6 Integrating a REST API into Android Application

Objectives

This experiment teaches you how to integrate Android apps with RESTful web services and how to access data from them using simple GET and POST requests.

Introduction

REST describes a set of architectural principles by which data can be transmitted over a standardized interface (such as HTTP). The acronym REST stands for Representational State Transfer, this basically means that each unique URL is a representation of some object.

Exposing a system's resources through a RESTful API is a flexible way to provide different kinds of applications with data formatted in a standard way.

To find any resource you need on the Internet, you can use the URL (Uniform Resource Locator) class. The constructor of this class takes string parameter of the following structure:

```
    Protocol://host:port/path
EX  http://www.mocky.io/v2/583fd94e2400005015 83b533
```

In this experiment, we will request the API from the server using **Retrofit** RESTful API. So we will build a simple android application to get a JSON object from the server. Then, we will print the data in a textview component. In this experiment, we will get a JSON of the following structure:

```
[
{
"name": ""
"age": ""
"id": ""
}
,{
"name": ""
"age": ""
"id": ""
}
]
```

Retrofit

Retrofit is a REST Client for Android and Java. This library is easy learn and has more features. This is beginner friendly compared to other Networking libraries.You can GET,POST,PUT,DELETE ..etc using this library.
In Retrofit, we need to create 3 classes:
1) **POJO (Plain Old Java Object) or Model Class** – The json retrieved from the server is added to this class.
2) **Interface** : Now we need to create an interface for managing url calls like GET,POST..etc. This is the service class.
3) **RestAdapter Class** : This is RestClient Class. Gson is used in default for the retrofit. You can use setup your own converter for this purpose like jackson .

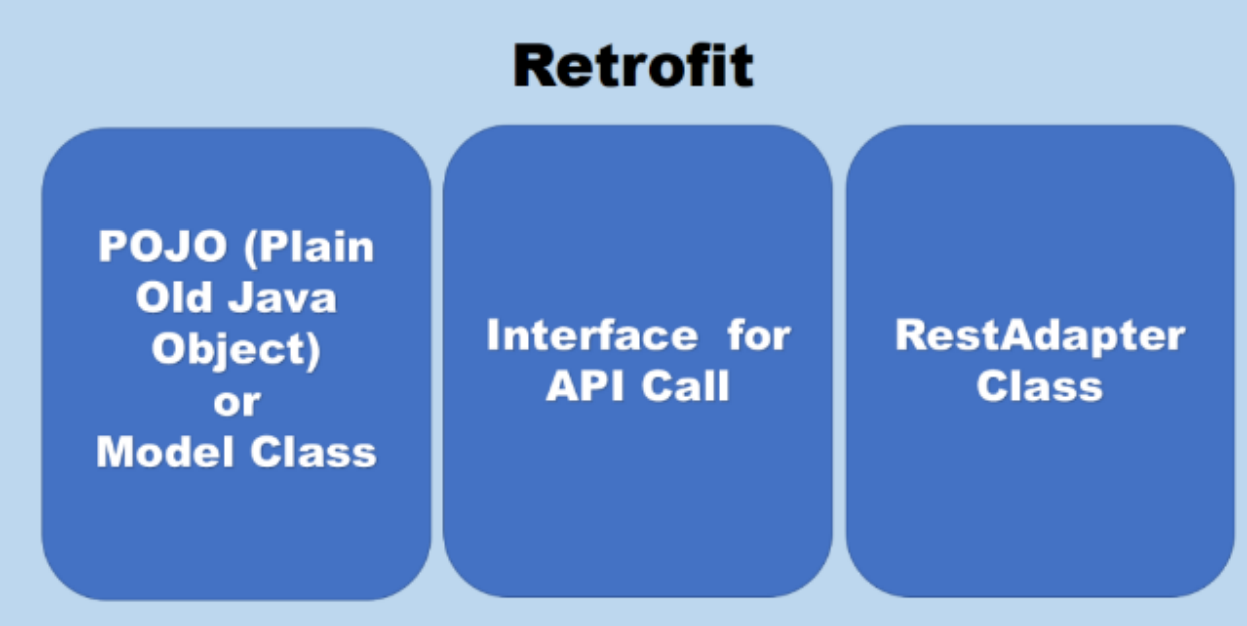


Figure 1

Procedure:

For this lab, we will be creating a "Students Grade" application. It is a simple app that get students information as JSON object from the server and view the students name, ID and grade in textView.

