

## Lab 3: JSON Parser



### MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Hiểu được định dạng dữ liệu JSON
- ✓ Lấy và parser dữ liệu JSON từ Server
- ✓ Sử dụng thư viện của hãng thứ ba

### BÀI 1 (2 ĐIỂM)

Viết chương trình đơn giản lấy dữ liệu json từ url về android và hiển thị danh sách contacts lên listview. Sử dụng JSON parser trong android để thực hiện.

### HƯỚNG DẪN:

- ✓ Mở xampp và start server
- ✓ Tạo thư mục mới đặt tên hoten\_massv, sau đó copy file index.php từ thư mục tài nguyên của lab vào thư mục vừa tạo
- ✓ Tạo mới project trong android studio xây dựng giao diện chưa listview như bên dưới
- ✓ Bởi vì chúng ta get dữ liệu từ JSON file bằng cách gọi HTTP, nên chúng ta cần thêm quyền INTERNET trong AndroidManifest.xml. mở file AndroidManifest.xml và thêm quyền như bên dưới

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

- ✓ Thêm listview vào layout main

```
<?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:id="@+id/activity_main"
```

```

        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="example.com.android.lab3_ex1_contact.MainActivity">

        <ListView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/listContact"
        />
    </LinearLayout>

```

- ✓ Tạo class mới đặt tên `HttpHandler.java` và sử dụng code bên dưới. hàm `makeServiceCall()` sẽ gọi http cho một url cụ thể và get dữ liệu trả về. Trong trường hợp này, chúng ta sử dụng để lấy json từ url

```

package example.com.android.lab3_ex1_contact;

import android.util.Log;

import java.io.BufferedReader;
import java.io.BufferedInputStream;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class HttpHandler {
    private static final String TAG = HttpHandler.class.getSimpleName();

    public HttpHandler() {

    }

    public String makeServiceCall(String reqUrl) {
        String response = null;
        try {
            URL url = new URL(reqUrl);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            // read the response
            InputStream in = new BufferedInputStream(conn.getInputStream());
            response = convertStreamToString(in);
        } catch (Exception e) {
            e.printStackTrace();
            Log.e(TAG, "Exception: " + e.getMessage());
        }

        return response;
    }
}

```

```

private String convertStreamToString(InputStream is) {
    BufferedReader read = new BufferedReader(new InputStreamReader(is));
    StringBuilder sb = new StringBuilder();
    String line;
    try {
        while ((line = read.readLine()) != null) {
            sb.append(line).append("\n");
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            is.close();
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
    return sb.toString();
}
}

```

- ✓ Tạo mới layout đặt tên là list\_item.xml với nội dung bên dưới. file này sẽ được sử dụng để render từng item trong list

```

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

    <TextView
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingBottom="2dip"
        android:paddingTop="6dip"
        android:text="name"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="16sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingBottom="2dip"
        android:text="email"
        android:textColor="@color/colorAccent" />

    <TextView
        android:id="@+id/mobile"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingBottom="2dip"

```

```

        android:text="mobile"
        android:textColor="#5d5d5d" />
    </LinearLayout>

```

- ✓ Mở file MainActivity.java và khai báo cho list view và ảnh xạ.

```
private ListView lvContact;
```

- ✓ Tạo class mới đặt tên Contact.java với code như bên dưới. Đây là model của Contact dựa theo file JSON trả về

```

package example.com.android.lab3_ex1_contact;

public class Contact {
    String id;
    String name, email, address, gender, mobile, home, office;

    public Contact() {

    }

    public Contact(String id, String name, String email, String address,
String gender, String mobile, String home, String office) {
        this.id = id;
        this.name = name;
        this.email = email;
        this.address = address;
        this.gender = gender;
        this.mobile = mobile;
        this.home = home;
        this.office = office;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getEmail() {
        return email;
    }

```

```

    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public String getGender() {
        return gender;
    }

    public void setGender(String gender) {
        this.gender = gender;
    }

    public String getMobile() {
        return mobile;
    }

    public void setMobile(String mobile) {
        this.mobile = mobile;
    }

    public String getHome() {
        return home;
    }

    public void setHome(String home) {
        this.home = home;
    }

    public String getOffice() {
        return office;
    }

    public void setOffice(String office) {
        this.office = office;
    }
}

```

- ✓ Tạo class mới đặt tên ContactAdapter.java với code như bên dưới. Lớp này được sử dụng để đổ dữ liệu lên listview

```

package example.com.android.lab3_ex1_contact;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;

