

Android Programming

Assignment 9

In Class Exercise Notes

Please follow the below guidelines to configure Firebase, get data from RecyclerView Adapter and implement login mechanism

A) Steps to Configure FirebaseAuth

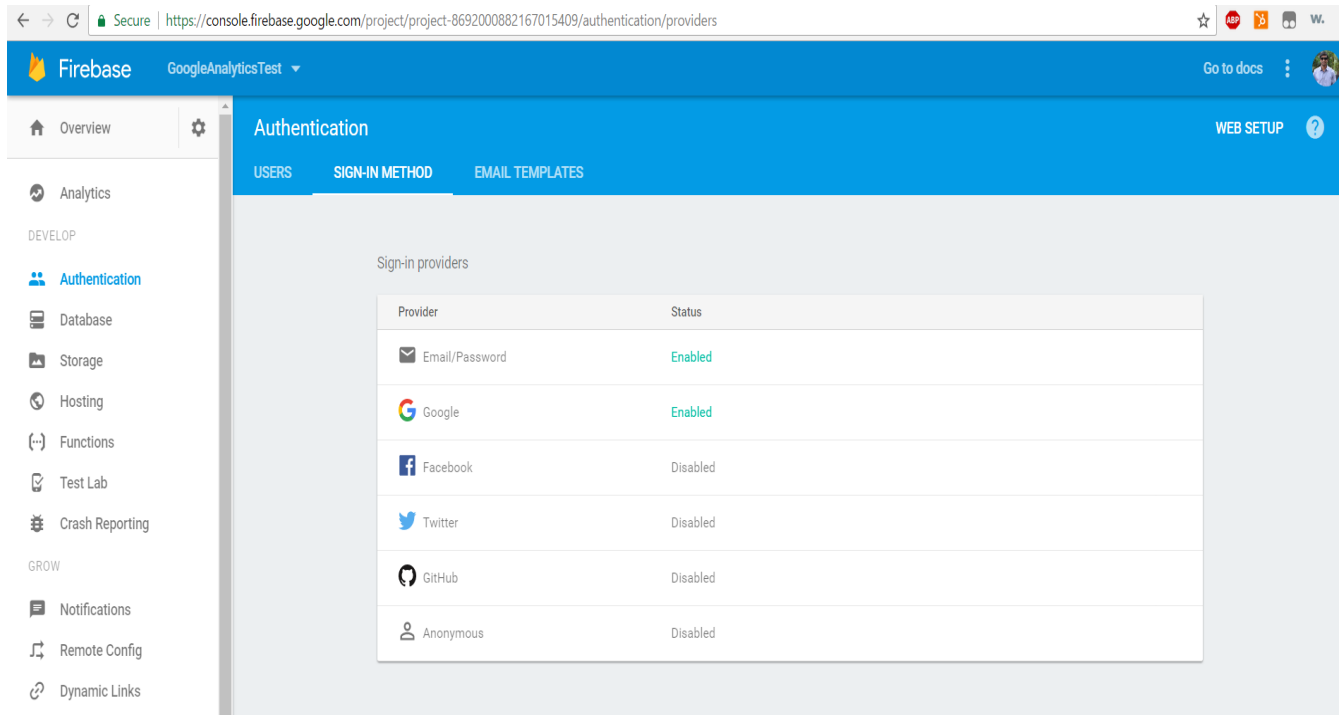
1. Signup at Firebase and create a new app by providing the app name and url. Paste the .json file created by firebase to your project's folder under app directory.

