WEB ICP10

Venubabu Linga – vl6hw@umsystem.edu

GitHub : https://github.com/VenubabuLinga/WebDevCourse/tree/main/Moblie_dev/ICP10

Sarath Chandra kunisetty – skkh2@umsystem.edu

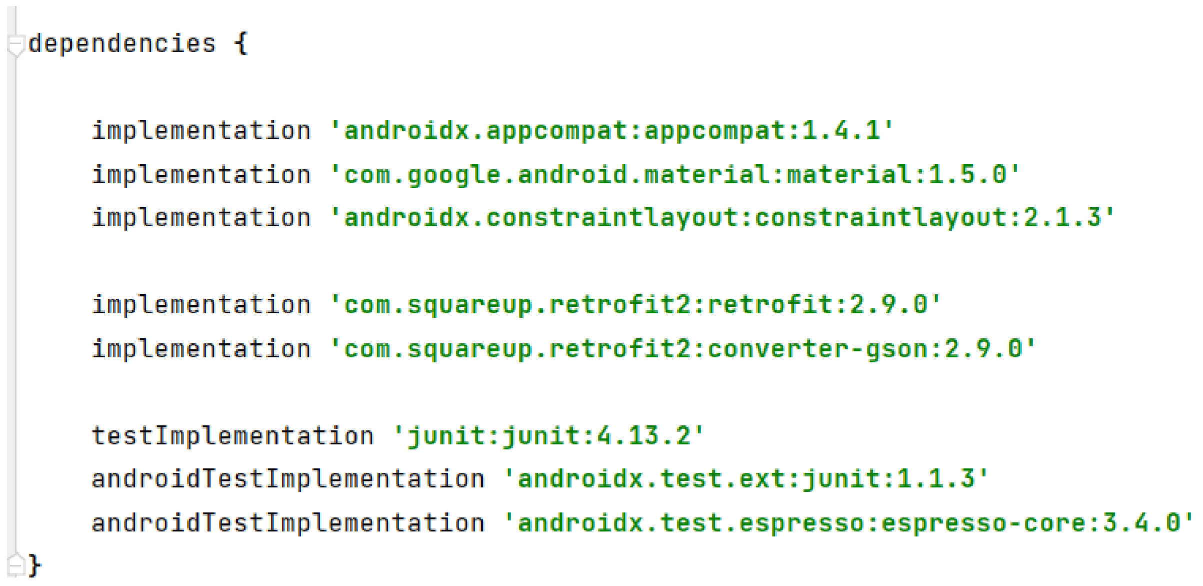
GitHub link: https://github.com/kunisettysarath/WebMobileProgramming-Spring22/tree/main/MobileDevelopment/ICP's/ICP10

In this ICP, we are going to use the Retrofit HTTP client which turns the API into java interface and the JSONPlace holder to provides API an interface for fetching resources.

Retrofit : An HTTP client for Android and Java that is type-safe.

JSONPlace holder : For testing and prototyping, we can use a free dummy API.