```

## FPT POLYTECHNIC

```

import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.HashMap;
public class ContactAdapter extends BaseAdapter {

    Context context;
    ArrayList<Contact> contactList;

    public ContactAdapter(Context context, ArrayList<Contact> contactList) {
        this.context = context;
        this.contactList = contactList;
    }

    @Override
    public int getCount() {
        return contactList.size();
    }

    @Override
    public Object getItem(int i) {
        return contactList.get(i);
    }

    @Override
    public long getItemId(int i) {
        return 0;
    }

    public static class ViewHolder {
        TextView tvName, tvEmail, tvMobile;
    }

    @Override
    public View getView(int i, View view, ViewGroup viewGroup) {
        LayoutInflater inflater = (LayoutInflater)
context.getSystemService(context.LAYOUT_INFLATER_SERVICE);
        ViewHolder viewHolder;
        if (view == null) {
            viewHolder = new ViewHolder();
            view = inflater.inflate(R.layout.list_item, null);
            viewHolder.tvName = (TextView) view.findViewById(R.id.name);
            viewHolder.tvEmail = (TextView) view.findViewById(R.id.email);
            viewHolder.tvMobile = (TextView) view.findViewById(R.id.mobile);
            view.setTag(viewHolder);
        } else {
            viewHolder = (ViewHolder) view.getTag();
        }
        Contact contact = contactList.get(i);
        viewHolder.tvName.setText(contact.getName());
        viewHolder.tvEmail.setText(contact.getEmail());
        viewHolder.tvMobile.setText(contact.getMobile());
        return view;
    }
}

```

- ✓ Tạo class mới đặt tên GetContact.java với code như bên dưới. Lớp này được sử dụng để get file JSON từ phía server và tương tác đổ dữ liệu lên view

```
package example.com.android.lab3_ex1_contact;

import android.app.ProgressDialog;
import android.content.Context;
import android.os.AsyncTask;
import android.util.Log;
import android.widget.ListView;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONObject;

import java.util.ArrayList;

public class GetContact extends AsyncTask<Void, Void, Void> {

    private String TAG = MainActivity.class.getSimpleName();

    // URL to get contacts JSON
    //public static String url = "http://192.168.80.1:8080/contacts/";

    public static String url = "http://api.androidhive.info/contacts/";

    ArrayList<Contact> contactList;

    private ProgressDialog pDialog;
    private ListView lv;
    Context context;
    ContactAdapter adapter;

    public GetContact(Context context, ListView lv) {
        this.lv = lv;
        this.context = context;
        contactList = new ArrayList<>();
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new ProgressDialog(context);
        pDialog.setMessage("Please wait...");
        pDialog.setCancelable(false);
        pDialog.show();
    }

    @Override
    protected Void doInBackground(Void... voids) {
        HttpHandler handler = new HttpHandler();

        // making request to url and getting response
        String jsonStr = handler.makeServiceCall(url);
        Log.e(TAG, "Response from url: " + jsonStr);
    }
}
```

```

        if (jsonStr != null) {
            try {
                JSONObject jsonObject = new JSONObject(jsonStr);
                // Getting JSON Array node
                JSONArray contacts = jsonObject.getJSONArray("contacts");

                // looping through all Contacts
                for (int i = 0; i < contacts.length(); i++) {
                    JSONObject c = contacts.getJSONObject(i);

                    String id = c.getString("id");
                    String name = c.getString("name");
                    String email = c.getString("email");
                    String address = c.getString("address");
                    String gender = c.getString("gender");

                    // Phone node is JSON Object
                    JSONObject phone = c.getJSONObject("phone");
                    String mobile = phone.getString("mobile");
                    String home = phone.getString("home");
                    String office = phone.getString("office");

                    Contact contact = new Contact();
                    contact.setId(id);
                    contact.setName(name);
                    contact.setEmail(email);
                    contact.setMobile(mobile);
                    contactList.add(contact);
                }
            } catch (Exception e) {
                e.printStackTrace();
                Log.e(TAG, "Json parsing error: " + e.getMessage());
            }
        } else {
            Log.e(TAG, "Couldn't get json from server.");
        }
        return null;
    }

    @Override
    protected void onPostExecute(Void aVoid) {
        super.onPostExecute(aVoid);
        if (pDialog.isShowing()) {
            pDialog.dismiss();
        }

        adapter = new ContactAdapter(context, contactList);
        lv.setAdapter(adapter);
    }
}

```

✓ Cuối cùng khai báo lớp GetContact vừa tạo và thực thi code



```

public class MainActivity extends AppCompatActivity {

    private ListView lvContact;

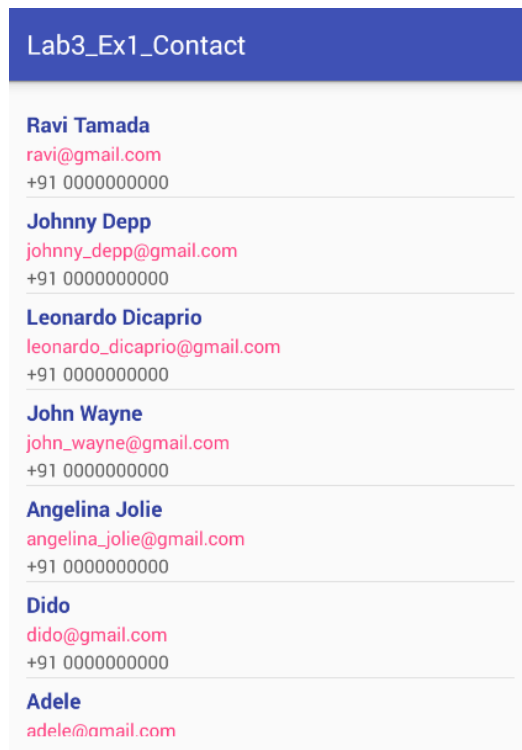
    GetContact getContact;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        lvContact = (ListView) findViewById(R.id.listContact);

        getContact = new GetContact(this, lvContact);
        getContact.execute();
    }
}

```

✓ Chạy ứng dụng và xem kết quả như bên dưới



## BÀI 2 (2 ĐIỂM)

Ở bài này sinh viên thực hiện lấy dữ liệu từ file JSON ở phía server về android và hiển thị lên textview. Sinh viên sử dụng thư viện Volley của Third party để tương tác file JSON. Yêu cầu sinh viên phân biệt được 2 cách lấy dữ liệu là lấy theo Object và lấy theo Array.