Link: <https://www.firebase.com/>

- a. Add SHA finger print to your app while adding an app to your project on Firebase console. Refer to the link below for getting SHA fingerprint

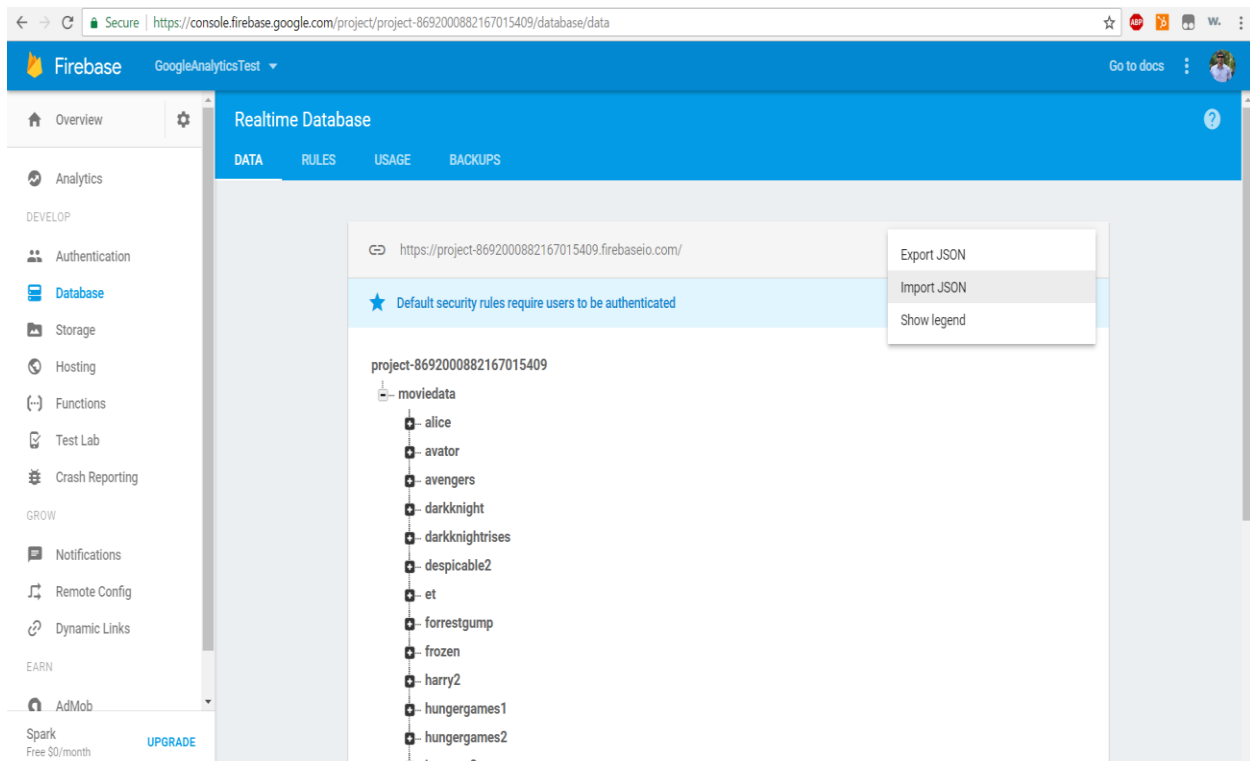
<http://stackoverflow.com/questions/27609442/how-to-get-the-sha-1-fingerprint-certificate-in-android-studio-for-debug-mode>

- b. Once you create project and add an android app, you need to go to Authentication tab and set up sign in method. Enable both Google and Email authentication.



2. Download the below movie.json file and then select the newly created app and click on 'Import Data' in Firebase and upload json file

