



ANDROID NETWORK

BÀI 3: JSON PARSER

- ❖ Định dạng dữ liệu JSON
- ❖ JSON parser
- ❖ Lấy và parser dữ liệu JSON từ Server
- ❖ Thư viện Volley, Retrofit

- ✓ JSON (JavaScript Object Notation) là một chuẩn thuần văn bản và có cấu trúc để giúp việc truyền tải dữ liệu.
- ✓ Nó có nguồn gốc từ ngôn ngữ kịch bản javascript để trình bày cấu trúc dữ liệu đơn giản kết hợp với mảng, được gọi là object.
- ✓ Dù nó có mối quan hệ với javascript nhưng nó là ngôn ngữ độc lập có thể được phân tích trong các ngôn ngữ khác.

- ✓ JSON có các kiểu dữ liệu:
 - Number
 - String
 - Boolean
 - Array
 - Object
 - null
- ✓ Mục đích của JSON cũng giống như XML tuy nhiên cách viết ngắn hơn và cách phân tích (parser) cũng dễ hơn.

- ✓ Cú pháp json theo các qui định sau:
 - Dữ liệu gồm các cặp tên / giá trị
 - Dữ liệu được phân cách dấu phẩy “,”
 - Dấu ngoặc nhọn “{ }” dùng để giữ các đối tượng (object).
 - Dấu ngoặc vuông “[]” dùng để giữ các mảng (array)
- ✓ Dữ liệu được viết là các **cặp tên / giá trị**. Một cặp tên / giá trị gồm 1 trường tên đặt trong dấu nháy kép, tiếp theo là dấu hai chấm và tiếp theo là giá trị (giá trị có dấu nháy kép hay không tùy theo kiểu dữ liệu).

Ví dụ: “ten”:“teo”

 “tuoi”:20

- ✓ Đối tượng JSON (JSON Object): được viết trong cặp dấu ngoặc nhọn " { } ". Trong một đối tượng có có nhiều cặp tên / giá trị phân cách nhau bằng dấu phẩy ",".

```
{ "ten" : "teo" , "tuoi" : 20 }
```

- ✓ Mảng JSON (JSON Array): được bọc trong dấu ngoặc vuông " [] ". Một mảng có thể chứa nhiều đối tượng (object).

```
{  
  "nhanvien" : [  
    { "ten" : "teo" , "tuoi" : 20 },  
    { "ten" : "ti" , "tuoi" : 25 },  
    { "ten" : "tin" , "tuoi" : 35 }  
  ]  
}
```

- ✓ Đối tượng JSON (JSON Object): được viết trong cặp dấu ngoặc nhọn " { } ". Trong một đối tượng có có nhiều cặp tên / giá trị phân cách nhau bằng dấu phẩy ",".

```
{ "ten" : "teo" , "tuoi" : 20 }
```

- ✓ Mảng JSON (JSON Array): được bọc trong dấu ngoặc vuông " [] ". Một mảng có thể chứa nhiều đối tượng (object).

```
{  
  "nhanvien" : [  
    { "ten" : "teo" , "tuoi" : 20 },  
    { "ten" : "ti" , "tuoi" : 25 },  
    { "ten" : "tin" , "tuoi" : 35 }  
  ]  
}
```

- ✓ Ta thấy cách viết JSON đơn giản hơn XML

```
<root>
  <nhanvien>
    <ten>teo</ten>
    <tuoi>20</tuoi>
  </nhanvien>
  <nhanvien>
    <ten>ti</ten>
    <tuoi>25</tuoi>
  </nhanvien>
  <nhanvien>
    <ten>tin</ten>
    <tuoi>35</tuoi>
  </nhanvien>
</root>
```


Phân tích JSON:

- ✓ Tạo đối tượng JSONObject

```
JSONObject jsonobject = new JSONObject(result);
```