**HƯỚNG DẪN:**

- ✓ Tạo mới project android trong android studio
- ✓ Bây giờ tạo tiếp Package dưới thư mục src và đặt tên là app.
- ✓ Sinh viên copy file volley.jar từ thư mục tài nguyên lab vào folder libs của project android
- ✓ Mở file build.gradle và Import thư viện vừa copy vào

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2',
        |     exclude group: 'com.android.support', module: 'support-annotations'
    })
    ⚡ compile 'com.android.support:appcompat-v7:24.2.1'
    compile files('libs/volley.jar')
    testCompile 'junit:junit:4.12'
}
```

- ✓ Tạo class mới đặt tên ApplicationController.java trong package app. Đây sẽ là singleton class nơi mà ta initialize tất cả các core object volley.

```
package example.com.android.lab3_ex2_volley.app;

import android.app.Application;
import android.text.TextUtils;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.Volley;

public class ApplicationController extends Application {
    public static final String TAG = ApplicationController.class.getSimpleName();

    private RequestQueue mRequestQueue;

    private static ApplicationController mInstance;

    @Override
    public void onCreate() {
        super.onCreate();
        mInstance = this;
    }

    public static synchronized ApplicationController getInstance() {
        return mInstance;
    }
}
```

```

public RequestQueue getRequestQueue() {
    if (mRequestQueue == null) {
        mRequestQueue = Volley.newRequestQueue(getApplicationContext());
    }
    return mRequestQueue;
}

public <T> void addToRequestQueue(Request<T> req, String tag) {
    req.setTag(TextUtils.isEmpty(tag) ? TAG : tag);
    getRequestQueue().add(req);
}

public <T> void addToRequestQueue(Request<T> req) {
    req.setTag(TAG);
    getRequestQueue().add(req);
}

public void cancelPendingRequests(Object tag) {
    if (mRequestQueue != null) {
        mRequestQueue.cancelAll(tag);
    }
}
}

```

- ✓ AppController.java phải được thực thi khi ứng dụng chạy. Nên sinh viên add class này vào trong file AndroidManifest.xml sử dụng attribute name cho <application> tag.

```

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="example.com.android.lab3_ex2_volley">

    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:name=".app.AppController"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Lab3_Ex2_Volley"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>

```

## FPT POLYTECHNIC

- ✓ Mở file layout main activity và thêm code như bên dưới. trong layout này chúng ta sử dụng thêm 2 buttons và 1 textview. Trong 2 button. Một sẽ gọi json object request và cái thứ 2 sẽ gọi json array. Textview được sử dụng để hiển thị kết quả json parsed.

```
<?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:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="example.com.android.lab3_ex2_volley.MainActivity">

    <Button
        android:id="@+id/btnObjRequest"
        android:layout_width="match_parent"
        android:layout_marginTop="20dp"
        android:layout_height="wrap_content"
        android:text="Make JSON Object Request" />

    <Button
        android:id="@+id/btnArrRequest"
        android:layout_marginTop="20dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Make JSON Array Request" />

    <TextView
        android:id="@+id/txtResponse"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="40dp"
        android:gravity="center"
        android:padding="4dp"
        android:text="" />
</LinearLayout>
```

- ✓ Bây giờ mở file MainActivity.java và thêm những code đơn giản như sau, thêm button để bắt sự kiện và khởi tạo các đối tượng khác. ở dưới sinh viên có thể thấy có 2 methods 1 là makeJsonObjectRequest() được sử dụng get data của 1 object trong file JSON và 2 là makeJsonArrayRequest() được sử dụng để get Mảng Object trong file JSON.

```
package example.com.android.lab3_ex2_volley;

import android.app.ProgressDialog;
import android.support.v7.app.AppCompatActivity;
```

## FPT POLYTECHNIC

```

import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.VolleyLog;
import com.android.volley.toolbox.JsonArrayRequest;
import com.android.volley.toolbox.JsonObjectRequest;

import org.json.JSONArray;
import org.json.JSONObject;

import example.com.android.lab3_ex2_volley.app.AppController;

public class MainActivity extends AppCompatActivity {

    // json object response url
    private String urlJsonObj =
"http://192.168.80.1:8080/contacts/person_object.json";

    // json array response url
    private String urlJsonArry =
"http://192.168.80.1:8080/contacts/person_array.json";

    private static String TAG = MainActivity.class.getSimpleName();
    private Button btnMakeObjectRequest, btnMakeArrayRequest;

    // progress dialog
    private ProgressDialog pDialog;

    private TextView txtResponse;

