Name : Mujahid Ullah

Subject: Android Development

Submitted To Sir Junaid

Date : 1-Jan-2021

Assignment No 9

Retrofit Plugin Android Studio and Load Images by using glide library.With Step By Step Example

In Android, Retrofit is a REST Client for Java and Android by Square inc under Apache 2.0 license. Its a simple network library that used for network transactions. By using this library we can seamlessly capture JSON response from web service/web API. It's easy and fast library to retrieve and upload the data(JSON or any other structured data) via a REST based web service.

Need of Retrofit In Android:

We have a lot of network libraries that used to fetch and send the data from/to server. In our previous article we use Volley library for network transactions but Retrofit is an ultimate replacement of Volley and all other libraries. Retrofit is better alternative of other libraries in terms of performance, ease of use, extensibility and others.

Difference Between Retrofit and other libraries:

In Android, retrofit library is different from other network libraries because it gives us an easy to use platform through which we don't need to parse JSON responses as they are done by library itself. It used GSON library in the background to parser the response data. What we need to do is define a POJO (Plain Old Java Object) to parse the response.

Performance benchmarks for Android AsyncTask, Volley and Retrofit (in milliseconds, lower value is better):

Here is the main difference between our three mainly used techniques for implementing API's is our android app. You can see the difference in performance that for one discussion means for one network request and response they will take how much time.

1. AsyncTask:

one(1) discussion: 941 ms

Seven(7) discussions: 4539 ms

Twenty Five(25) discussions: 13957 ms

2. Volley:

one(1) discussion: 560 ms

Seven(7) discussions: 2202 ms

Twenty Five(25) discussions: 4275 ms

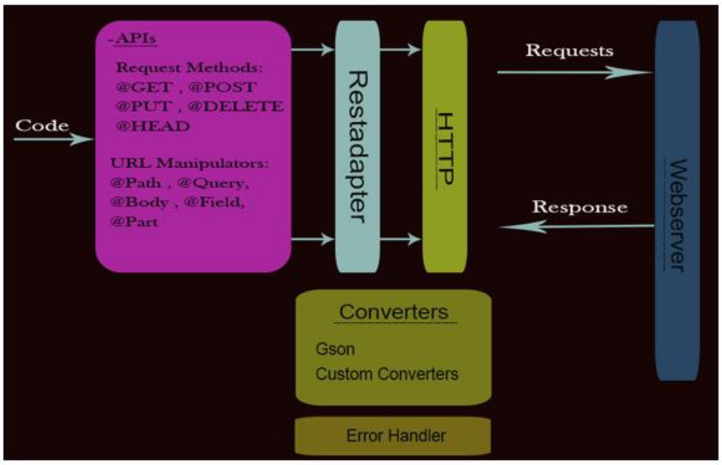
3. Retrofit:

one(1) discussion: 312 ms

Seven(7) discussions: 889 ms

Twenty Five(25) discussions: 1059 ms

Steps to Integrate Retrofit 1.9 in our Android Studio Project.



Example : Retrofit

Step 1: Create a new project And Name It RetrofitExample.

Step 2: Open Gradle Scripts > build.gradle and add Retrofit , RecyclerView and Glide Library dependency in it.

Step 3: Add Internet Permission in the AndroidManifest.xml:

For network transactions we need to define Internet permission in our Manifest file.

Step 4: Open res -> layout -> activity_main.xml (or) main.xml and add following code:

In this step we create a RecyclerView in our XML file.

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

Step 5:MainActivity.java Code:

```
package com.cal.retrofit_glide_recycler_view;


import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ProgressBar;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;
import java.util.Arrays;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class MainActivity extends AppCompatActivity {

    private RecyclerView recyclerView;
    private ArrayList<AndroidVersion> data;
    private DataAdapter adapter;
    private ProgressBar pb;

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

    private void initViews() {
        recyclerView = (RecyclerView) findViewById(R.id.card_recycler_view);
        //recyclerView.addOnItemTouchListener(new RecyclerItemCli);
        recyclerView.setHasFixedSize(true);
        RecyclerView.LayoutManager layoutManager = new
LinearLayoutManager(getApplicationContext());
        recyclerView.setLayoutManager(layoutManager);
        pb = (ProgressBar) findViewById(R.id.pb);
        pb.setVisibility(View.VISIBLE);
        loadJSON();

    }

    private void loadJSON() {
        Retrofit retrofit = new Retrofit.Builder()
                .baseUrl("http://api.learn2crack.com/")
                .addConverterFactory(GsonConverterFactory.create())
                .build();

        final RequestInterface request= retrofit.create(RequestInterface.class);
        Call<JsonResponse> call = request.getJSON();
        call.enqueue(new Callback<JsonResponse>() {
            @Override
            public void onResponse(Call<JsonResponse> call, Response<JsonResponse>
response) {
                if(response.isSuccessful()){
                    pb.setVisibility(View.INVISIBLE);
                    JsonResponse res = response.body();
                    data = new ArrayList<>(Arrays.asList(res.getAndroid()));
                    adapter = new DataAdapter(data);
                    recyclerView.setAdapter(adapter);
                }

            }

            @Override
            public void onFailure(Call<JsonResponse> call, Throwable t) {
                Log.d("Failure", t.getMessage());
            }
        });
    }
}
```

Step 6: activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/card_recycler_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    <ProgressBar
        android:id="@+id/pb"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"/>