Create a new Android Project:

1. In Android Studio, create a new project:
   - If you don't have a project opened, in the **Welcome to Android Studio** window, click **Start a new Android Studio project**.
   - If you have a project opened, select **File > New Project**.
2. In the New Project screen, enter the following values:
   - Application Name: "Student Grade"
   - Company Domain: "birzeit.edu"
3. Click **Next**.
4. In the **Target Android Devices** screen, keep the default values and click **Next**.
5. In the **Add an Activity to Mobile** screen, select **Empty Activity** and click **Next**.
6. In the **Customize the Activity** screen, keep the default values and click **Next**.

Adding Retrofit Libraries to Project:

1. Open **Gradle Scripts -> build.gradel(Module:app)** file and add the following libraries to **dependencies** section.

```
compile 'com.google.code.gson:gson:2.6.2'
compile 'com.squareup.retrofit2:retrofit:2.1.0'
compile 'com.squareup.retrofit2:converter-gson:2.1.0'
```

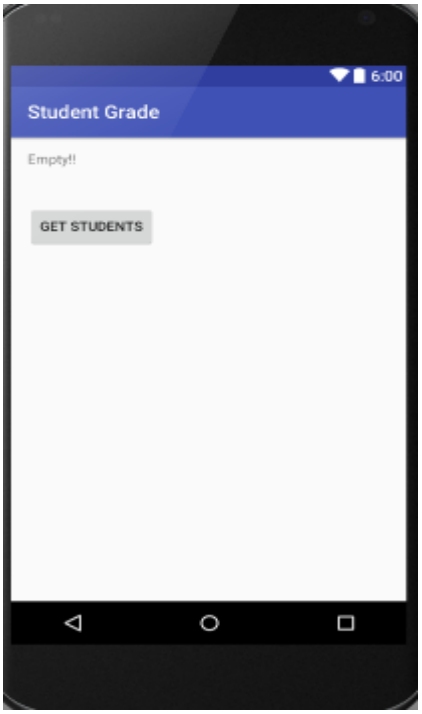
2. Rebuild project: **build -> Rebuild Project**.

Add internet permission to Manifest file:

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

Add TextView and Button to activity_main.xml file:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:orientation="vertical"
    tools:context="edu.birzeit.studentgrade.MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Empty!!"
        android:id="@+id/showStudents" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="get Students"
        android:id="@+id/getStudents"
        android:layout_alignParentStart="true"
        android:layout_marginTop="42dp" />
</LinearLayout>
```



- **Create Student.java class under "edu.birzeit.studentgrade" package:**