    // temporary string to show the parsed response
    private String jsonResponse;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnMakeObjectRequest = (Button) findViewById(R.id.btnObjRequest);
        btnMakeArrayRequest = (Button) findViewById(R.id.btnArrRequest);
        txtResponse = (TextView) findViewById(R.id.txtResponse);

        pDialog = new ProgressDialog(this);
        pDialog.setMessage("Please wait...");
        pDialog.setCancelable(false);

        btnMakeObjectRequest.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

```

```

        makeJsonObjectRequest();
    }
});
btnMakeArrayRequest.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        makeJsonArrayRequest();
    }
});

private void makeJsonArrayRequest() {
    showDialog();

    JsonRequest arrayRequest = new JsonRequest(urlJsonArray,
    new Response.Listener<JSONArray>() {
        @Override
        public void onResponse(JSONArray jsonArray) {
            Log.d(TAG, jsonArray.toString());
            try {
                // Parsing json array response
                // loop through each json object
                jsonResponse = "";
                for (int i = 0; i < jsonArray.length(); i++) {
                    JSONObject person = (JSONObject) jsonArray.get(i);

                    String name = person.getString("name");
                    String email = person.getString("email");
                    JSONObject phone = person.getJSONObject("phone");
                    String home = phone.getString("home");
                    String mobile = phone.getString("mobile");

                    jsonResponse += "Name: " + name + "\n\n";
                    jsonResponse += "Email: " + email + "\n\n";
                    jsonResponse += "Home: " + home + "\n\n";
                    jsonResponse += "Mobile: " + mobile + "\n\n";

                }
                txtResponse.setText(jsonResponse);
            } catch (Exception e) {
                e.printStackTrace();
                Toast.makeText(getApplicationContext(), "Error: "
                + e.getMessage(), Toast.LENGTH_LONG).show();
            }

            hideDialog();
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError volleyError) {
            VolleyLog.d(TAG, "Error: " + volleyError.getMessage());

            Toast.makeText(getApplicationContext(), volleyError.getMessage(), Toast.LENGTH
            _LONG).show();

            hideDialog();
        }
    });
}

```

```

        // Adding request to request queue
        AppController.getInstance().addToRequestQueue(arrayRequest);

    }

    private void makeJsonObjectRequest() {
        showDialog();

        JsonObjectRequest jsonObjReq = new
        JsonObjectRequest(Request.Method.GET, urlJsonObj,
            null, new Response.Listener<JSONObject>() {
                @Override
                public void onResponse(JSONObject jsonObject) {
                    Log.d(TAG, jsonObject.toString());

                    try {
                        // Parsing json object response
                        // response will be a json object
                        String name = jsonObject.getString("name");
                        String email = jsonObject.getString("email");
                        JSONObject phone = jsonObject.getJSONObject("phone");
                        String home = phone.getString("home");
                        String mobile = phone.getString("mobile");

                        jsonResponse = "";
                        jsonResponse += "Name: " + name + "\n\n";
                        jsonResponse += "Email: " + email + "\n\n";
                        jsonResponse += "Home: " + home + "\n\n";
                        jsonResponse += "Phone: " + mobile + "\n\n";

                        txtResponse.setText(jsonResponse);

                    } catch (Exception e) {
                        e.printStackTrace();
                        Toast.makeText(getApplicationContext(), "Error: " +
                        e.getMessage(), Toast.LENGTH_LONG).show();
                    }
                    hideDialog();
                }
            }, new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError volleyError) {
                    VolleyLog.d(TAG, "Error: " + volleyError.getMessage());
                    Toast.makeText(getApplicationContext(), "Error: " +
                    volleyError.getMessage(), Toast.LENGTH_LONG).show();
                    hideDialog();
                }
            });

        // Adding request to request queue

        AppController.getInstance().addToRequestQueue(jsonObjReq);
    }

    private void showDialog() {
        if (!pDialog.isShowing()) {
            pDialog.show();
        }
    }

```

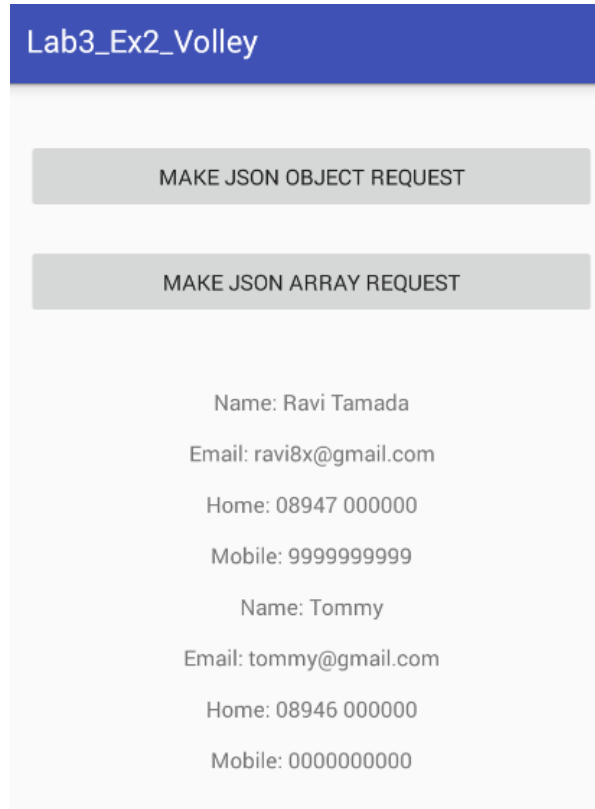
```

