Luke Talcott – ldt9gn@umsystem.edu

GitHub link:
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.

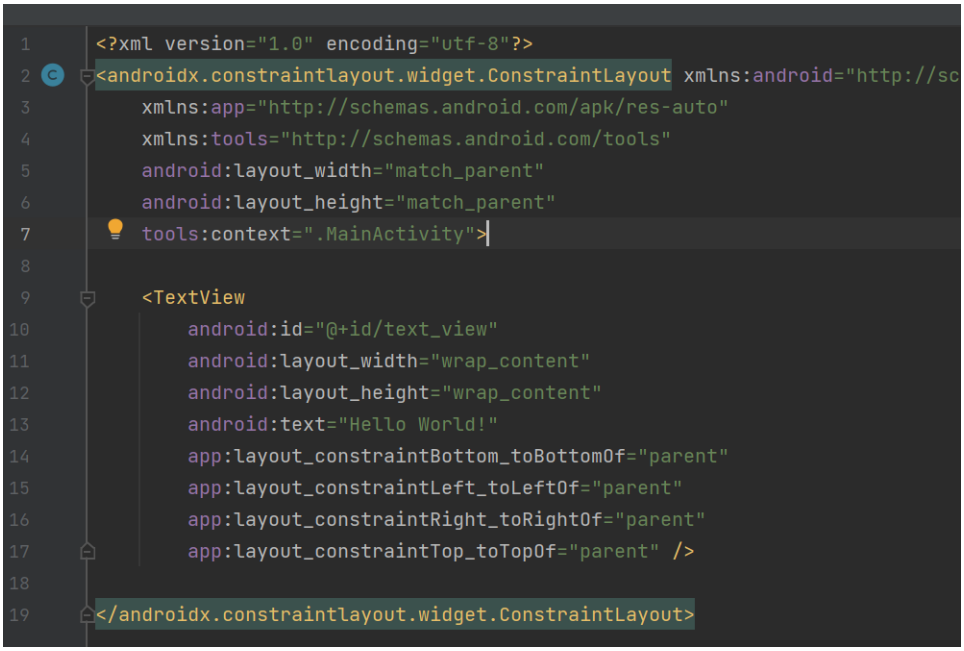
GSON is a Java Library that can used to serialize and deserialize Java Objects and convert them into their JSON representation and JSON strings into a Java Object.

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

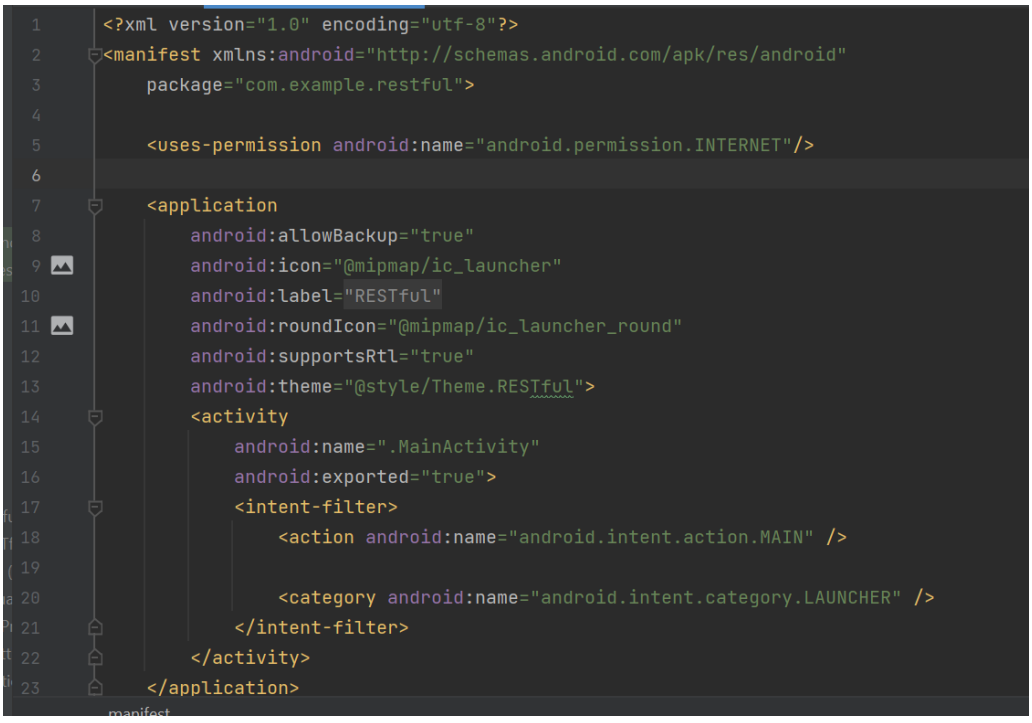


```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://sc
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/text_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Android Manifest XML



```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.restful">

    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="RESTful"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.RESTful">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
```

Java

```java
package com.example.restful;

import ...

public class MainActivity extends AppCompatActivity {

    private TextView textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        textView = findViewById(R.id.text_view);

        Retrofit retrofit = new Retrofit.Builder()
                .baseUrl("https://api.github.com/")
                .addConverterFactory(GsonConverterFactory.create())
                .build();

        AppCollection apiCollection = retrofit.create(AppCollection.class);

        Call<List<user>> userCall = apiCollection.getData();

        userCall.enqueue(new Callback<List<user>>() {
```

```java
        userCall.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 = "";

                        data += "ID: " + user.getId() + "/n";
                        data += "ID: " + user.getUserName() + "/n/n";

                        textView.append(data);
                    }
                }
            }

            @Override
            public void onFailure(Call<List<user>> call, Throwable t) {
                Toast.makeText( context: MainActivity.this,  text: "Error", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

```java
package com.example.restful;

import ...

public interface AppCollection {

    @GET("users")
    Call<List<user>> getData();
}
```

```java
package com.example.restful;

import ...

public class user {

    private int id;

    @SerializedName("Login")
    private String userName;

    public int getId() { return id; }

    public String getUserName() { return userName; }
}
```

Gradle

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

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.