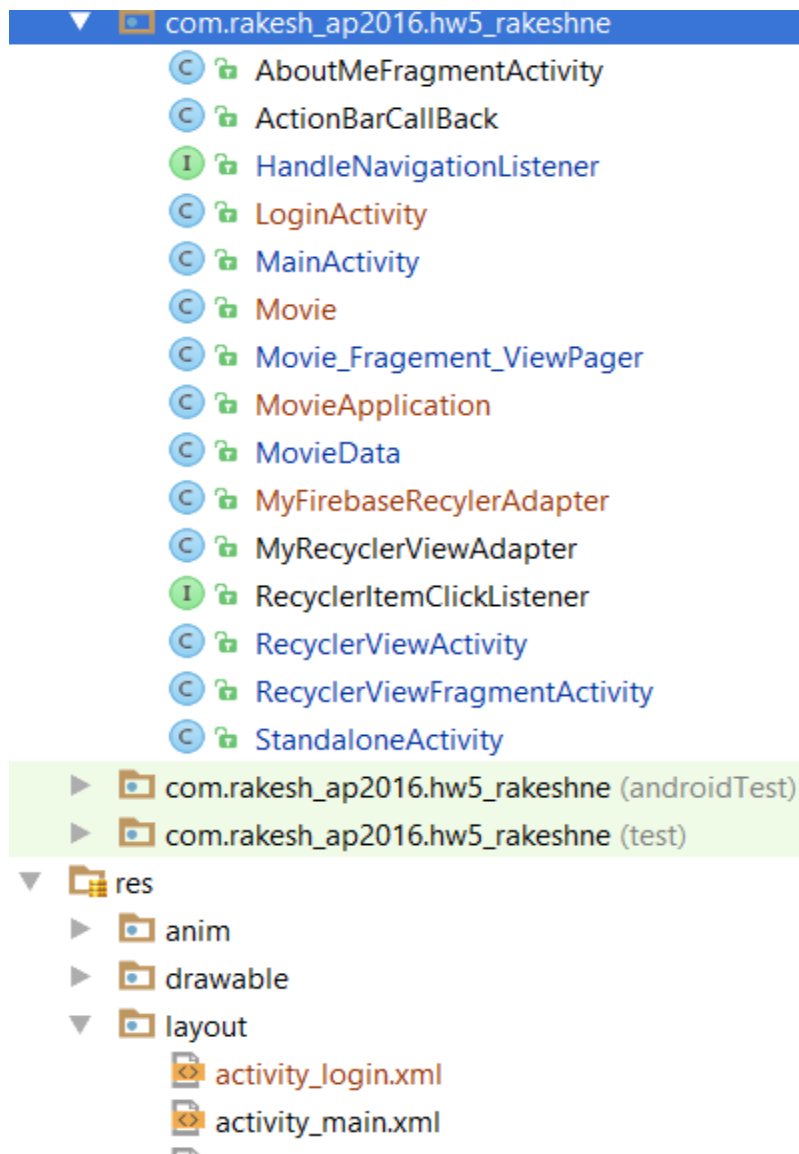
File: **movies.json**



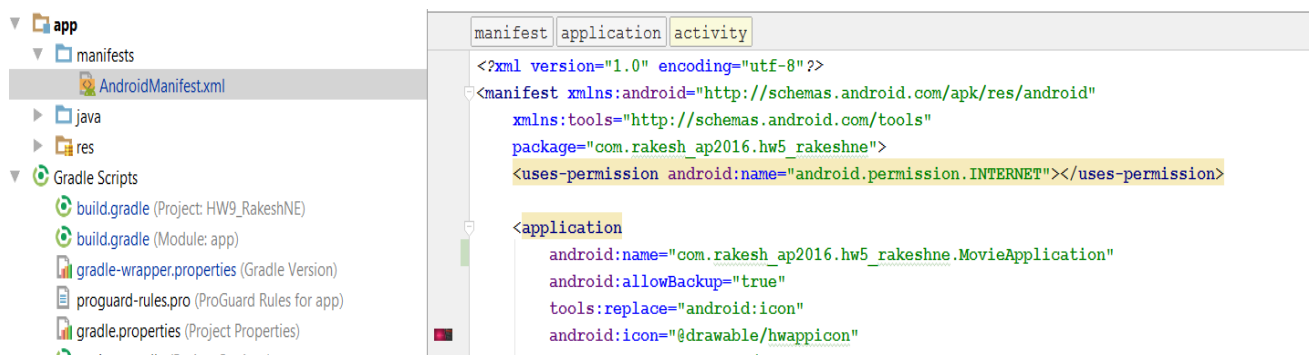
3. Start with a copy of recycler view homework 5 (not async homework) as a starter code for this Homework