    }

    private void hideDialog() {
        if (pDialog.isShowing()) {
            pDialog.dismiss();
        }
    }
}

```

- ✓ Chạy ứng dụng và xem kết quả như bên dưới



### BÀI 3 (2 ĐIỂM)

Viết chương trình parse JSON data từ url và hiển thị lên RecyclerView sử dụng thư viện Retrofit của Third party. Retrofit là một thư viện HTTP client của Square Inc. thư viện này sẽ làm cho kết nối server trở nên dễ dàng hơn và dễ sử dụng.

#### HƯỚNG DẪN:

- ✓ Tạo mới project trong android studio
- ✓ Mở file build.gradle lên và import thư viện Retrofit và GSON. Khi sử dụng



## FPT POLYTECHNIC

Retrofit thì thư viện GSON sẽ giúp chúng ta convert từ Java objects thành JSON và ngược lại.

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:23.2.1'
    testCompile 'junit:junit:4.12'
    compile 'com.squareup.retrofit2:retrofit:2.0.0-beta4'
    compile 'com.squareup.retrofit2:converter-gson:2.0.0-beta4'
    compile 'com.android.support:appcompat-v7:23.2.0'
    compile 'com.android.support:recyclerview-v7:23.2.0'
    compile 'com.android.support:cardview-v7:23.2.0'
}
```

- ✓ Thêm internet permission trong file AndroidManifest.xml

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

- ✓ Mở file layout chính và thêm 1 RecyclerView

```
<?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:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="example.com.android.lab3_ex4_retrofit.MainActivity">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/card_recycler_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

    </android.support.v7.widget.RecyclerView>

</LinearLayout>
```

- ✓ Tạo file layout mới đặt tên card\_row. Thêm 3 textview hiển thị android version name, version number và API Level

```
<?xml version="1.0" encoding="utf-8" ?>
<android.support.v7.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
```

```

        android:layout_marginLeft="5dp"
        android:layout_marginRight="5dp"
        android:layout_marginTop="5dp"
        card_view:cardCornerRadius="5dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_margin="10dp"
            android:orientation="vertical">

            <TextView
                android:id="@+id/tv_name"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginBottom="10dp"
                android:layout_marginTop="10dp"
                android:textSize="18sp"
                android:textStyle="bold" />

            <TextView
                android:id="@+id/tv_version"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />

            <TextView
                android:id="@+id/tv_api_level"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />

        </LinearLayout>
    </android.support.v7.widget.CardView>

```

- ✓ Tạo class mới đặt tên AndroidVersion.java, đây là class model lưu trữ Android version name, version number và API level giống với file JSON chúng ta get.

```

package example.com.android.lab3_ex4_retrofit;

public class AndroidVersion {
    private String ver;
    private String name;
    private String api;

    public String getVer() {
        return ver;
    }

    public String getName() {
        return name;
    }

    public String getApi() {
        return api;
    }
}

```

- ✓ Tạo class mới đặt tên JSONResponse. Tạo một mảng AndroidVersion đặt tên android và tạo getter cho nó.

```
package example.com.android.lab3_ex4_retrofit;

public class JSONResponse {
    private AndroidVersion[] android;

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

- ✓ Tạo Retrofit interface. Khai báo endpoint. Ở đây chúng ta sử dụng GET để nhận JSON data. Endpoint được định nghĩa @GET annotation. Và request url <http://api.learn2crack.com/android/jsonandroid>, nơi <http://api.learn2crack.com> là url cơ sở và [android/jsonandroid](#) là endpoint.

```
package example.com.android.lab3_ex4_retrofit;

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

public interface RequestInterface {
    @GET("android/jsonandroid")
    Call<JSONResponse> getJSON();
}
```

- ✓ Tạo class mới đặt tên DataAdapter.java, với code như bên dưới

```
package example.com.android.lab3_ex4_retrofit;

import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import java.util.ArrayList;

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

    private ArrayList<AndroidVersion> android;

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

```

@Override
public DataAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
    View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.card_row, parent,
false);
    return new ViewHolder(view);
}

@Override
public void onBindViewHolder(DataAdapter.ViewHolder viewHolder, int i) {
    viewHolder.tv_name.setText(android.get(i).getName());
    viewHolder.tv_version.setText(android.get(i).getVer());
    viewHolder.tv_api_level.setText(android.get(i).getApi());
}

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

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

        tv_name = (TextView) view.findViewById(R.id.tv_name);
        tv_version = (TextView) view.findViewById(R.id.tv_version);
        tv_api_level = (TextView) view.findViewById(R.id.tv_api_level);
    }
}
}

```

- ✓ Bây giờ mở file MainActivity.java và tạo method initViews(), để ánh xạ các view đã khai báo và ở phương thức loadJSON() chúng ta sẽ khởi tạo Retrofit để thực hiện công việc kết nối với mạng.