- ✓ Lấy chuỗi

```
String jsonstring= jsonobject.getString("TENCHUOI");
```

- ✓ Lấy boolean

```
String jsonboolean=jsonobject.getBoolean("TENBOOLEAN");
```

✓ Lấy mảng

```
JSONArray jsonarray=jsonobject.getJSONArray("ARRAYNAME");
```

✓ Lấy Item từ mảng

```
for(int i=0 ; i<jsonarray.length( ) ; i+ +)
```

```
{
```

```
    JSONObject oneobject=jsonarray.getJSONObject(i);
```

```
    String item1=oneobject.getString("STRINGNAMEINARRAY");
```

```
    String item1=oneobject.getString("STRINGNAMEINARRAY");
```

```
}
```

JSON cũng giống như là XML nên cũng có thể dùng thao tác làm việc với HTTP để đọc dữ liệu từ server xuống, sau khi đọc xong ta sẽ phân tích JSON.

✓ Lấy dữ liệu từ server với Http

```
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;  
}
```

Get file JSON từ phía server và đổ dữ liệu lên view

```
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();
    progressDialog = new ProgressDialog(context);
    progressDialog.setMessage("Please wait...");
    progressDialog.setCancelable(false);
    progressDialog.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);  
}
```

Gọi getContact trong MainActivity

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




DEMO

JSON Parser



- ✓ Volley được Google giới thiệu vào khoảng tháng 6/2013. Volley là một thư viện dùng để send và receive response từ Server sử dụng giao thức HTTP.
- ✓ Volley có các điểm nổi bật dưới đây:
 - Tự động lập lịch (scheduling) cho các request.
 - Caching Response.
 - Hỗ trợ set độ ưu tiên cho các request (priority).
 - Hỗ trợ nhiều kết quả trả về (String, JSONObject, JSONArray, Bitmap...)
 - Có thể huỷ Request.

Các class sử dụng trong Volley:

- ✓ ***RequestQueue***: Là hàng đợi giữ các Request.
- ✓ ***Request***: là lớp cơ sở của các Request trong Volley, chứa thông tin về request HTTP.
- ✓ ***StringRequest***: Kế thừa từ Request, là class đại diện cho request trả về String.
- ✓ ***JSONObjectRequest***: Là HTTP request có response trả về là JSONObject.
- ✓ ***JSONArrayRequest***: Là HTTP request có response trả về là JSONArray.
- ✓ ***ImageRequest***: Là HTTP request có response trả về là Bitmap.

- ✓ Trước tiên chúng ta phải import thư viện này vào Android Studio. Copy và paste dòng dưới đây là **dependencies** trong file **build.gradle** của module **app**

```
dependencies {  
    ...  
    compile 'com.android.volley:volley:1.0.0'  
}
```

Nhấn **Async Now** để Android Studio download và nạp thư viện vào project.

- ✓ Để sử dụng Volley chúng ta phải cấp quyền Internet trong AndroidManifest.xml

- ✓ Hoặc import thư viện vào project và khai báo **dependencies** trong file **build.gradle** của module **app**

```
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'  
}
```

Nhấn **Async Now** để Android Studio download và nạp thư viện vào project.



DEMO

Sử dụng Thư viện Volley



Retrofit là một Rest Client cho Android và Java và được tạo ra bởi Square. Nó giúp cho việc nhận và tải lên JSON (hoặc dữ liệu khác) một cách khá dễ dàng tới một Webservice dựa trên mô hình REST.

Mở file build.gradle lên và import thư viện Retrofit và GSON. Khi sử dụng 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'  
}
```



DEMO

Sử dụng Thư viện Retrofit



SINH VIÊN TÌM HIỂU THÊM CÁC THƯ VIỆN KHÁC HỖ TRỢ JSON

- ❖ Định dạng dữ liệu JSON
- ❖ JSON parser
- ❖ Lấy và parser dữ liệu JSON từ Server
- ❖ Thư viện Volley, Retrofit