4. Download the below starter code and MoveData.java and copy it to respective folder inside android studio project.

File: **starter_code.zip** and **MovieData.java**



5. Update the AndroidManifest.xml file by adding the MovieApplication Package Path inside the application tag and also provide Internet permission in your manifest file



6. Update the AndroidManifest.xml by adding the LoginActivity and make it as Launcher activity.

```
<activity
    android:name=".MainActivity"
    android:configChanges="orientation|screenSize"
    android:label="HW9_RakeshNE"
    android:theme="@style/AppTheme.NoActionBar">

</activity>
<activity android:name=".LoginActivity"
    android:theme="@style/AppTheme.NoActionBar"
    android:configChanges="orientation|screenSize">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

7. Add dependencies:

Please use the updated in your dependencies. If you update the gradle plugin or android sdk tools then you will have to replace some dependencies in your file. Check the latest version of SDK tools and add them. **(Note: Everytime you update anything in SDK manager, make sure that all your dependencies are updated. Else it throws error saying version conflict)**

-Add the following dependencies in build.gradle(Module.app)

```
dependencies {

compile 'com.squareup.picasso:picasso:2.5.2'
compile 'com.firebaseui:firebase-ui:1.2.0'
compile 'com.google.android.gms:play-services:10.2.1'
compile 'com.google.android.gms:play-services-auth:10.2.1'
compile 'com.google.firebase:firebase-core:10.2.1'
compile 'com.google.firebase:firebase-auth:10.2.1'
compile 'com.google.firebase:firebase-database:10.2.1'
```

-Add packaging options to build.gradle(Module.app) under android section.

```
android {
```

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

-Add this line at the end of your build.gradle(Module.app) file

```
apply plugin: 'com.google.gms.google-services'
```

-Add the following repositories tag in your build.gradle(Module.app). It will not be present, you need to type the following lines after your dependencies tag

```
repositories {  
    maven {  
        url "https://jitpack.io"  
    }  
    jcenter{  
        url "http://jcenter.bintray.com/"  
    }  
    maven { url 'https://maven.fabric.io/public' }  
}
```

-Add this line in your build.gradle(Project:hw9) under dependencies tag

```
classpath 'com.google.gms:google-services:3.0.0'
```

Now your project should successfully build without any errors.

IMPORTANT:

GenyMotion:

Before running your app please install Google play services to the GenyMotion and update it once you installed on to your emulator. Refer to the link below for help:

https://github.com/codepath/android_guides/wiki/Genymotion-2.0-Emulators-with-Google-Play-support

Android Emulator:

You can check your notification bar for updating Google play services. If you don't have any such notification in your emulator then please refer to the link below:

<http://stackoverflow.com/questions/35476182/updating-google-play-services-in-emulator?rq=1>

Once you completed installing and updating your Google Play services, you can try running your app. You will be able to use google sign-in method only if you register your Gmail account in your emulator.

8. When you run your app, Login screen should be displayed first. You create a user or you can login directly through email if you have added a user in the firebase manually. On successful login, it should take to the main activity like your assignment 5 app.

9. Add a new item to your navigation drawer called "Logout". And in your "onNavigatedItemSelected(Menuitem item), add the below code to logout from Firebase authentication and you are taken back to Login screen again.

```

FirebaseAuth auth = FirebaseAuth.getInstance();
auth.signOut();
Intent intent = new Intent(this, LoginActivity.class);
startActivity(intent);

```

10. Changes in MyFirestoreRecyclerViewAdapter.