✓

```

package example.com.android.lab3_ex4_retrofit;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;

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;

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

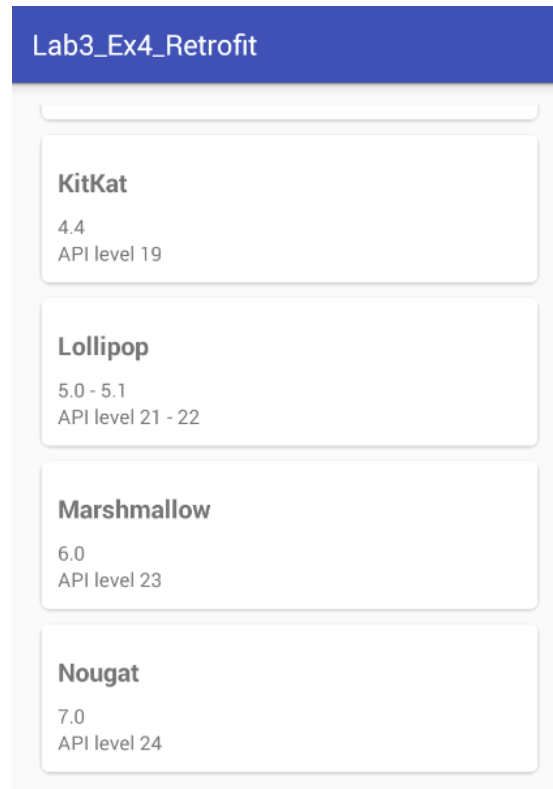
    private void initView() {
        recyclerView = (RecyclerView) findViewById(R.id.card_recycler_view);
        recyclerView.setHasFixedSize(true);
        RecyclerView.LayoutManager layoutManager = new
LinearLayoutManager(getApplicationContext());
        recyclerView.setLayoutManager(layoutManager);
        loadJSON();
    }

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

                JSONResponse jsonResponse = response.body();
                data = new
ArrayList<>(Arrays.asList(jsonResponse.getAndroid()));
                adapter = new DataAdapter(data);
                recyclerView.setAdapter(adapter);
            }

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

✓ Chạy ứng dụng và xem kết quả như bên dưới



## BÀI 4 (2 ĐIỂM)

Viết chương trình hiển thị danh sách contacts lên listview từ file JSON phía server về android có hình ảnh. Sử dụng thư viện Retrofit fetch data và picasso để load hình.

### HƯỚNG DẪN:

- ✓ Tạo class mới đặt tên là Contact.java với nội dung như bên dưới

```
package example.com.android.lab3_ex5_retrofit;

import com.google.gson.annotations.Expose;
import com.google.gson.annotations.SerializedName;

/**
 * Created by Han on 27/12/2016.
 */

public class Contact {
    @SerializedName("id")
    @Expose
    private String id;

    @SerializedName("name")
    @Expose
    private String name;
```

## FPT POLYTECHNIC

```

@SerializedName("email")
@Expose
private String email;

@SerializedName("address")
@Expose
private String address;

@SerializedName("gender")
@Expose
private String gender;

@SerializedName("profile_pic")
@Expose
private String profilePic;

@SerializedName("phone")
@Expose
private Phone phone;

/**
 * @return The id
 */
public String getId() {
    return id;
}

/**
 * @param id The id
 */
public void setId(String id) {
    this.id = id;
}

/**
 * @return The name
 */
public String getName() {
    return name;
}

/**
 * @param name The name
 */
public void setName(String name) {
    this.name = name;
}

/**
 * @return The email
 */
public String getEmail() {
    return email;
}

/**
 * @param email The email
 */

```

## FPT POLYTECHNIC

```

public void setEmail(String email) {
    this.email = email;
}

/**
 * @return The address
 */
public String getAddress() {
    return address;
}

/**
 * @param address The address
 */
public void setAddress(String address) {
    this.address = address;
}

/**
 * @return The gender
 */
public String getGender() {
    return gender;
}

/**
 * @param gender The gender
 */
public void setGender(String gender) {
    this.gender = gender;
}

/**
 * @return The profilePic
 */
public String getProfilePic() {
    return profilePic;
}

/**
 * @param profilePic The profile_pic
 */
public void setProfilePic(String profilePic) {
    this.profilePic = profilePic;
}

/**
 * @return The phone
 */
public Phone getPhone() {
    return phone;
}

/**
 * @param phone The phone
 */
public void setPhone(Phone phone) {
    this.phone = phone;
}

```



```

    }

    public class Phone {

        @SerializedName("mobile")
        @Expose
        private String mobile;

        @SerializedName("home")
        @Expose
        private String home;

        @SerializedName("office")
        @Expose
        private String office;

        /**
         * @return The mobile
         */
        public String getMobile() {
            return mobile;
        }

        /**
         * @param mobile The mobile
         */
        public void setMobile(String mobile) {
            this.mobile = mobile;
        }

        /**
         * @return The home
         */
        public String getHome() {
            return home;
        }

        /**
         * @param home The home
         */
        public void setHome(String home) {
            this.home = home;
        }

        /**
         * @return The office
         */
        public String getOffice() {
            return office;
        }

        /**
         * @param office The office
         */
        public void setOffice(String office) {
            this.office = office;
        }
    }

```

- ✓ Tạo class mới đặt tên ContactList.java chứa danh sách contact và thực hiện tạo getter setter.