</RelativeLayout>
```

Step 7:DataAdapter.java Code:

```
package com.cal.retrofit_glide_recycler_view;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

/**
 * Created by Pronious on 26/09/2017.
 */

public class DataAdapter extends RecyclerView.Adapter<DataAdapter.ViewHolder> {

    private ArrayList<AndroidVersion> android;


    public DataAdapter(ArrayList<AndroidVersion> android) {
        this.android = android;
    }

    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.card_row, parent,
false);
        view.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

            }
        });
        return new ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {

        holder.tv_name.setText(android.get(position).getName());
        holder.tv_version.setText(android.get(position).getVer());
        holder.tv_api.setText(android.get(position).getApi());


    }
    @Override
    public int getItemCount() {
        return android.size();
    }

    public class ViewHolder extends RecyclerView.ViewHolder {
        private TextView tv_name, tv_version, tv_api;
        public ViewHolder(View itemView) {
            super(itemView);

            tv_name = itemView.findViewById(R.id.tv_name);
            tv_version = itemView.findViewById(R.id.tv_version);
            tv_api = itemView.findViewById(R.id.tv_api_level);
        }
    }
}
```

Step 8: Model_Class.java Code:

```
package com.cal.retrofit_glide_recycler_view;


import com.google.gson.annotations.SerializedName;

public class Model_class {

    private String ver;
    private String name;
    private String api;

    public String getVer() {
        return ver;
    }

    public String getName() {
        return name;
    }

    public String getApi() {
        return api;
    }
}
```

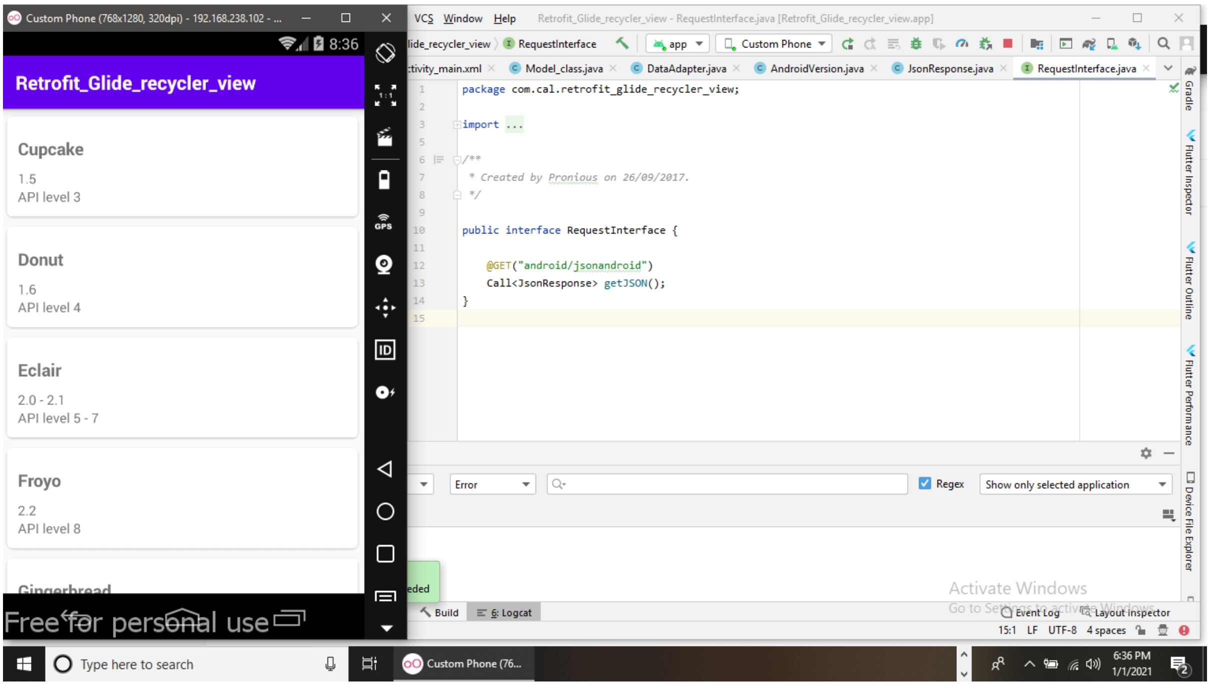
Step 9: JsonResponse.java

```
package com.cal.retrofit_glide_recycler_view;



public class JsonResponse {

private AndroidVersion[] android;

    public AndroidVersion[] getAndroid() {
        return android;
        }
    }
```

Step 10: RequestInterface.java Code:

```
package com.cal.retrofit_glide_recycler_view;

import retrofit2.Call;
import retrofit2.http.GET;

/**
 * Created by Pronious on 26/09/2017.
 */

public interface RequestInterface {

    @GET("android/jsonandroid")
    Call<JsonResponse> getJSON();
}
```

OutPut:



Custom Phone (768x1280, 320dpi) - 192.168.238.102 - ...

Retrofit_Glide_recycler_view

Cupcake
1.5
API level 3

Donut
1.6
API level 4

Eclair
2.0 - 2.1
API level 5 - 7

Froyo
2.2
API level 8

Gingerbread

```
package com.cal.retrofit_glide_recycler_view;

import ...

/**
 * Created by Pronious on 26/09/2017.
 */

public interface RequestInterface {

    @GET("android/jsonandroid")
    Call<JsonResponse> getJSON();
}
```