheCurrentList = sp.getString("currentWalletName",
            null);
103         if(nameOfTheCurrentList == null && !(this instanceof
            WalletsActivity)){
104             Intent i = new Intent(this, WalletsActivity.class);
105             i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.
                FLAG_ACTIVITY_CLEAR_TASK);
106             startActivity(i);
107         }
108         else{
109             currentList.setName(nameOfTheCurrentList);
110         }
111         currentList.setOwner(sp.getString("currentWalletOwner", null));
112         currentList.setPushId(sp.getString("currentWallet", null));
113         currentList.setOwnerID(sp.getString("currentWalletOwnerID", null))
            ;
114         try {
115             if (this instanceof MainActivity) {
116                 getSupportActionBar().setTitle("Wallet : "+ currentList.
                    getName());
117             }
118         }
119         catch (Exception e){
120             e.printStackTrace();
121         }
122         // Toast.makeText(this, "You are:"+ user.getName()+user.
            getEncodedEmail(), Toast.LENGTH_LONG).show();
123     }
124     @Override
125     protected void onStart() {
126         super.onStart();
127         if(!(this instanceof LoginActivity || this instanceof
            RegisterActivity)){
128             auth.addAuthStateListener(authStateListener);
129         }
130     }
131
132     @Override
133     protected void onResume() {
134         super.onResume();
135     }
136
137
138
139     @Override
140     protected void onStop() {
141         super.onStop();
142         if(authStateListener != null){
143             if(!(this instanceof LoginActivity || this instanceof
                RegisterActivity)) {
144                 auth.removeAuthStateListener(authStateListener);
145             }
146         }

```

```

147     }
148     protected void signin(String usermail, String passwordString){
149         progressDialog = new ProgressDialog(this);
150         progressDialog.setMessage("Signing in.. Please Wait..");
151         progressDialog.setCancelable(false);
152         progressDialog.setIndeterminate(true);
153         progressDialog.show();
154         final String mail = usermail.replace(".",",");
155         auth.signInWithEmailAndPassword(usermail, passwordString).
            addOnCompleteListener(this, new OnCompleteListener<AuthResult>
            >() {
156             @Override
157             public void onComplete(@NonNull Task<AuthResult> task) {
158                 if(!task.isSuccessful()){
159                     progressDialog.hide();
160                     try{
161                         throw task.getException();
162                     } catch (Exception e) {
163                         Toast.makeText(BaseActivity.this, "Either Username
                            or password is wrong",Toast.LENGTH_LONG).show
                            ();
164                     }
165                 }
166                 else{
167                     final SharedPreferences.Editor ed = sp.edit();
168                     reference = Constants.getDatabase().getReference().
                        child("users").child(mail);
169                     reference.addListenerForSingleValueEvent(new
                        ValueEventListener() {
170                         @Override
171                         public void onDataChange(DataSnapshot dataSnapshot
                            ) {
172                             user = dataSnapshot.getValue(User.class);
173                             ed.putString("username", user.getName());
174                             ed.putString("encodedEmail", user.
                                getEncodedEmail());
175                             ed.commit();
176                             Intent i = new Intent(BaseActivity.this,
                                MainActivity.class);
177                             i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
                                Intent.FLAG_ACTIVITY_CLEAR_TASK);
178                             progressDialog.hide();
179                             startActivity(i);
180                         }
181
182                         @Override
183                         public void onCancelled(DatabaseError
                            databaseError) {
184                             progressDialog.hide();
185                             Toast.makeText(BaseActivity.this, "Some Error
                                Occured",Toast.LENGTH_SHORT).show();
186                         }
187                     });
188                 }
            }

```

```

189         }
190     });
191 }
192
193 @Override
194 public boolean onCreateOptionsMenu(Menu menu) {
195     getMenuInflater().inflate(Constants.menu, menu);
196     return true;
197 }
198 @Override
199 public boolean onOptionsItemSelected(MenuItem item) {
200     int id = item.getItemId();
201     if(id == R.id.logout){
202         signOutUser();
203     }
204     else if(id == R.id.add_user_to_list){
205         Intent i = new Intent(BaseActivity.this, AddUserToList.class);
206         startActivity(i);
207     }
208     return super.onOptionsItemSelected(item);
209 }
210 protected void signOutUser(){
211     auth.signOut();
212     SharedPreferences.Editor editor = sp.edit();
213     editor.putString("encodedEmail", null);
214     editor.putString("username", null);
215     editor.putString("currentWallet", null);
216     editor.putString("currentWalletName", null);
217     editor.putString("currentWalletOwner", null);
218     editor.commit();
219     user = null;
220     getToLoginScreen();
221 }
222
223 }

```