```
package example.com.android.lab3_ex5_retrofit;

import com.google.gson.annotations.Expose;
import com.google.gson.annotations.SerializedName;

import java.util.ArrayList;

public class ContactList {
    @SerializedName("contacts")
    @Expose
    private ArrayList<Contact> contacts = new ArrayList<>();

    /**
     * @return The contacts
     */
    public ArrayList<Contact> getContacts() {
        return contacts;
    }

    /**
     * @param contacts The contacts
     */
    public void setContacts(ArrayList<Contact> contacts) {
        this.contacts = contacts;
    }
}
```

- ✓ Tạo interface và định nghĩa endpoint(file json)

```
package example.com.android.lab3_ex5_retrofit;

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

public interface ApiService {
    /**
     * Retrofit get annotation with our URL
     * And our method that will return us the List of ContactList
     */
    @GET("/json_data.json")
    Call<ContactList> getMyJSON();
}
```

- ✓ Để gửi yêu cầu kết nối mạng tới API. Chúng ta cần sử dụng Retrofit Builder

class và url cụ thể cho service. Tạo class mới đặt tên RetroClient.java

```
package example.com.android.lab3_ex5_retrofit;

import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class RetroClient {

    private static final String ROOT_URL = "http://pratikbutani.x10.mx";

    /**
     * Get Retrofit Instance
     */
    private static Retrofit getRetrofitInstance() {
        return new Retrofit.Builder()
            .baseUrl(ROOT_URL)
            .addConverterFactory(GsonConverterFactory.create())
            .build();
    }

    /**
     * Get API Service
     */
    @return API Service
    public static ApiService getApiService() {
        return getRetrofitInstance().create(ApiService.class);
    }
}
```

✓ Tạo file layout mới đặt diện một dòng đặt tên là layout\_row\_view.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_margin="5dp"
        android:src="@mipmap/ic_launcher" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:layout_toRightOf="@+id/imageView"
        android:orientation="vertical">

        <TextView
            android:id="@+id/textViewName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
```

```

        android:layout_margin="5dp"
        tools:text="TextView" />

    <TextView
        android:id="@+id/textViewEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        tools:text="TextView" />

</LinearLayout>
</RelativeLayout>

```

- ✓ Tạo class mới đặt tên MyContactAdapter.java, lớp này sẽ xử lý việc đổ dữ liệu lên listview

```

package example.com.android.lab3_ex5_retrofit;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.TextView;

import com.squareup.picasso.Picasso;

import java.util.List;

public class MyContactAdapter extends ArrayAdapter<Contact> {

    List<Contact> contactList;
    Context context;
    private LayoutInflater mInflater;

    // Constructors
    public MyContactAdapter(Context context, List<Contact> objects) {
        super(context, 0, objects);
        this.context = context;
        this.mInflater = LayoutInflater.from(context);
        contactList = objects;
    }

    @Override
    public Contact getItem(int position) {
        return contactList.get(position);
    }

    private static class ViewHolder {
        public final RelativeLayout rootView;
        public final ImageView imageView;
        public final TextView textViewName;
        public final TextView textViewEmail;
    }
}

```

```

        private ViewHolder(RelativeLayout rootView, ImageView imageView,
            TextView textViewName, TextView textViewEmail) {
            this.rootView = rootView;
            this.imageView = imageView;
            this.textViewName = textViewName;
            this.textViewEmail = textViewEmail;
        }

        public static ViewHolder create(RelativeLayout rootView) {
            ImageView imageView = (ImageView)
            rootView.findViewById(R.id.imageView);
            TextView textViewName = (TextView)
            rootView.findViewById(R.id.textViewName);
            TextView textViewEmail = (TextView)
            rootView.findViewById(R.id.textViewEmail);
            return new ViewHolder(rootView, imageView, textViewName,
            textViewEmail);
        }
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        final ViewHolder vh;
        if (convertView == null) {
            View view = mInflater.inflate(R.layout.layout_row_view, parent,
            false);
            vh = ViewHolder.create((RelativeLayout) view);
            view.setTag(vh);
        } else {
            vh = (ViewHolder) convertView.getTag();
        }

        Contact item = getItem(position);

        vh.textViewName.setText(item.getName());
        vh.textViewEmail.setText(item.getEmail());

        Picasso.with(context).load(item.getProfilePic()).placeholder(R.mipmap.ic_launcher).error(R.mipmap.ic_launcher).into(vh.imageView);

        return vh.rootView;
    }
}

```

- ✓ Tạo class mới đặt tên InternetConnection.java, lớp này sẽ kiểm tra việc kết nối mạng của ứng dụng

```

package example.com.android.lab3_ex5_retrofit;

import android.content.Context;
import android.net.ConnectivityManager;
import android.support.annotation.NonNull;
public class InternetConnection {
    /**
     * CHECK WHETHER INTERNET CONNECTION IS AVAILABLE OR NOT