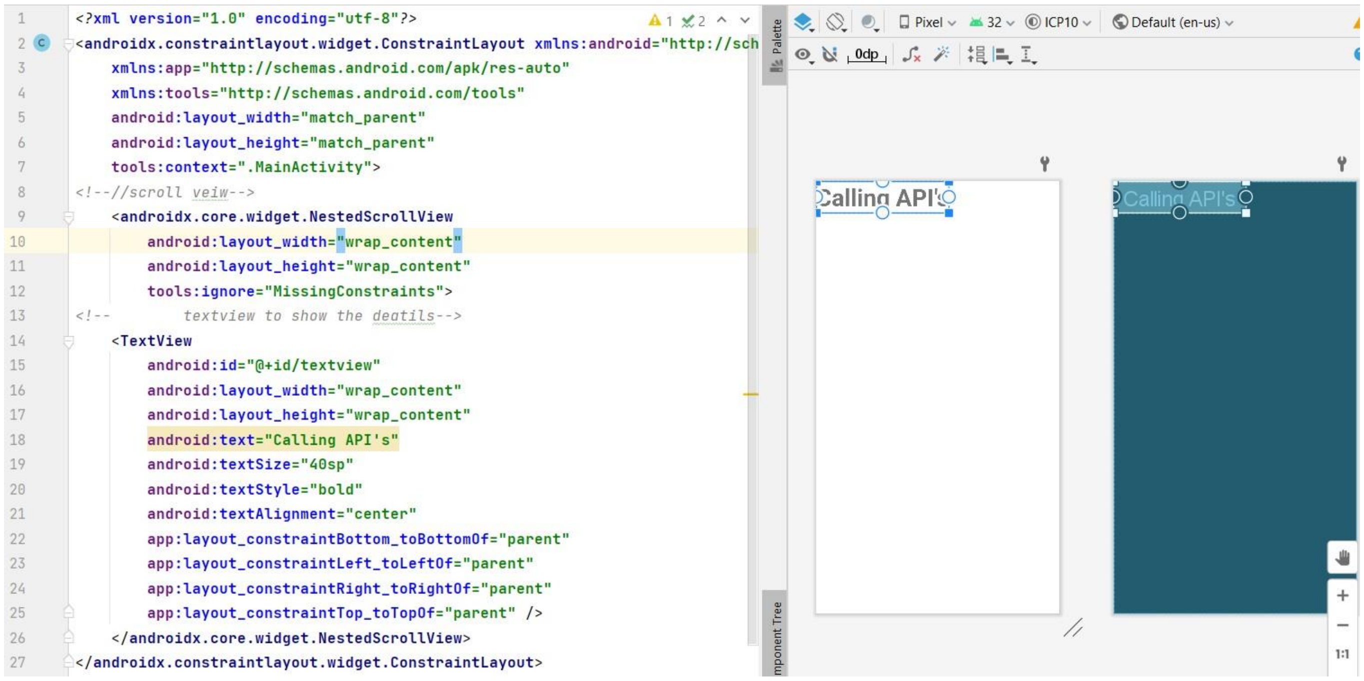
Firstly, we are importing the dependencies for Retrofit and JSONPlaceHolder.



```
dependencies {

    implementation 'androidx.appcompat:appcompat:1.4.1'
    implementation 'com.google.android.material:material:1.5.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.3'

    implementation 'com.squareup.retrofit2:retrofit:2.9.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.9.0'

    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
}
```

1. **Activity_main.xml**

In main activity file we have added the text view in NestedScrollView to show the details from the API call.



