

Luke Talcott – [ldt9gn@umsystem.edu](mailto:ldt9gn@umsystem.edu)

GitHub link:

[https://github.com/LukeT11/WebMobileProgramming/tree/main/Mobile\\_Lessons/Mobile\\_ICP3](https://github.com/LukeT11/WebMobileProgramming/tree/main/Mobile_Lessons/Mobile_ICP3)

Mobile ICP 3

## RESTful API and Retrofit

### Introduction

RESTful API (Representational State Transfer) is an application programming interface that uses a specific architectural style for fetching information from a remote service. RESTful API uses HTTP requests to GET, PUT, POST, and DELETE data where a browser sends requests to and receives responses from a server.

Retrofit is an easy way to create a type-safe HTTP client for Android and Java and turns your HTTP API into a Java interface. It can be used to retrieve and upload JSON.

GSON is a Java Library that can be used to serialize and deserialize Java Objects and convert them into their JSON representation and JSON strings into a Java Object.

### Tasks

Mobile ICP3:

- Implementation of Retrofit and GSON converters
- Uses permission for internet
- Uses retrofit request Method @GET for “users”
- Java class ‘user’ to serialize login id and username
- GSONConverterFactory class to generate an implementation of GitHub API
- Call ‘user’ class and outputs the response request as a list outputting the user id and username as a text

## Activity Main XML

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity">
8
9      <TextView
10         android:id="@+id/text_view"
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:text="Hello World!"
14         app:layout_constraintBottom_toBottomOf="parent"
15         app:layout_constraintLeft_toLeftOf="parent"
16         app:layout_constraintRight_toRightOf="parent"
17         app:layout_constraintTop_toTopOf="parent" />
18
19  </androidx.constraintlayout.widget.ConstraintLayout>
```

## Android Manifest XML

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.example.restful">
4
5      <uses-permission android:name="android.permission.INTERNET"/>
6
7      <application
8         android:allowBackup="true"
9         android:icon="@mipmap/ic_launcher"
10        android:label="RESTful"
11        android:roundIcon="@mipmap/ic_launcher_round"
12        android:supportRtl="true"
13        android:theme="@style/Theme.RESTful">
14        <activity
15            android:name=".MainActivity"
16            android:exported="true">
17            <intent-filter>
18                <action android:name="android.intent.action.MAIN" />
19
20                <category android:name="android.intent.category.LAUNCHER" />
21            </intent-filter>
22        </activity>
23    </application>
```

## Java

```
1  package com.example.restful;
2
3  import ...
4
17
18  public class MainActivity extends AppCompatActivity {
19
20      private TextView textView;
21
22      @Override
23      protected void onCreate(Bundle savedInstanceState) {
24          super.onCreate(savedInstanceState);
25          setContentView(R.layout.activity_main);
26
27          textView = findViewById(R.id.text_view);
28
29          Retrofit retrofit = new Retrofit.Builder()
30              .baseUrl("https://api.github.com/")
31              .addConverterFactory(GsonConverterFactory.create())
32              .build();
33
34          AppCollection apiCollection = retrofit.create(AppCollection.class);
35
36          Call<List<user>> userCall = apiCollection.getData();
37
38          userCall.enqueue(new Callback<List<user>>() {
39
40              @Override
41              public void onResponse(Call<List<user>> call, Response<List<user>> response) {
42
43                  if (response.isSuccessful()) {
44                      List<user> users = response.body();
45
46                      for (user user: users) {
47                          String data = "";
48
49                          data += "ID: " + user.getId() + "/n";
50                          data += "ID: " + user.getUserName() + "/n/n";
51
52                          textView.append(data);
53                      }
54                  }
55              }
56
57              @Override
58              public void onFailure(Call<List<user>> call, Throwable t) {
59                  Toast.makeText(context: MainActivity.this, text: "Error", Toast.LENGTH_SHORT).show();
60              }
61          });
62      }
```

```

1 package com.example.restful;
2
3 import ...
4
5
6
7
8
9 public interface AppCollection {
10
11     @GET("users")
12     Call<List<user>> getData();
13 }

```

```

1 package com.example.restful;
2
3 import ...
4
5
6
7 public class user {
8
9     private int id;
10
11     @SerializedName("Login")
12     private String userName;
13
14     public int getId() { return id; }
15
16
17     public String getUserName() { return userName; }
18
19 }
20
21
22

```

## Gradle

```

35
36     implementation 'com.squareup.retrofit2:retrofit:2.9.0'
37     implementation 'com.squareup.retrofit2:converter-gson:2.9.0'
38

```

## Contribution

I worked independently, so I am the sole contributor.

## Conclusion

I used RESTful API HTTP request @GET to request and receive the 'login' info of a GitHub user. I used retrofit's GSON converter to generate and use GitHub service API and to serialize and deserialize Java Objects such the GitHub id and login in as the user id's. Then I used a Java class to output the received data of the user's GitHub data and output it as a text within a list.