```

```

    */
    public static boolean checkConnection(@NonNull Context context) {
        return ((ConnectivityManager) context.getSystemService
            (Context.CONNECTIVITY_SERVICE)).getActiveNetworkInfo() !=
null;
    }
}

```

- ✓ Qua file activity\_main.xml và Thêm id cho layout tổng

```

<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    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"
    android:fitsSystemWindows="true"
    android:id="@+id/parentLayout"
    tools:context="example.com.android.lab3_ex5_retrofit.MainActivity">

```

- ✓ Thay đổi icon ở view floatbutton

```

<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_margin="16dp"
    app:srcCompat="@android:drawable/stat_sys_download" />

```

- ✓ Bây giờ quay lại mở file MainActivity.java, khai báo và ánh xạ listview, cuối cùng send request lên server và get data.

```

package example.com.android.lab3_ex5_retrofit;

import android.app.ProgressDialog;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Gravity;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;
import android.widget.Toast;

import java.util.ArrayList;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class MainActivity extends AppCompatActivity {
    private ListView listView;

```

```

private View parentView;

private ArrayList<Contact> contactList;
private MyContactAdapter adapter;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    contactList = new ArrayList<>();

    parentView = findViewById(R.id.parentLayout);

    listView = (ListView) findViewById(R.id.listview);
    listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
        Snackbar.make(parentView,
contactList.get(position).getName() + " => " +
contactList.get(position).getPhone().getHome(),
Snackbar.LENGTH_LONG).show();
    }
});
    Toast toast = Toast.makeText(getApplicationContext(),
R.string.string_click_to_load, Toast.LENGTH_LONG);
    toast.setGravity(Gravity.CENTER, 0, 0);
    toast.show();

    FloatingActionButton fab = (FloatingActionButton)
findViewById(R.id.fab);
    assert fab != null;
    fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        if
        (InternetConnection.checkConnection(getApplicationContext())) {
            final ProgressDialog dialog;

            dialog = new ProgressDialog(MainActivity.this);

            dialog.setTitle(getString(R.string.string_getting_json_title));

            dialog.setMessage(getString(R.string.string_getting_json_message));
            dialog.show();

            //Creating an object of our api interface
            ApiService api = RetroClient.getApiService();

            // Calling JSON
            Call<ContactList> call = api.getMyJSON();

```

```

        // Enqueue Callback will be call when get response...
        call.enqueue(new Callback<ContactList>() {
            @Override
            public void onResponse(Call<ContactList> call,
Response<ContactList> response) {
                //Dismiss Dialog
                dialog.dismiss();

                if(response.isSuccessful()) {
                    // Got Successfully
                    contactList = response.body().getContacts();

                    // Binding that List to Adapter
                    adapter = new
MyContactAdapter(MainActivity.this, contactList);
                    listView.setAdapter(adapter);

                } else {
                    Snackbar.make(parentView,
R.string.string_some_thing_wrong, Snackbar.LENGTH_LONG).show();
                }
            }
            @Override
            public void onFailure(Call<ContactList> call,
Throwable t) {
                dialog.dismiss();
            }
        });

        } else {
            Snackbar.make(parentView,
R.string.string_internet_connection_not_available,
Snackbar.LENGTH_LONG).show();
        }

    }

    });
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is
present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }
}

```



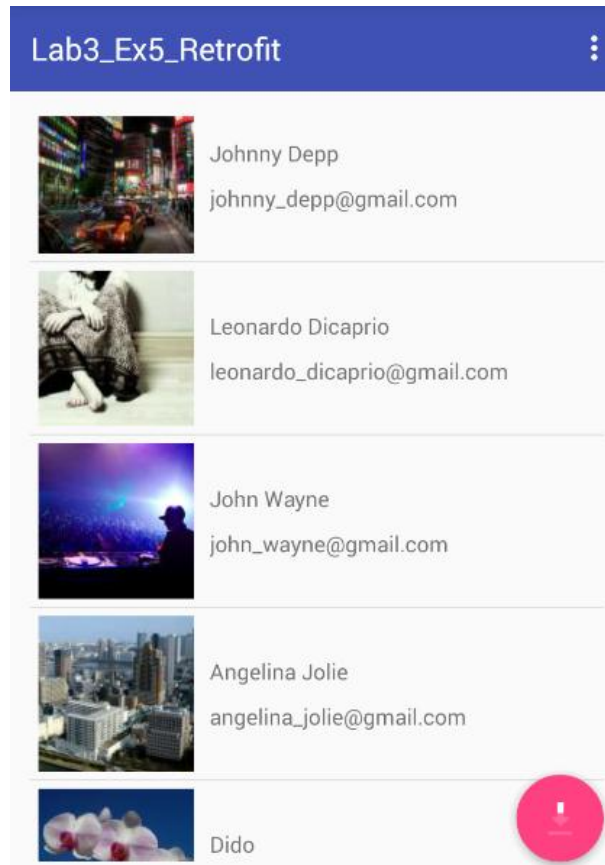
```

    }

    return super.onOptionsItemSelected(item);
}
}

```

- ✓ Chạy ứng dụng click vào button download và xem kết quả như bên dưới.



## BÀI 5 (2 ĐIỂM):

Giảng viên cho thêm