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://sch
    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"
    tools:context=".MainActivity">
<!--//scroll veiw-->
    <androidx.core.widget.NestedScrollView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:ignore="MissingConstraints">
<!--        textview to show the deatils-->
    <TextView
        android:id="@+id/textview"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Calling API's"
        android:textSize="40sp"
        android:textStyle="bold"
        android:textAlignment="center"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    </androidx.core.widget.NestedScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>
```

2. MainActivity.java :

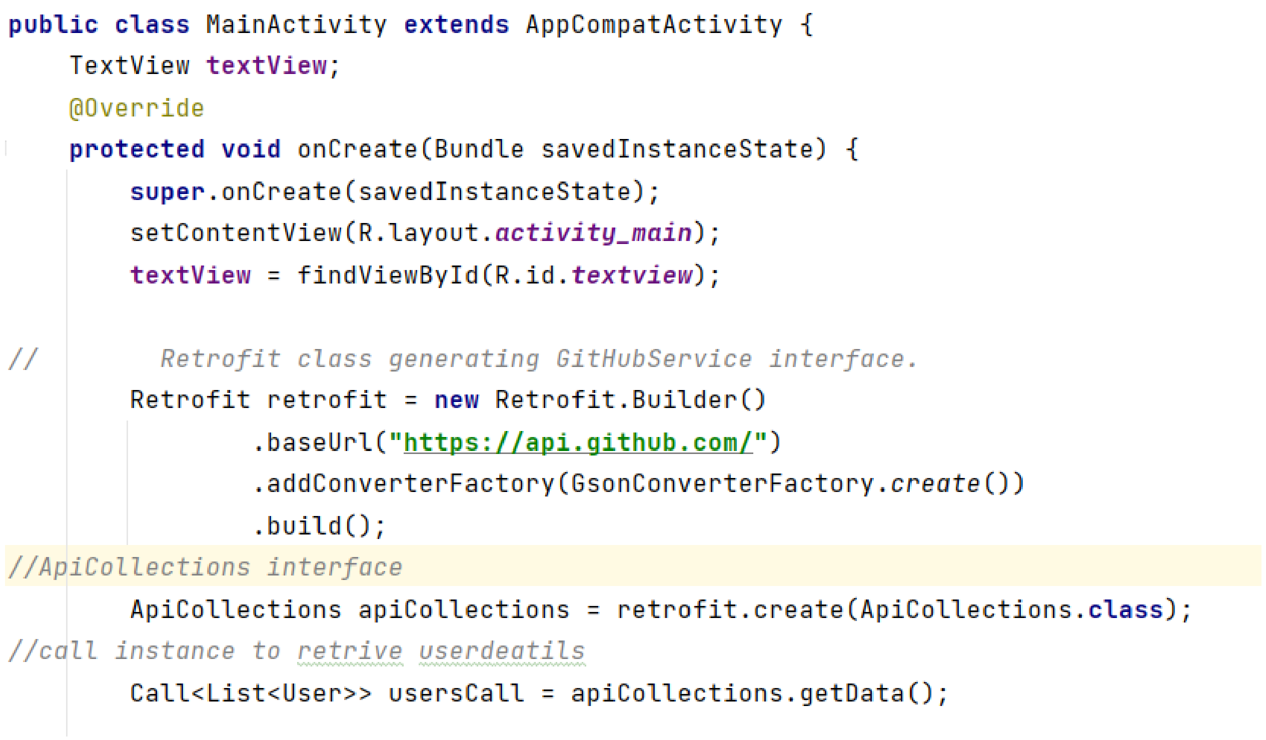
In this java class we have used the Retrofit class to generate the GitHubService interface.

Also we do used the ApiCollections interface and also the call instance to retrive user details.

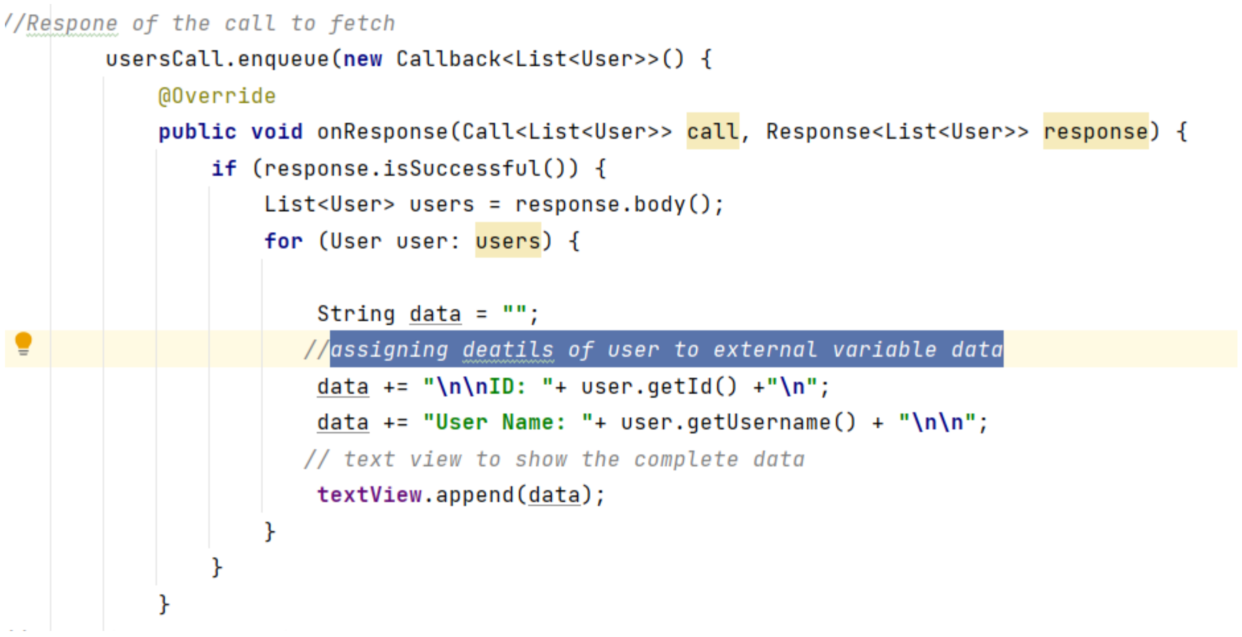
assigning details of user’s from the API call from the response to the external variable data.

We are showing the Data into the Text View of the activity.

Input Code:

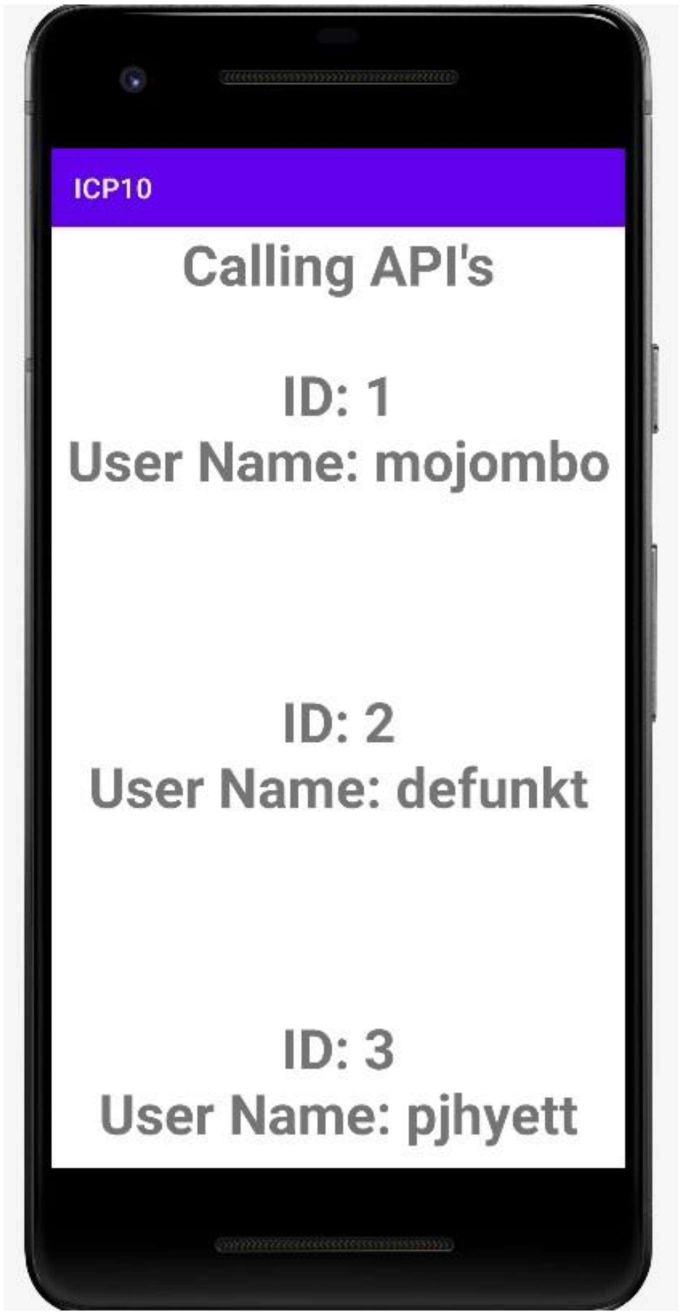