```

public void SetOnItemClickListener(final RecyclerViewItemClickListener mItemClickListener){
    this.mItemClickListener = mItemClickListener;
}

@Override
protected void populateViewHolder(MovieViewHolder movieViewHolder, Movie movie, int i) {

    //TODO: Populate viewHolder by setting the movie attributes to cardview fields
    //movieViewHolder.nameTV.setText(movie.getName());
}

//TODO: Populate ViewHolder and add listeners.
public static class MovieViewHolder extends RecyclerView.ViewHolder {
    //TODO: instantiate all your text views and image view
    //TODO: Add on click listeners like your assignment 5 implementation
    // Its mostly copy paste work for this class
    public MovieViewHolder(View v) {
        super(v);
    }
}

```

11. Initialization of Firestore adapter in your fragment (onCreateView()) / activity (onCreate())

I have initialized my recycler view in a fragment. If you have done in an activity then there will be small changes with respect to getting context objects, etc

```

DatabaseReference childRef =
FirebaseDatabase.getInstance().getReference().child("moviedata").getRef();

myFirestoreRecyclerViewAdapter = new MyFirestoreRecyclerViewAdapter(Movie.class,
R.layout.layout_card_view, MyFirestoreRecyclerViewAdapter.MovieViewHolder.class,
childRef, getContext());

movieData = new MovieData();
mRecyclerView = (RecyclerView) rootView.findViewById(R.id.cardList); //recyclerview

```

```

mLayoutManager = new LinearLayoutManager(getActivity());
mRecyclerView.setLayoutManager(mLayoutManager);

mRecyclerView.setAdapter(myFirebaseRecyclerAdapter);

if (movieData.getSize() == 0) {
    movieData.setAdapter(myFirebaseRecyclerAdapter);
    movieData.setContext(getActivity()); //getApplicationContext()-activity is used
    movieData.initializeDataFromCloud();
}

```

12. Adding FirebaseRecyclerAdapter click listener – onItemClick and onItemLongClick

```

myFirebaseRecyclerAdapter.setOnItemClickListener(new RecyclerViewItemClickListener() {
    @Override
    public void onItemClick(View v, final int position) {
        //handleNavigationListener.navigateToCardSelected(v, position);
        HashMap<String, ?> movie = (HashMap<String, ?>) movieData.getItem(position);
        String id = (String) movie.get("id");
        DatabaseReference ref = movieData.getFirebaseRef();
        ref.child(id).addListenerForSingleValueEvent(new com.google.firebase.database.ValueEventListener() {
            @Override
            public void onDataChange(com.google.firebase.database.DataSnapshot dataSnapshot) {
                HashMap<String, String> movie = (HashMap<String, String>) dataSnapshot.getValue();
                handleNavigationListener.navigateToMovieSelected(position, movie);
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {
                Log.d("My Test", "The Read failed: " + databaseError.getMessage());
            }
        });
    }

    @Override
    public void onItemLongClick(View v, int position) {
        //getActivity().startActionMode(new ActionBarCallBack(position,myRecyclerViewAdapter,movieData.getMoviesList()));
    }
}

```

Handle navigation listener will make my code to replace the movie detail fragment at the activity level. I have implemented this way. You can follow whatever you had done in assignment5 and make the necessary changes.

My navigateToMovieSelected(position,movie) implementation at the activity level will look like this:

```

@Override
public void navigateToMovieSelected(int position, HashMap movie) {
    getSupportFragmentManager().beginTransaction()
        .replace(R.id.recyclerContainer, Movie_Fragment_ViewPager.newInstance(movie))
        .addToBackStack("store")
        .commit();
}

```


Where Movie_Fragment_ViewPager is my movie detail fragment class.

13. Handling pop up menu clicks for duplicate and delete buttons for your movie cards on recycler view, onOverflowMenuClick(View v, int position) implementation

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    HashMap movie;
    switch (item.getItemId()) {
        case R.id.duplicate:
            movie = (HashMap) ((HashMap) movieData.getItem(position)).clone();
            movie.put("id", ((String) movie.get("id") + "_new"));
            movieData.addItemToServer(movie);

            return true;
        case R.id.delete:
            movie = (HashMap) ((HashMap) movieData.getItem(position));
            movieData.removeItemFromServer(movie);

            return true;
        default:
            return false;
    }
}
```

14. Changes in MovieData class

Please download the MovieData.class from blackboard

It has all the functions required for handling Firestore database