```java
package edu.birzeit.studentgrade;
public class Student {
    private  int id;
    private  String name;
    private int grade;
    public Student(int id, String name, int grade) {
        this.id = id;
        this.name = name;
        this.grade = grade;
    }
    public int getId() {return id;}
    public void setId(int id) {this.id = id;}
    public String getName() {return name;}
    public void setName(String name) {this.name = name;}
    public int getGrade() {return grade;}
    public void setGrade(int grade) {this.grade = grade;}
}
```

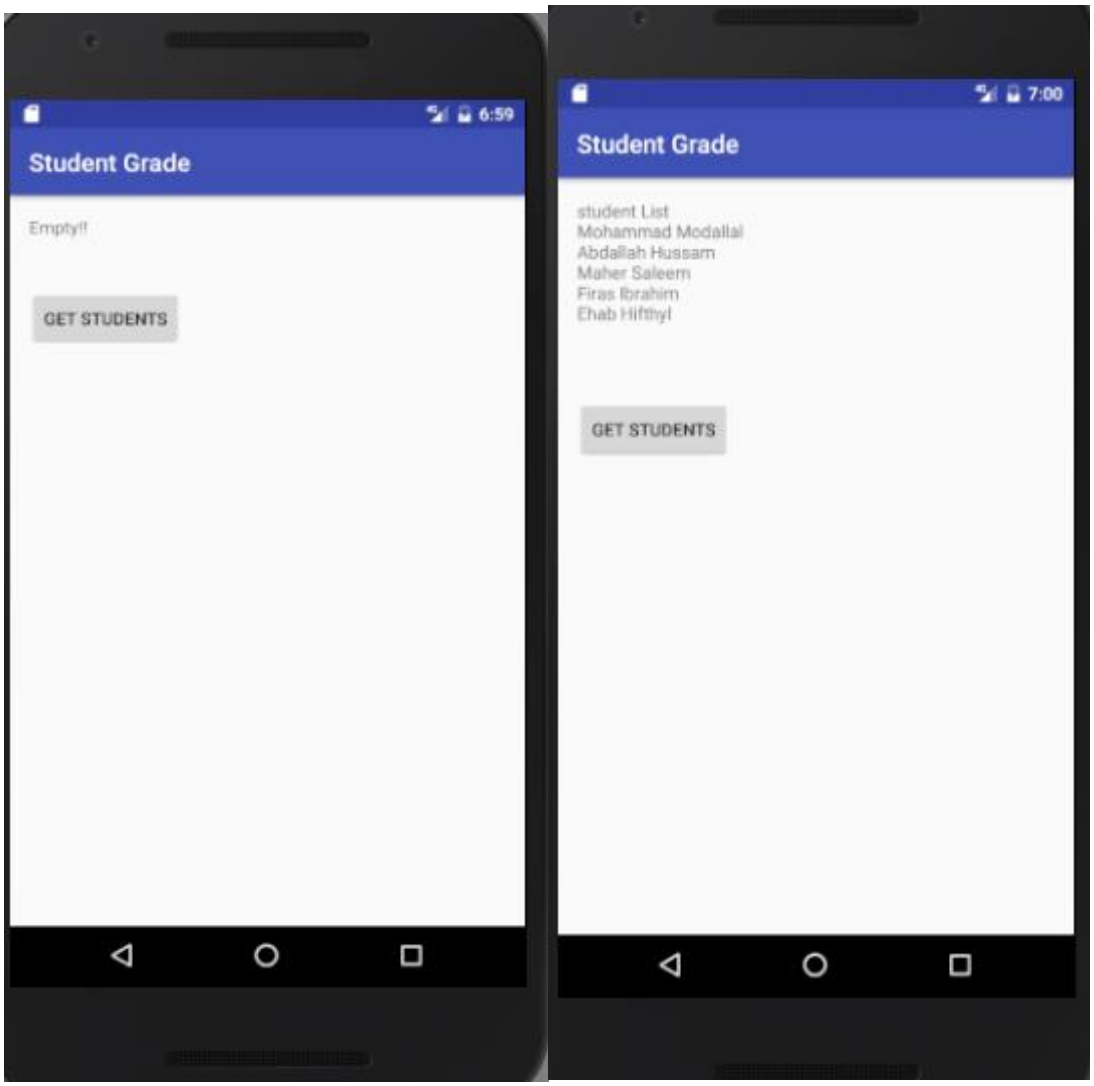
- **Create interface named MyApiEndPointInterface under "edu.birzeit.studentgrade" package** :

```java
package edu.birzeit.studentgrade;
import java.util.List;
import retrofit2.Call;
import retrofit2.http.GET;
import retrofit2.http.Path;
public interface MyApiEndPointInterface {
    @GET("{username}")
    Call<List<Student>> getStudent(@Path("username") String username);
}
```

- **Add the following code to MainActivity.java:**

```
package edu.birzeit.studentgrade;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import java.util.List;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;
public class MainActivity extends AppCompatActivity {
    Button button;
    TextView show;
    private static final String BASE_URL = "http://www.mocky.io/v2/";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        show=(TextView)findViewById(R.id.showStudents);
        button = (Button) findViewById(R.id.getStudents);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                show.setText("student List\n");
                try {
                    Retrofit retrofit = new
Retrofit.Builder().baseUrl(BASE_URL).addConverterFactory(GsonConverterFactory.create()).build
();
                    MyApiEndPointInterface apiService =
retrofit.create(MyApiEndPointInterface.class);
                    Call<List<Student>> repos =
apiService.getStudent("583fd94e240000501583b533");

                    repos.enqueue(new Callback<List<Student>>() {
                        @Override
                        public void onResponse(Call<List<Student>> call,
Response<List<Student>> response) {
                            int statusCode = response.code();
                            List<Student> studentObjects= response.body();
                            for (int i = 0; i < studentObjects.size(); i++) {

show.setText(show.getText()+studentObjects.get(i).getName()+"\n");
                            }
                        }

                        @Override
                        public void onFailure(Call<List<Student>> call, Throwable t) {
                            // Log error here since request failed

Toast.makeText(MainActivity.this,"error",Toast.LENGTH_LONG).show();
                        }
                    });
                }catch(Exception e)
                {
                }
            }
        });
    }
}
```

- **Run the application on the emulator (See Figure below).**



ToDo

This part will be given to you by the teacher assistant in the lab time.