```java
public class MainActivity extends AppCompatActivity {
    TextView textView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textView = findViewById(R.id.textview);

//          Retrofit class generating GitHubService interface.
        Retrofit retrofit = new Retrofit.Builder()
                .baseUrl("https://api.github.com/")
                .addConverterFactory(GsonConverterFactory.create())
                .build();
//ApiCollections interface
        ApiCollections apiCollections = retrofit.create(ApiCollections.class);
//call instance to retrive userdeatils
        Call<List<User>> usersCall = apiCollections.getData();
```



```java
//Respone of the call to fetch
        usersCall.enqueue(new Callback<List<User>>() {
            @Override
            public void onResponse(Call<List<User>> call, Response<List<User>> response) {
                if (response.isSuccessful()) {
                    List<User> users = response.body();
                    for (User user: users) {

                        String data = "";
                       //assigning deatils of user to external variable data
                        data += "\n\nID: "+ user.getId() +"\n";
                        data += "User Name: "+ user.getUsername() + "\n\n";
                       // text view to show the complete data
                        textView.append(data);
                    }
                }
            }
```

<u>Ouput :</u>



ICP10

Calling API's

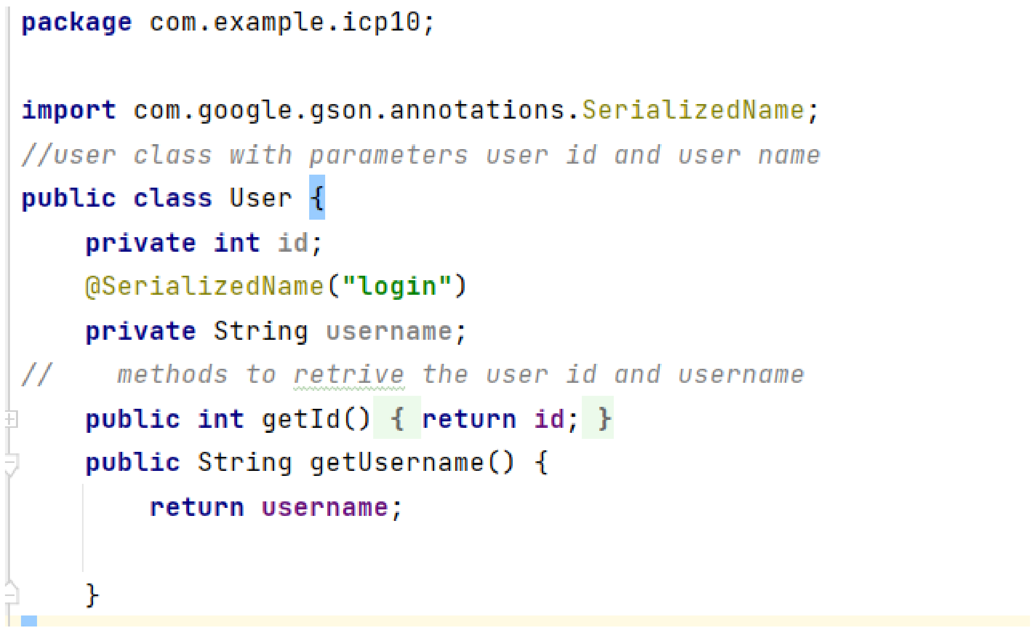
ID: 1
User Name: mojombo

ID: 2
User Name: defunkt

ID: 3
User Name: pjhyett

User.Java :

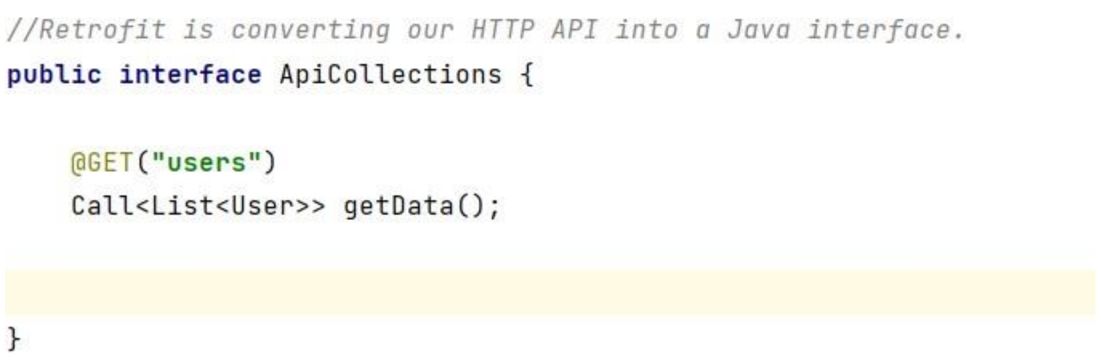
It has the methods to retrieve the user ID and user Name.



```java
package com.example.icp10;

import com.google.gson.annotations.SerializedName;
//user class with parameters user id and user name
public class User {
    private int id;
    @SerializedName("login")
    private String username;
//    methods to retrive the user id and username
    public int getId() { return id; }
    public String getUsername() {
        return username;

    }
```

ApiCollections.Java:

An interface with call instance.



```java
//Retrofit is converting our HTTP API into a Java interface.
public interface ApiCollections {

    @GET("users")
    Call<List<User>> getData();


}
```

Conclusion:

From this ICP we came to know how to use the HTTP API's in android applications to fetch the details using the Retrofit and show in Android page.